UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Examination in	INF3580 — Semantic Technologies
Day of examination:	10. juni 2007
Examination hours:	09:00 – 12:00

This problem set consists of 7 pages.

Appendices:	None
Permitted aids:	Any printed course material

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Problem 1 RDF

Consider the RDF document below:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ifi: <http://www.ifi.uio.no/people#> .


ifi:steffen a foaf:Person;
    foaf:knows ifi:vigdis,
               [a foaf:Person;
                  foaf:mbox <mailto:karlsson@taket.se>;
                  foaf:name "Karlsson" ] ;
    foaf:name "Steffen Gilje";
    foaf:homepage <http://www.ifi.uio.no/~steffen/>.
```

Answer the following questions:

(a) Draw a graph representation of this RDF document.

(b) Which URI is used to identify Steffen Gilje in this document?

(c) What kind of resource represents Karlsson in this document?

(d) The `foaf:interest` property is often used to represent an interest of a `foaf:Person`, through indicating a `foaf:Document` whose `foaf:topic`

(Continued on page 2.)

broadly characterises that interest.

Add statements to the document above saying that Steffen is interested in Semantic Web technologies, as represented by the `foaf:Document` indicated by `http://workingontologist.com`, and that the `foaf:topic` of that book is “Semantic Web”.

Problem 2 SPARQL

Consider the following RDF document that contains information about NASA space probe missions.

```
@prefix nasa: <http://www.nasa.gov/vocab#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

nasa:voyager a nasa:SpaceProbe;
             nasa:objective nasa:jupiter, nasa:saturn,
                            nasa:uranus, nasa:neptune.

nasa:mariner9 a nasa:SpaceProbe;
              nasa:objective nasa:mars.

nasa:spirit a nasa:SpaceProbe;
            nasa:objective nasa:mars.

nasa:pioneer10 a nasa:SpaceProbe;
               nasa:objective nasa:jupiter.

nasa:jupiter a nasa:Planet;
             nasa:moon nasa:io, nasa:ganymede, nasa:callisto;
             nasa:distanceFromSun "778"^^xsd:double .

nasa:saturn  a nasa:Planet;
             nasa:distanceFromSun "1426"^^xsd:double .

nasa:neptune a nasa:Planet;
             nasa:moon nasa:triton, nasa:proteus, nasa:naiad;
             nasa:distanceFromSun "4503"^^xsd:double .

nasa:uranus a nasa:Planet;
            nasa:distanceFromSun "2876"^^xsd:double .

nasa:mars a nasa:Planet;
```

(Continued on page 3.)

```
        nasa:distanceFromSun "227"^^xsd:double .
```

Distance from the sun is measured in millions of kilometers. Write a SPARQL query that retrieves

(a) all spaceprobes,

(b) all moons,

(c) for each planet that has one or more moons; the planet and its moons

(d) a planet and its distance from the sun,

(e) all planets that lie further than 2000 million kilometers from the sun,

(f) all planets that have been visited by (i.e. been the objective of) more than one space probe

Problem 3 Manipulating RDF programmatically

Imaging that you are writing a web application to assist people in creating a FOAF file for themselves (admittedly there are many such already). The web page is supposed to present the user with a set of input fields, e.g.

Title (Mr, Mrs, Dr, etc)	▷
First Name	▷
Last Name	▷
Nickname	▷
Your Email Address	▷
Homepage	▷
Your Picture	▷
Phone Number	▷

and the application is supposed to constructs a FOAF graph from the input it receives through these fields.

The code below sketches one way one may go about implementing the back-end of this application, that is, of constructing the FOAF graph from the input fields. Notice the consecutively numbered points in the code marked with asterisks. These are for you to fill in, in accordance with the instructions below:

(Continued on page 4.)

```
final private String foafNS = "http://xmlns.com/foaf/0.1/";
private Model foafModel;
private Resource thePerson;

public void initFoafModel(String foafFileURL){
    foafModel = * Point 1 *
    thePerson = foafModel.createResource(foafFileURL + "#me");
    Resource personType = foafModel.createResource(foafNS + "Person");

    *Point 2*
}

public void addTitle(String title){

    * Point 3 *
}

public void addMail(String mailAdress){

    * Point 4 *
}
```

Instructions:

(a) Construct an empty `Model` object at point 1 using Jena's `ModelFactory` class.

(b) At point 2, add a statement to the graph held by `foafGraph` that says that the resource held by `thePerson` is one whose `rdf:type` is `foaf:Person`.

(c) Complete the method at point 3 to associate a `foaf:title` with `thePerson`.

(d) Complete the method at point 4 to associate a `foaf:mbox` with `thePerson`.

Problem 4 RDFS Reasoning

Below is an excerpt from an RDFS document that keeps track of the cars and deliveries of a delivery service. We assume that the namespaces `dlv`, `rdf`, `xsd` and `rdfs` are given.

(Continued on page 5.)

```
dlv:Car rdf:type rdfs:Class .
dlv:AssignedCar rdf:type rdfs:Class .
dlv:AssignedCar rdfs:subsetOf dlv:Car .
dlv:destination rdfs:domain dlv:AssignedCar .
dlv:destination rdfs:range dlv:City .
dlv:removalist rdfs:range dlv:Employee .
dlv:executive rdfs:range dlv:Employee .
dlv:mover rdfs:subPropertyOf dlv:removalist .
dlv:driver rdfs:subPropertyOf dlv:removalist .
dlv:deliveryDate rdfs:range xsd:dateTime .


dlv:AV43634 dlv:destination dlv:Trondhjem .
            dlv:driver dlv:BjarneBerg .
            dlv:mover  dlv:MagneMo .
            dlv:mover  dlv:HangHiu .
            dlv:executive dlv:LinnLarsson .
            dlv:deliveryDate "2010-10-26T21:00:00" .

dlv:HJ14522 dlv:destination dlv:Grorud .
            dlv:driver dlv:LinnLarsson .
            dlv:mover  dlv:HelgeHareide .
            dlv:executive dlv:LinnLarsson .
            dlv:deliveryDate "2010-09-11T15:00:00" .

dlv:SU56782 rdf:type dlv:Car .
```

For each of the triples below, determine whether it is derivable from the statements in the document. Justify your answer.

(a) `dlv:AV43634 rdf:type dlv:Car .`

(b) `dlv:AV43634 rdf:type dlv:AssignedCar .`

(c) `dlv:SU56782 rdf:type dlv:AssignedCar .`

(d) `"2010-09-11T15:00:00" rdf:type xsd:dateTime .`

(e) `dlv:Trondhjem rdf:type dlv:City .`

(f) `dlv:Grorud rdf:type dlv:City .`

(g) `dlv:HJ14522 dlv:removalist dlv:LinnLarsson .`

(h) `dlv:AV43634 dlv:removalist dlv:LinnLarsson .`

(Continued on page 6.)

(i) `_:blank dlv:employee dlv:MagneMo .`

(j) `_:blank _:blank _:blank .`

(k) `dlv:destination rdf:type rdf:Property .`

Problem 5 Description logics/OWL (25 %)

Express the following sentences in DL-notation, using the the class names `Vehicle`, `Car`, `ElectricCar`, `HybridCar`, `Engine`, `Wheel`, `CombustionEngine`, `ElectricEngine` and the role names `hasPart` and `hasEngine`.

(a) A car is a vehicle with an engine.

(a) A car is a vehicle with a combustion engine or an electric engine.

(b) Every car has exactly four wheels as part.

(c) A four-wheeled vehicle with a combustion engine or an electric engine is a car.

(d) An electric car is a car that has only an electric engine

(e) A hybrid car is a car that has a combustion engine as well as an electric engine.

Problem 6 Description logic/OWL semantics

Consider the ontology consisting of the following axioms:

Axiom 1: $PetShop \sqsubseteq Shop \sqcap \exists offers.Animal$

Axiom 2: $Vertebrate \sqsubseteq Animal$

Axiom 3: $Invertebrate \sqsubseteq Animal$

Axiom 4: $Vertebrate \sqsubseteq \neg Invertebrate$

(a) Is it true, according to this ontology, that an animal can be a vertebrate *as well as* an invertebrate? Justify your answer with reference to the axioms.

(Continued on page 7.)

(a) The ontology does not entail that the things on offer in a pet shop are either vertebrates or invertebrates. To show this, complete the countermodel below by assigning an appropriate subset of $\Delta^{\mathcal{I}}$ to $Animal^{\mathcal{I}}$ and an appropriate relation over $\Delta^{\mathcal{I}}$ to $offers^{\mathcal{I}}$

$\Delta^{\mathcal{I}}$	$= \{petsRus, fido, jellybean\}$
$Vertebrate^{\mathcal{I}}$	$= \{fido\}$
$Invertebrate^{\mathcal{I}}$	$= \{jellybean\}$
$PetShop^{\mathcal{I}} = Shop^{\mathcal{I}}$	$= \{petsRus\}$
$offers^{\mathcal{I}}$	$=$
$Animal^{\mathcal{I}}$	$=$

(c) Remedy this defect (which it may reasonably be taken to be) by adding a covering axiom that applies to animals.