

TU Vienna

2007 Winter Semester

End of Semester Examination

ESW
Einführung in Semantic Web
(ausgearbeitet von Schoaf)
Keine Garantie oder Gewährleistung!

Time: 2.0 Hours **Date of Examination:** January 28th, 2008

SID:

Instructions to the Candidate:

- **Please ensure that your student id number is filled in on this cover!**
- Answer all questions in this booklet in the spaces provided.
- The size of the empty space reflects roughly the expectation on the length of your answer.
- You may use any course material.
- For multiple-choice questions any number of correct answers is possible (zero, 1, ..., all).
- Text questions must be answered in a few sentences.
- If you run out of space, you can use the empty page on the left.

Question 1

(10%)

Formalize each of the following statements into description logic (DL). When a particular statement(or apart of it) cannot be fully expressed in DL, explain why not.

- Someone is a CATOWNER if he (or she) owns at least one CAT.

$\text{CATOWNER} \equiv \text{Person} \sqcap (\geq 1 \text{ ownsCAT})$

- sackpicker is a CAT.

$\text{sackpicker} : \text{CAT}$

- Always when someone is a CATOWNER he mustbe a CATLOVER.

$\text{CATOWNER} \sqsubseteq \text{CATLOVER}$

- If a particular person loves a particular cat, then that person also owns that cat.

You cannot express that someone loves a particular cat.
You cannot refer to that particular cat because there are no variables.

- A DOGLOVER is a PERSON who owns only DOGs as an ANIMAL (and not a single CAT, which is also an ANIMAL).

$\text{DOGLOVER} \equiv \text{Person} \sqcap (\forall \text{ ownsAnimal.DOG}) \sqcap \neg(\exists \text{ ownsAnimal.} \neg\text{DOG})$

Question 2

(10%)

In Topi cMaps the identification of subjects is done with subject identifiers (sometimes called *indicators*) and/or subject locators (sometimes called the *addresses*). Explain how these relate to the identification of resources in RDF.

subject identifiers:

When a URI is used to identify a subject indirectly, it is called a subject identifier. That means because the subject is non-addressable it refers to a so called "subject indicator" which is a human interpretable information resource which provides some identification information about the subject (like a HTML site).

subject address:

Is a URI to identify a subject directly. The resource of this URI is the subject itself (the subject address for the Website of [rho] is [http://kill.devc.at])

In RDF there is no difference for address or identifier. There are only identifiers as URIs and it doesn't matter whether they are real addresses to link somewhere.

Question 3

(9%)

Consider the following T-box:

```
Bundesland a owl:Class ;
    owl:oneOf (:Wien :Burgenland :Salzburg) .

:Person a owl:Class .

:istHauptGemeldet
    a owl:ObjectProperty ;
    rdfs:domain :Person .

:Wiener a owl:Class ;
    rdfs:subClassOf :Person ;
    rdfs:subClassOf
        [ a owl:Restriction ;
          owl:onProperty :istHauptGemeldet ;
          owl:someValuesFrom
              [ a owl:Class ;
                owl:oneOf (:Wien)
              ]
        ] .

:NichtWiener a owl:Class ;
    rdfs:subClassOf :Person ;
    owl:equivalentClass
        [ a owl:Class ;
          owl:complementOf :Wiener
        ] .

:Oesterreicher a owl:Class ;
    rdfs:subClassOf :Person ;
    owl:equivalentClass
        [ a owl:Class ;
          owl:unionOf (:Wiener :NichtWiener)
        ] .
```

- **Is it inconsistent? Justify your answer (e.g. by finding a model).**

Besides no prefixes are given the ontology is consistend.
(checked by <http://www.mindswap.org/2003/pellet/demo.shtml>)

- **If it is consistent, does it have another serious problem? Justify your answer.**

The Class [:Oesterreicher] would not only be Austrians but all non-Vienese persons. That is because [:NichtWiener] is a class of all persons that are not "Hauptgemeldet" in Vienna, therefore this would also be Queensland.

Question 4

(9%)

Given the following database schema for a relational database

Person := (id, Name, Birthdate)

Course := (id, Title, Code, Synopsis)

Student := (Person.id, Course.id, MatrikelNr)

- How could this be expressed in RDF-S (any notation will do)?

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix :         <http://SemWeb.exam.tuwien.ac.at#> .

:Person rdf:type owl:Class.
:Name rdf:type owl:ObjectProperty;
      rdfs:domain :Person.
:Birthdate rdf:type owl:ObjectProperty;
          rdfs:domain :Person.

:Course rdf:type owl:Class.
:Title rdf:type owl:ObjectProperty;
      rdfs:domain :Course.
:Code rdf:type owl:ObjectProperty;
      rdfs:domain :Course.
:Synopsis rdf:type owl:ObjectProperty;
          rdfs:domain :Course.

:Student rdf:type owl:Class.
:PersonId rdf:type owl:ObjectProperty;
          rdfs:domain :Student;
          rdfs:range :Person.
:visitsCourse rdf:type owl:ObjectProperty;
              rdfs:domain :Student;
              rdfs:range :Course.
:MatrikelNr rdf:type owl:ObjectProperty;
            rdfs:domain :Student.
```

- Write a SPARQL query against your RDF-S schema to select all students (only their birthdates) from the course ESW (use proper namespaces).

```
PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:     <http://www.w3.org/2000/01/rdf-schema#>
PREFIX :         <http://SemWeb.exam.tuwien.ac.at#>
SELECT ?birthday
WHERE {
    ?course :Title 'ESW' .
    ?Student :visitsCourse ?course .
    ?Student :Birthdate ?birthday .
}
```

- How would this SPARQL query be translated into an SQL query statement for the above?

```
SELECT Person.Birthdate
FROM Course, Student, Person
WHERE Course.Title = 'ESW'
      AND Course.id = Student.CourseId
      AND Student.id = Person.id
```

Question 5

(10%)

Given the following ontology (together with some instance data)

```
@prefix : <http://www.whatever.com/#> .

:Person a owl:Class .
:Opus a owl:Class .
:Painting a owl:Class ;
    rdfs:subClassOf :Opus .
:MasterPiece a owl:Class ;
    rdfs:subClassOf :Painting .
:hasCreated a owl:ObjectProperty .
:hasPainted a owl:ObjectProperty ;
    rdfs:subPropertyOf :hasCreated .
:Genius a owl:Class ;
    rdfs:subClassOf :Person ;
    owl:equivalentClass
        [ a owl:Restriction ;
          owl:onProperty :hasCreated ;
          owl:someValuesFrom :MasterPiece
        ] .
:VanGogh a :Person ;
    :hasPainted :TheOldMill .
:TheOldMill a :MasterPiece .
```

- If the SPARQL query

```
SELECT ?person
WHERE {
    ?person a <http://www.whatever.com/#Genius>
}
```

is used with software (such as RedlandlibRDF) which does not support DL inferencing, what would be the expected result?

```
person
→ unbound variable person
```

- If you **have to** use, say, Redland, but you need inferencing, what are your technical options? Describe at least one.

```
Edit and save the ontology in Redland,
then export it, maybe with the DIG-Format or as RDF and use a reasoner
like Pellet, FaCT, RACER, ...
then reload the inferred data into Redland to continue working.
```

Question 6

(15%)

Translate the following DL formulas into FOL (predicate logic):

- $CATLOVER \equiv \exists \text{ loves.CAT} \sqcap PERSON$

$CATLOVER(x) \Leftrightarrow [\exists y. \text{loves}(x,y) \wedge CAT(y)] \wedge PERSON(x)$

- $CATHATERS \sqsubseteq PERSON \sqcap \neg CATLOVER$

$\forall(x) CATHATERS(x) \Rightarrow PERSON(x) \wedge \neg CATLOVER(x)$

- $\forall \text{ loves.CAT} \sqsupseteq \exists \text{ hates.DOG}$

$\forall y. \text{loves}(x,y) \rightarrow CAT(y) \Leftarrow \exists y. \text{hates}(x,y) \wedge DOG(y)$

Translate the following FOL formulas into DL:

- $\forall d \in DOG : \forall p \in PERSON : \text{loves}(d, p) \implies \text{loves}(p, d)$

- $\forall c \in CAT : \exists p \in PERSON : \text{loves}(c, p) \implies \text{loves}(p, c)$

Both are not expressible with DL because there are no variables in DL. ??

Question 7

(15%)

Regarding property characteristics in OWL:

- If you tag a property p as *symmetric* and *functional*, what are the consequences for the property and the connected nodes, if any?

Symmetric: $P(x, y)$ if $P(y, x)$
Functional: $P(x, y) \ \& \ P(x, z)$ implies $y=z$
 $\rightarrow P(x, y) \ \& \ P(y, x)$ implies $x=y$
 \rightarrow not satisfy able

- If you tag a property p as *symmetric* and *inverse functional*, what are the consequences for the property and the connected nodes, if any?

Symmetric: $P(x, y)$ if $P(y, x)$
Inverse Functional: $P(y, x) \ \& \ P(z, x)$ implies $y=z$
 $\rightarrow P(x, y) \ \& \ P(y, x)$ implies $x=y$
 \rightarrow not satisfy able

- If you tag a property p as *transitive* and *functional*, what are the consequences for the property and the connected nodes, if any?

Transitive: $P(x, y) \ \& \ P(y, z)$ implies $P(x, z)$
Functional: $P(x, y) \ \& \ P(x, z)$ implies $y=z$
 $\rightarrow P(x, y) \ \& \ P(y, z)$ implies $y=z$
 \rightarrow There exists only one value for this relation.
 \rightarrow So why to make it transitive then?

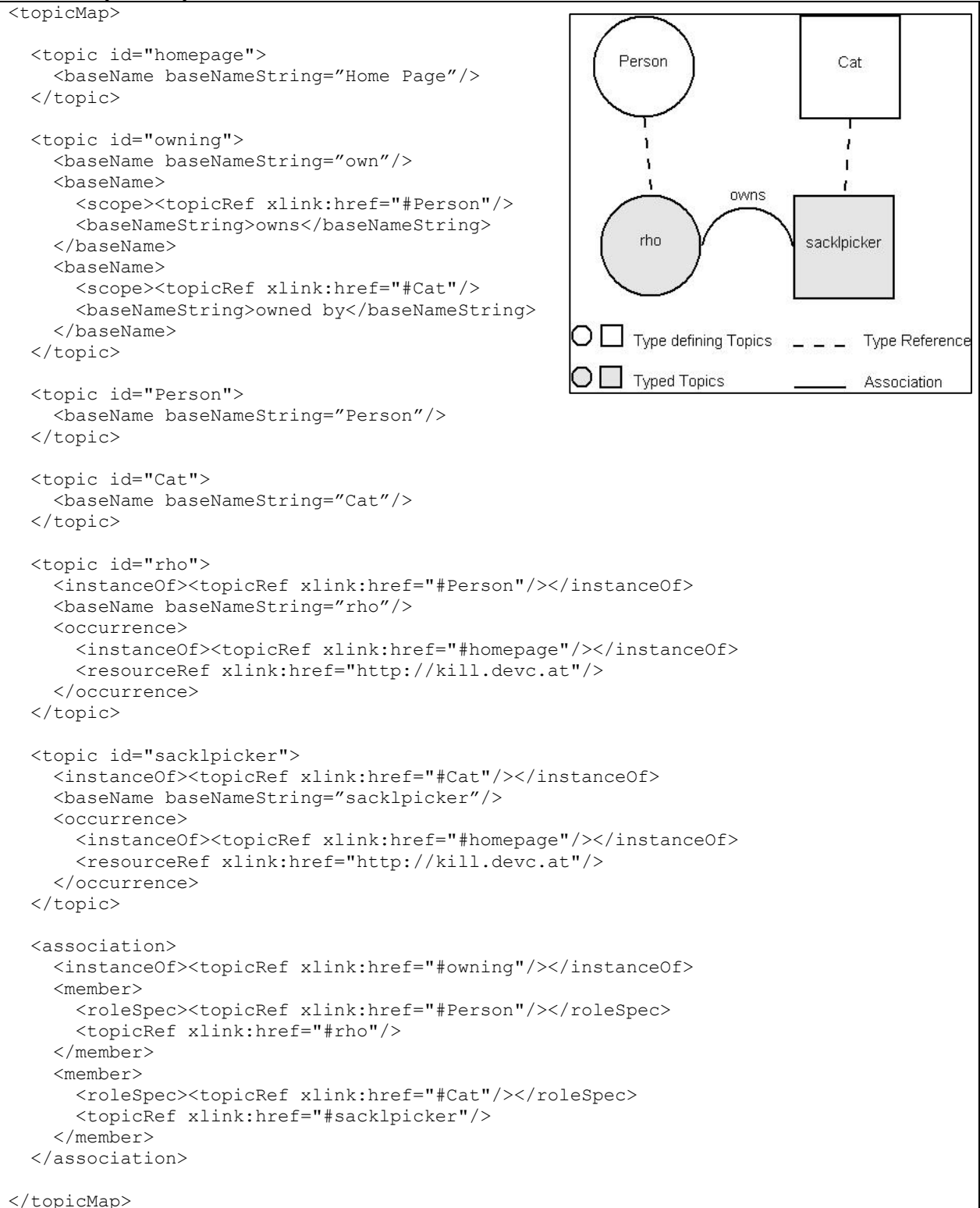
Question 8

(10%)

Given the following RDF triples, how would that translate into a topic map? Draw a diagram and label all nodes and arcs appropriately.

```
:rho isa :Person .
:sacklpicker isa :Cat .
:rho :owns :sacklpicker .
```

Diese Topic-Map ist bestimmt zu ausführlich, aber zum üben nicht schlecht:



Question 9 (12%)

Ad SPARQL:

- The query result is a single boolean value if ASK instead of SELECT is used. Under which circumstances can (should) ASK be preferred over the more generic SELECT?

Applications can use the ASK form to test whether or not a query pattern has a solution.

- You cannot write SELECT ... WHERE {NOT \$p a foaf:Person }. Why not?

Because you cannot negate a Triple?
Because there is no negation in SPARQL?
Because negations in SPARQL can only be combined with PrimaryExpressions?

- Describe a use case for using several data sets.

End of Paper.