

Problem 1:

- a) When is an XML document “valid”? When is an XML document “well-formed”? Is it possible to have a valid and not well-formed XML document? (2 credits)
- b) What is the difference between general and parameter entities? Please give an example for each. (2 credits)
- c) Given the following menu.xml with the following design:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE menu SYSTEM "cook.dtd">
3 <menu>
4   <categories>
5     <category name="Soup"/>
6     <category name="Meat"/>
7   </categories>
8   <items>
9     <item category="Soup" id="soup001">
10      <name>Gulaschsuppe </name>
11      <ingredient name="beef" amount="1" unit="kg"/>
12      <ingredient name="onion" amount="0.5" unit="cup"/>
13      <price currency="EUR">3.00</price>
14    </item>
15    <item category="Meat" id="meat001">
16      <name>Wiener Schnitzel</name>
17      <ingredient name="Chicken" amount="400" unit="g"/>
18      <ingredient name="Egg" amount="3"/>
19      <price currency="EUR">7.00</price>
20    </item>
21    <item category="Soup" id="soup002">
22      <name>Tomatoes Soup </name>
23      <ingredient name="tomatoes" amount="1" unit="kg"/>
24      <ingredient name="onion" amount="0.5" unit="cup"/>
25      <price currency="EUR">2.50</price>
26      <comments>Do not put too much salt.</comments>
27    </item>
28  </items>
29 </menu>
```

Extra remarks:

- Each item of menu has attribute category and an id (attribute type ID).
- An ingredient should contain a name, an amount and could have a unit.
- Currency could be either EUR or DM; the default is set to EUR

- i) Write a DTD for the above XML document. (3 credits)
- ii) Give an XPath expression that selects all id attributes of items containing a “comments” element. (3 credits)

Problem 2:

- a) What do XPath, XQuery and XSLT have in common? What is the main difference between XPath and XSLT w.r.t. their capabilities? (i.e. what can be done with XSLT but not with XPath.) Give an example. (2 credits)
- b) Name two ways to create an element in XSLT? How do these two ways differ? (2 credits)
- c) Refer to the menu.xml in Problem 1c.
- i) Write an XQuery that outputs a list of all items with a price in EUR, sorted by the EUR price in increasing order. (2 credits)
- ii) Write an XSLT that outputs the following XML file. Content of item name (i.e. Gulaschsuppe, Wiener Schnitzel, etc) and item category (i.e. Soup, Meat) should NOT be created by explicitly stating the literal value of the nodes. (4 credits)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Menu>
3   <Soup>
4     <name>Gulaschsuppe </name>
5     <name>Tomatoes Soup </name>
6   </Soup>
7   <Meat>
8     <name>Wiener Schnitzel</name>
9   </Meat>
10 </Menu>
```

Problem 3:

a) What is the difference between the RDF document

```
:tab1 :hasColumnNames "Europe", "Asia", "USA" .
```

and

```
:tab1 :hasColumnNames ( "Europe" "Asia" "USA" ) .
```

Give a triple representation of each of them.

(3 credits)

b) Design a RDF document to capture the following contents:

A company sells cars, e.g., the Avanti 2006, turbines, e.g., the TXX06, and service policies, e.g., the 2-years-guarantee for the Avanti 2006.

Service policies cover only products sold by the company (and not any third-party products).

Can you design your RDF document in such a way that you can infer that the company sells Avanti 2006 without having to state it explicitly? If so, do so and sketch how the inference is done; otherwise argue why not.

Can you express that the company does not sell ships? If so, do so; otherwise argue why not.

(4 credits)

c) In some spreadsheet application, tables like

	Europe	USA	Asia
2004	104	94	54
2005	121	97	65
2006	133	98	87

are captured in RDF documents as follows:

```
1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3 @prefix : <http://www.cgnm.de/spreadsheet#> .
4 :Table rdf:type rdfs:Class .
5 :Row rdf:type rdfs:Class .
6 :hasColumnNames rdf:type rdf:Property ; rdfs:domain :Table ; rdfs:Seq .
7 :hasRow rdf:type rdf:Property ; rdfs:domain :Table ; rdfs:range :Row .
8 :hasRowName rdf:type rdf:Property ; rdfs:domain :Row .
9 :containsData rdf:type rdf:Property ; rdfs:domain :Row ; rdfs:range rdf:Seq .
10
11 :tab1 rdf:type :Table ;
12   :hasColumnNames [ rdf:_1 "Europe"; rdf:_2 "USA"; rdf:_3 "Asia" ] ;
13   :hasRow [ :hasRowName "2004"; :containsData [ rdf:_1 "104"; rdf:_2 "94"; rdf:_3 "54" ] ] ,
14     [ :hasRowName "2005"; :containsData [ rdf:_1 "121"; rdf:_2 "97"; rdf:_3 "65" ] ] ,
15     [ :hasRowName "2006"; :containsData [ rdf:_1 "133"; rdf:_2 "98"; rdf:_3 "87" ] ] .
```

Write a SPARQL query that transposes the table, i.e., interchanges rows and columns to yield the following result: (see next page)

	2004	2005	2006
Europe	104	121	133
USA	94	97	98
Asia	54	65	87

(3 credits)

Problem 4:

- a) State three advantages of using description logic as the basis for an ontology language. (2 credits)
- b) Represent the following knowledge base about beers and breweries in OWL (Abstract or OWL/XML syntax). Indicate clearly which are classes, roles and individuals, indicating any properties, e.g., disjointness, symmetry, functional, cardinality, etc.

Beers are brewed by breweries.

Breweries brew beers.

Beers can have at least 3 ingredients and at most 6.

Water is an ingredient.

A beer must have at least one ingredient which is water.

Zapfler is a beer.

Jeva is a beer.

Becks is a beer.

ABC is a brewer.

XYZ is a brewer.

ABC brews only one brand of beer.

ABC brews Zapfler.

ABC brews Becks.

ABC does not brew the same beers as XYZ.

Are there any inconsistencies in your ontology? If so, justify your answer. (6 credits)

- c) Considering your ontology of the previous question, define a statement saying that all the beers brewed by “XYZ” belong to a category of beers known as “Pilsner”. (2 credits)