

problem 1:

- a) How do XML schema differ from DTD? Given an example to show what can only be done with XML Schema and not with DTD? (2 credits)
- b) Design a XML schema with a target namespace as specific as possible for the following XML document:

```
1 <purchase>
2   <media>
3     <movie number="7890">
4       <name>My fair lady</name>
5       <price>12</price>
6     </movie>
7     <cd number="564">
8       <title>We love music</title>
9       <price>17</price>
10    </cd>
11  </media>
12</purchase>
```

- Use at least one "ref" (refer to previously declared elements)
- Set <http://www.w3.org/2001/XMLSchema> as default namespace

Apply necessary changes to the XML document(if there are any) such that it references the schema and is valid w.r.t. the schema.

(5 credits)

- c) Explain what it means if a schema is valid as oppose to a document being valid w.r.t. schema. Is the following schema valid? Explain why and why not. If invalid, what would be the changes to make it valid?

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:abc="http://cgnm/person"
3 targetNamespace=" http://cgnm/person " elementFormDefault="qualified">
4   <xs:element name="person">
5     <xs:complexType>
6       <xs:sequence>
7         <xs:element name="first-name" type="xs:string"/>
8         <xs:element name="last-name" type="xs:string"/>
9         <xs:element name="middle-initial" type="xs:string" minOccurs="0"/>
10        <xs:any namespace="###targetNamespace" minOccurs="0"
11 maxOccurs="unbounded"/>
12      </xs:sequence>
13      <xs:attribute name="version" type="xs:string"/>
14    </xs:complexType>
15  </xs:element>
16</xs:schema>
```

(3 credits)

problem 2:

a) What is the difference between the two XPath queries

```
//section[1]
```

and

```
/descendant::section[1]
```

Sketch an example XML document for which both queries return different results. Compute these results step by step. Describe the meaning of both queries in plain text. (3 credits)

b) Given is the following topic hierarchy encoded in XML:

```
1 <topics>
2 <computer-science>
3 <information-systems>
4 <data-bases/>
5 <xml-technologies>
6 <xpath/>
7 <xslt/>
8 </xml-technologies>
9 </information-systems>
10 <machine-learning/>
11 </computer-science>
12 <mathematics>
13 <algebra/>
14 </mathematics>
15 </topics>
```

Sketch two XSLT programs that render this topic hierarchy to HTML such that it looks as follows in a browser:

1 1. computer-science	1 1. computer-science
2 1. information-systems	2 1.1. information-systems
3 1. data-bases	3 1.1.1. data-bases
4 2. xml-technologies	4 1.1.2. xml-technologies
5 1. xpath	5 1.1.2.1. xpath
6 2. xslt	6 1.1.2.2. xslt
7 2. machine-learning	7 1.2. machine-learning
8 2. mathematics	8 2. mathematics
9 1. algebra	9 2.1. algebra

(5 credits)

c) Is it possible to call a XPath function from within XSLT?

Assume there is a complex template matching `topics` elements

```
1 <xsl:template match="topics">
2 ...
3 </xsl:template>
```

in a XSLT stylesheet `library.xsl` written by a third-party, so you cannot make changes to it.

Is it possible to use this template from within an XPath expression? Write a XSLT code fragment, that counts the number of result nodes returned by this template by means of the XPath `count` function. (2 credits)

problem 3:

- a) What is understood by "validating" and "static typechecking" in XML processing? Give an example for each technique. How can they be used in XQuery? (3 credits)
- b) Write a XQuery query that transposes the following table:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <table>
3 <row><col>A</col><col>B</col><col>C</col></row>
4 <row><col>1</col><col>2</col><col>3</col></row>
5 <row><col>x</col><col>y</col><col>z</col></row>
6 </table>
```

I.e., it interchanges rows and columns so that the result looks as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <table>
3 <row><col>A</col><col>1</col><col>x</col></row>
4 <row><col>B</col><col>2</col><col>y</col></row>
5 <row><col>C</col><col>3</col><col>z</col></row>
6 </table>
```

Will this query be performed efficiently?

Sketch a XSLT query for this task.

(5 credits)

- c) Given the following DTD:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!ELEMENT uni (student+)>
3 <!ELEMENT student (name, take+)>
4 <!ELEMENT take (grade)>
5 <!ATTLIST take
6 sessionNo IDREF #REQUIRED >
7 <!ELEMENT session (courseno,semester,instructor) >
8 <!ATTLIST session
9 sessionNo ID #REQUIRED >
```

Note: All undeclared elements are considered to be #PCDATA.

Write an XQuery query to find all students who have taken at least two courses with the same instructor. (2 credits)

problem 4:

- a) What representations are commonly used for OWL? Represent the following example in each of them:

married persons are exactly all individuals who have a human partner.

Also shortly survey advantages and disadvantages of each representation. (3 credits)

- b) Represent the following knowledge base in OWL:

Hector is a dog.
Hector's fur is brown.
Hector's eyes are blue.

Write an OWL expression for the class of all possible fur colors of dogs.

Is it possible to express in OWL, that Hector's fur and eyes colors do not match? If yes, write the OWL statement, if no, argue why. Can this statement be inferred from the knowledge base? Explain.

Is it possible in OWL to form the class of all individuals whose fur and eye colors match? If yes, write the OWL statement, if no, is there an additional description logics construct that allows to form this class? (4 credits)

- c) Given the following RDF example

```
1 @prefix xs: <http://www.w3.org/2001/XMLSchema#> .
2 @prefix : <http://www.cgnm.de/rdf/family#> .
3 :Anne :age "45"^^xs:integer; :marriedTo :Bert ; :motherOf :Clara, :Dennis .
4 :Bert :age "49"^^xs:integer; :marriedTo :Anne ; :fatherOf :Clara, :Dennis .
5 :Clara :age "24"^^xs:integer; :marriedTo :Emil; :motherOf :Fred, :Gisa .
6 :Dennis :age "22"^^xs:integer.
7 :Emil :age "27"^^xs:integer; :marriedTo :Clara; :fatherOf :Fred, :Gisa .
8 :Fred :age "2"^^xs:integer.
9 :Gisa :age "1"^^xs:integer.
```

Write a SPARQL query that returns all persons having at least 2 children.

Write a SPARQL query that returns all persons without children.

(3 credits)

- d) OWL allows to specify that an object property is symmetric, functional or inverse-functional, but in the SHOIN(D) calculus we did not have corresponding language constructs. Is OWL consequently more expressive than SHOIN(D) ? (+1 credit)