Two hours

Special instructions:
This paper will be taken on-line and this is the paper format which will be available as a back-up

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Semi-structured Data and the Web

Time: ???:0

Marker’s feedback version

Please answer ALL questions provided.

The exam is CLOSED BOOK.

The exam contains MULTIPLE CHOICE QUESTIONS: each of these questions may have one or more correct answers, and you have to select all correct answers for full marks.

The use of electronic calculators is not permitted.
Answer both questions

1. (35 marks)

a) Describe each of following concepts in one or two sentences:

i) a validating XML parser (2 marks)
ii) the DOM tree or infoset of an XML document (2 marks)
iii) the PSVI of an XML document (2 marks)

Marker’s feedback
all three are bookwork questions, but we look for proper explanations using correct terminology; in particular, for 1.c, we indeed want to see in how far the infoset and the PSVI differ

b) Describe 2 important properties of XML documents that XML schema languages are designed to describe or constrain, and use an example to illustrate these properties.

(4 marks)

Marker’s feedback
following 2 things: — different kinds of statements that we can make in a schema – e.g., what is allowed/expected – with — different properties of an XML document that we can describe in a schema – e.g., the document structure (element names and their nesting – and its content – e.g., the datatype of attribute values. And we were asking for the latter only.

c) Consider the following XML fragment.

```
<name xmlns:mns="white"
     xmlns:yns="pink"
     xmlns="blue">
  <lastname xmlns="green">Smith</lastname>
  <yns:firstname xmlns="white">
    <fullname>John</fullname>
    <mns:nickname xmlns:mns="pink">Jo</mns:nickname>
    <mns:nickname>Yo</mns:nickname>
  </yns:firstname>
</name>
```
i) Is it well-formed? In case it is not, explain why not. In case it is, give, for each element, its universal (or expanded) name. (3 marks)

ii) Comment, in two sentences, on the way namespaces are used in this document: does this seem to be a useful or beneficial way? (2 marks)

**Marker’s feedback**

(ii) often failed to mention that, for a snippet like the one given, the use of namespaces seems over the top and some aren’t used – i.e., there is no benefit in declaring them at all.

d) For the following XQuery, describe the result when answered over the two XML documents nodes.xml and articles.xml given below and explain your answer. (8 marks)

**XQuery:**

```xml
<report>
  {for $sale in doc("sales.xml")//sale,
   $art in doc("articles.xml")/articlelist/article
   where $sale/@on = $art/@id
   return <product id="{$art/@id}"
                   price="{$art/@price}"
                   sold="{$sale/@number}" >
     { $art/description }
   </product>
  }
</report>
```

**articles.xml:**

```xml
<articlelist>
  <article id="1" price="3.00">
    <description>super wheel</description>
    <supplier>t2</supplier>
  </article>
  <article id="2" price="3.50">
    <description>great bike</description>
    <supplier>t1</supplier>
  </article>
  <article id="3" price="3.60">
    <description>nice scooter</description>
    <supplier>t3</supplier>
  </article>
</articlelist>
```
sales.xml:

```xml
<!DOCTYPE sales [
<!ELEMENT sales (sale+)>
<!ELEMENT sale EMPTY>
<!ATTLIST sale
  on CDATA "1"
  number CDATA #IMPLIED> ]>
<sales>
  <sale number="300"/>
  <sale number="2" on="3"/>
  <sale number="22"/>
  <sale on="7" number="300"/>
</sales>
```

Marker’s feedback

spot the default attribute value set in the DTD, thus resulting in too few elements returned; some failed to give any explanation; some failed to return the right structure of elements.

e)  i) How many trees are accepted by the grammar $G = (\{N,M\},\{a,b\},\{N\}, P)$, where $P$ is given below?

\[
P = \{ N \rightarrow a (M,M,M), \\
        M \rightarrow b \varepsilon \}
\]

(2 marks)

- not enough information given to be able to say
- no tree at all
- exactly one tree
- at least two trees, but finitely many
- infinitely many trees

ii) How many XML documents with root element $N$ are valid with respect to the following DTD?

```xml
<!ELEMENT N (B+)>
<!ELEMENT B EMPTY>
```

(2 marks)

- not enough information given to be able to say
- no document at all
- exactly one document
• at least two documents, but finitely many
• infinitely many documents

iii) Which of the following statements are true for the given DTD d.dtd and the XML schema s.xsd? (3 marks)

d.dtd:

```xml
<!ELEMENT a (b,c) >
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA) >
<!ATTLIST a IsNice CDATA #IMPLIED "f">
```

s.xsd:

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="a">
    <xs:complexType>
        <xs:all>
            <xs:element name="b" type="xs:anySimpleType"/>
            <xs:element name="c" type="xs:anySimpleType"/>
        </xs:all>
        <xs:attribute name="IsNice" use="optional"/>
    </xs:complexType>
</xs:element>
</xs:schema>
```

• every document that validates against d.dtd also validates against s.xsd.
• every document that validates against s.xsd also validates against d.dtd.
• there are documents that validate against d.dtd but do not validate against s.xsd.
• there are documents that validate against s.xsd but do not validate against d.dtd.
• none of the above

Marker’s feedback
often answered incorrectly - in particular, students chose contradicting answers such as, e.g., the first and the third.

f) Explain when and why it can be useful to use schemas that correspond to single-type grammars. (5 marks)
Marker’s feedback
half, and we expected students to answer both “why” and “when” (only 8 students got full marks), i.e., we expected a description of a situation where I wanted to use a schema that corresponds to a single-type grammar and an explanation of why using such a grammar in this situation would be beneficial.
2. (35 marks)

a) True or False: An XML Document can be well-formed but not valid with respect to a DTD. (1 mark)

b) Consider the following XML Document:

```xml
<!DOCTYPE books [
<!ELEMENT books (book)+>
<!ELEMENT book (title, author+)>  
<!ELEMENT title (#PCDATA)>    
<!ELEMENT author (#PCDATA)> ]>
<books>
  <book>
    <title>Sister Outsider: Essays & Speeches</title>
    <author>Audre Lorde</author>
  </book>
  <book>
    <title>Pride and Prejudice</title>
    <author>Jane Austen</author>
  </book>
</books>
```

- The document is valid but not well-formed.
- The document is neither well-formed nor valid.
- The document is well-formed but not valid.
- The document is both well-formed and valid.

(2 marks)

c) XML’s draconian error handling (i.e., treating well-formedness errors as fatal)

- conforms with Postel’s law.
- violates Postel’s law.
- conforms to the spirit (i.e., its intention), but violates the letter (i.e., the literal, direct interpretation) of Postel’s law.
- conforms to the letter (i.e., the literal, direct interpretation), but violates the spirit (i.e., its intention) of Postel’s law.

(2 marks)

d) An XML Document with a DOCTYPE declaration
• can be internally valid.
• must be internally valid.
• can be externally validated against another DTD.
• must be externally validated against another DTD.
• can never be externally validated against another DTD.

(2 marks)

e) Briefly (in 4 - 5 sentences) compare the advantages and disadvantages of internal vs. external validation. (4 marks)

Marker’s feedback
Points (up to 4) were given for correct advantages and disadvantages listed. Minor errors in definition or pro/con were tolerated, but major ones could cost points.
Internal validation does not refer to having an internal DTD *subset* i.e. to the physical inclusion in the file. Answers centered on this generally were completely wrong (i.e., 0) though I did give a point for some arguments about physical separation (which is true for external validation and subset).
External validation more easily accommodates using multiple schemas.
Time is not a significant advantage or disadvantage of internal vs. external, esp. due to network concerns (since those apply to external subsets as well). This should be familiar as you all did plenty of external validation in the course work: the files were local, you could configure a one click move, it clearly had no major impact. Since the distinction naturally applies to DTD contexts, namespaces cannot be an issue.
Validation modality does not affect the possible tightness or looseness of the schema. Arbitrary DTDs can be used in either mode. Both modes produce PSVIs.
Some people provided definitions (even somewhat correct ones) without listing advantages and disadvantages. This didn’t answer the question and got 0.

f) True or False: DTD limitations tend to encourage modelling with attributes. (1 mark)

g) Which of the following WXS features alter the PSVI?

• Groups.
• Abstract types.
h) Briefly (in 4 - 5 sentences) explain how XPath can be used as a schema language and the advantages or disadvantages of using it as such.  

```
Marker's feedback

Some people did not understand that the question was asking to consider XPath (a query language, typically used to retrieve parts of a document) could be used as a schema language (i.e., to check the structure). The question did not ask for a comparison of XPath and (e.g.,) XSLT or XQuery. Nor did it ask for a detailed description of XPath. Many people just gave advantages or disadvantages of XPath (e.g., over XQuery or SAX). Where these made sense for XPath as a query language, I tried to give points for that.

Some people claimed that XPath could not handle certain things, such as namespaces or datatypes. This is incorrect, at least in general. XPath functions can be used for type checking of atomics and XPath is certainly namespace sensitive. XPath does not generate a PSVI, of course, and there are other differences. All other things being equal, this cost a point. It is not true that most schema languages use XPath...only Schematron does.

By and large XPath can be less sensitive to changes in the document. That is, it can be used to validate particular fragments of a document even when the rest changes wildly.

Every XPath either succeeds or returns no results. If it succeeds then it has checked that constraint. That was the point of using conditional expressions in class to simulate returning IS VALID.

You do not have to use schematron to use XPath to validate a document. The success or failure of the query is similar to the success or failure of validating against a DTD.
```

i) Which of the following are well-formed XML documents?

```
• <html><TITEL/></html>
• <foo bar="a" baz='a'/>
• <foo bar="a" baz='a'/> <foo bar="a" baz='a'/>
• <html><head><title></head></title><body></body></html>
• <foo />
```
j) Write a DTD that describes the following XML document. Strive for a fairly tight, but sensible DTD. (It should be about 6 declarations long).

```xml
<discussion>
  <topic id="htk" title="Hume’s theory of knowledge">
    <discussants>
      <person nick="bijan"/>
      <person nick="ruth"/>
    </discussants>
  </topic>
</discussion>
```

(6 marks)

**Marker’s feedback**
Rather a lot of people had syntactically correct DTDs that modelled the document accurately. Good job. A high number of the remaining had minor, easily repaired errors and received full marks.

k) True or false: Every XML Document can be (externally) validated against an infinite number of DTDs.

(1 mark)
1) Consider the following XQuery (recall that declaring a global variable external means that it will be set by the calling environment):

```xquery
declare variable $aBool as xs:boolean external;
declare variable $anInt as xs:integer external;
declare variable $str1 as xs:string external;
declare variable $str2 as xs:string external;
if ($aBool) then ($anInt + 1) else concat($str1, $str2)
```

i) This query has type
- `xs:integer`
- `xs:positiveInteger`
- `xs:string`
- `union(xs:string, xs:integer)`
- `union(xs:string, xs:positiveInteger)`

(2 marks)

ii) True or false: Can it be cast to `xs:string`?  

(1 mark)

iii) XQuery’s type system is primarily
- strong, manifest, and nominal
- strong, latent, and structural
- strong, latent, and nominal
- weak, latent, and nominal
- weak, manifest, and structural

(2 marks)

iv) Briefly (in 2-3 sentences) explain why one might create a DTD instead of a WXS for a new project.

(2 marks)

---

**Marker’s feedback**

The key point that most people hit is that DTDs are simpler (easier to read/write, etc.). The desire to inline the schema was accepted, but does not seem that compelling. Rather fewer people pointed out that DTDs can declare entities (a very good reason).

DTDs (except for entities) can be expressed as WXS but not the reverse. There is no reason to think that DTDs are hugely more efficient than the correspondingly expressive WXSs. DTDs are less constraining than WXS. For example, a DTD can not constrain element content to be only integers, or only 3 characters long. Some people gave conditions which permit the use of DTDs (e.g., no namespaces) but did not give a positive reason for using them.