Informatics 1 Data & Analysis
Tutorial 5

Week 7, Semester 2, 2011

- Please answer questions 1–6 of this worksheet in advance of the tutorial, and bring with you all work, including printouts of code and other results. Tutorials cannot function properly unless you do the work in advance.
- Data & Analysis tutorial exercises are not assessed, but are compulsory and an important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Please also think about the **Tutorial Discussion** points.
- Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your tutor, and if possible attend another tutorial group in the same week.
- *Recommended Reading:* Selected sections from `An Introduction to XML and Web Technologies` by *Miller and Schwartzbach* as identified in the lecture slides.

Introduction

In the previous tutorials, you have been working with relational databases. In this tutorial, you will be working with *XML*, the *Extensible Markup Language*, which takes a different approach to storing and querying large bodies of data. You will be working with files *restaurants.xml* and *restaurants.xq* that you can download from the course webpage.

Question 1

Draw the XPath data model (tree structure) for the following xml document:

```
<patrasrestaurants>
     <restaurant name="La Pasteria">
        <address>13 Alonisou Street, Patras, 261 35</address>
        <cuisine>Italian</cuisine>
        <phoneno>2610325833</phoneno>
     </restaurant>
```

```
        <restaurant name="Kalamarakia">
         <address>21 Poseidonos Street, Patras, 264 45</address>
         <cuisine>Greek</cuisine>
         <phoneno>2610428066</phoneno>
      </restaurant>
</patrasrestaurants>
```

Question 2

Design a DTD for an XML database of restaurants. The DTD should be designed so that the document *restaurants.xml* is valid with respect to it.

Question 3

Validate the given *restaurants.xml* file with your DTD file. Y ou can check your XML file for validity and well-formedness using the command `xml-xparse` on a DICE machine. Copy the *restaurants.xml* file and include the document type declaration that specifies your DTD, together in a single file. You will be able to run the validity and well-formedness checks by entering the following command into a terminal window on a DICE machine:

xml-xparse restaurants.xml

Question 4

Write path expressions to extract the following information as a text string from any XML document matching your restaurants DTD. Note: As well as the lecture slides, you might want to consult an XPath tutorial like the one at `http://www.w3schools.com/xpath/`.

(a) Retrieve the addresses of all restaurants.

(b) Retrieve the names of all restaurants.

(c) Retrieve the address of the *Kalpna* restaurant.

(d) Retrieve the phone numbers of all restaurants that serve *Indian* food.

(e) Retrieve the names of all restaurants which serve *Italian* food.

Question 5

The purpose of this question is to execute your path expressions on the file *restaurants.xml* using the Galax query tool available on the DICE machines. Download *restaurants.xq* to your directory and open the file in a text editor. For each of the queries, follow the instructions given below to run Galax. Write down or print your results to bring to the tutorial.

(a) Add your query in the allocated space in the file *restaurants.xq*. You will need to make sure your query correctly references the XML file to be checked: the easiest way is to start paths with `doc("restaurants.xml")/`

(b) Make sure all other queries are commented out (as described in the note below).

(c) Save the file.

(d) Open a terminal window.

(e) Change directory to your current working directory.

(f) Enter the following and hit return: `galax-run restaurants.xq`

Note: Galax can only run one query at a time, so you will need to comment out your other queries, before you are able to run a new query. In general, you should always comment out all your queries, except the one your are currently trying to run. You can comment out a line by enclosing it in mirrored (noseless) smiley-faces. E.g. the following line is commented out.

```
(:  This line is commented out :)
```

Question 6

How would the same data be naturally represented in a relational database? Use the SQL data definition language (DDL) to define an SQL schema for representing the same data.

Tutorial Discussion

(a) What are the strengths and weaknesses of the relational approach versus XML?

(b) Given these strengths and weaknesses, name a context in which you would prefer to use XML.

(c) Given these strengths and weaknesses, name a context in which you would prefer to use a RDB.