XML-Praktikum WS 2016/17

1. Unit: Querying with XPath and XQuery

For the exercises, it is useful to call saxon directly from the shel instead of using the DBIS XQuery Web-Interface, to get more detailed error messages.

Exercise 1.1 (Mondial - Headquarters of Organizations) Solve the first exercises as far as possible by XPath.

a) Give the names of all countries where some organization has its headquarter.

b) Give the names of all countries where no organization has its headquarter.

c) Give the names of all cities that have more than 1000000 inhabitants and where some organization has its headquarter.

d) Give the names of all cities that have more than 1000000 inhabitants and where no organization has its headquarter.

e) Give the names of all cities where an organization has its headquarter, and which are the capital of a member country of this organization.

Exercise 1.2 (Mondial - Country Radius)
The *radius* of a country is defined as the largest distance between the country's capital and anything (city, lake, mountain etc.) that is known to be located in that country.

a) State an XQuery query that returns for every country its radius.
The distance between two pairs `(lat1,long1)` and `(lat2,long2)` can be computed as follows:

```
$dist :=
6370*math:acos(math:cos($lat1 div 180*3.14)*math:cos($lat2 div 180*3.14)*
       math:cos(($long1 - $long2) div 180*3.14)
        + math:sin($lat1 div 180*3.14)*math:sin($lat2 div 180*3.14))
```

b) Sketch how this query must be formulated in SQL against the relational variant of Mondial.

c) For what can the result of this query be useful?

Exercise 1.3 (Mondial - neighbor populations in descending order) Give for each country the sum of the population of its neighbors (in descending order, with those countries with no neighbors coming last).

Exercise 1.4 (Mondial - Lowest Highest Mountain)
Give the lowest mountain which is the highest one on its continent.

Exercise 1.5 (Mondial - Organizations and Continents)
List the names of all organizations with at least one member country on each continent.

Exercise 1.6 (Mondial - Non-Coverable Organizations)
Give the smallest (wrt. number of members) organization O_1 which is not covered by any other organization O_2 (i.e. for all other organizations O_2, O_1 has at least one member which is not a member of O_2).

... continues on the back ...

Exercise 1.7 (Web Data Extraction: Germany-View)

Generate an XML "Germany-View": Use the Web-Resource

`http://www.geohive.com/cntry/germany.aspx`

to create an XML document according to the following DTD:

```
<!ELEMENT country (name,population,provinces,cities)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT population (#PCDATA)>
<!ELEMENT provinces (province*)>
<!ELEMENT cities (city*)>
<!ELEMENT province (name,area,population)>
<!ELEMENT area (#PCDATA)>
<!ELEMENT city (name,population)>

<!ATTLIST country capital IDREF #IMPLIED>
<!ATTLIST province capital IDREF #IMPLIED>
<!ATTLIST city id ID #REQUIRED>
```

Recommendation: copy the XML(HTML) source to your local computer for experimenting with it.

Apply your program then to the page for France.

Exercise 1.8 (Web Data Extraction: Wikipedia)

a) Write an XQuery function that is invoked with the name of a mountain (e.g. "Mont Blanc") that returns a small XML fragment with data about that mountain.

b) Write an XQuery statement that invokes the function for all mountains in Mondial that are located in Germany.

Exercise 1.9 (User-defined Function: Functional Programming – Fibonacci)
This exercise deals with XQuery as a functional programming language. There is nothing about XML in it.

- Write a recursive XQuery function that computes the n-th Fibonacci Number (defined as $fib(n) := fib(n-1) + fib(n-2)$, $\text{fib}(0) := 0$, $\text{fib}(1) := 1$).
- Give the asymptotic complexity of your solution.
- Implement a linear algorithm in XQuery.

Exercise 1.10 (User-defined function: Recursive Network Length)
Write a recursive function that computes the total length of a river system of a given river. Consider that rivers may low into or through lakes.