1. Unit: Exercises to XML

Information about the XML course can be found at
`http://www.stud.informatik.uni-goettingen.de/xml-lecture`

Exercise 1.1 (XML-Tree vs Directory-Tree)

Load `mondial-europe.xml` into *xmllint* and browse through the directory structure. First, change into the `country` element of Germany, then into the city of Göttingen. Then, change into the next city in the document.
Links to *xmllint* and the Mondial database can be found at
`http://www.stud.informatik.uni-goettingen.de/xml-lecture`.

Exercise 1.2 (Student-DTD)

- Write a DTD for XML documents with student data:
  name, address and a student id, one or more subjects (computer science, law, chemistry, sociology etc)
- Write an XML document containing student data conforming to the DTD, and check it for validity using *xmllint*.

Exercise 1.3 (HTML-XHTML)

- Find a simple HTML document (e.g. your own personal student homepage) and convert it by hand from HTML to XHTML.
- Check the XHTML document for validity using the XHTML validator
  (`http://validator.w3.org/detailed.html`).

Hint: In your home directory in the CIP pool, there is a directory `public_html` which is your personal web directory. Files there are accessible via
`http://student.ifi.informatik.uni-goettingen.de/~`<*username*>`/`<*filename*>.

Exercise 1.4 (DFAs and DTDs)

Consider the following DTD:

```
<!ELEMENT date (day,month,year?)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT a (date*)>
<!ELEMENT b (#PDCATA)>
<!ELEMENT c (b+|(b?,a)*)>
```

Define a finite automaton for each element definition which accepts the corresponding *content model*.

Exercise 1.5 (Is XML a context-free language?) Consider XML as a formal language.

(a) is the *language of all XML documents of a given document type*, specified by a DTD that does not contain any attributes context-free?

(b) consider the case where the DTD contains attributes.

(c) is the *language of all well-formed XML documents*, without known document type, or with no document type at all context-free?

Exercise 1.6 (XML Tree and XPath Axes) Consider the XPath axes in a document.

- Provide equivalent characterizations of the “following” axis and of the “preceding” axis
  i) in terms of “preorder” and “postorder”,
  ii) in terms of other axes.

Exercise 1.7 (XML to RDB) A possible model for storing (or indexing) XML data is based on relational tables (we ignore namespaces here).

(1) a table for storing element and text nodes:
   - first column: node identifier in Dewey Notation (e.g., 1.2.6.3 for the third child of the sixth child of the second child of the root node),
   - second column: number of the node when enumerated in *preorder*,
   - third column: number of the node when enumerated in *postorder*,
   - forth column: element type (or “text”),
   - fifth column: text content (or NULL).

(2) a table for storing attribute nodes:
   - first column: dewey identifier of the node where the attribute belongs to,
   - second column: attribute name,
   - third column: value.

a) Discuss whether the above information is sufficient for storing an XML document. Give the tables for a small example document.

b) Discuss what must be done when an update (modification, insertion, deletion) is executed.

c) Given a “current” element somewhere in the tree, characterize the following sets of nodes (i.e., the nodes that result from navigating along the different axes) by their dewey notation and, if possible, by their preorder / postorder information:
   - the parent
   - all children
   - all successors
   - all ancestors
   - all siblings
   - all predecessors according to document order
   - all successors according to document order
   - all attributes