Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

December 1, 2017

XML and Databases — Exercise Sheet 7 —

You only have to submit the parts marked as “Homework Exercises”, i.e. Part c). But please think about the questions in Part a) before the meeting! Send your homework solutions to the instructor via EMail: `brass@informatik.uni-halle.de` (with “`xml17`” in the subject line). The official deadline is December 7, 10:00 (before the lecture time).

Repetition Questions

a) Answer the following questions about XPath:

- What is the purpose of XPath? Name some applications of XPath (e.g., other XML technologies that use XPath).
- What kind of things does XPath compute, i.e. what is the basic data structure of the result (the return type of an XPath evaluator)?
- Name some important components of the XPath context.
- Why does XPath 2.0 have a static and a dynamic context? Explain two advantages of static type checking.
- If a query language like XPath (or relational algebra) has a
  - “core syntax” (in relational algebra only the five basic operations), and a
  - “full syntax” with abbreviations, which can be translated to the “core syntax”,

  what does one have to define in order to specify syntax and semantics completely?
- Where are axes used in XPath? Name some example axes of XPath. Which difference does it make whether an axis is classified as a “forward axis” or a “backward axis”?
- What are the possible node tests? Why does it suffice that one can only specify either the name or the type?
- Why is the node type test for text nodes written “`text()`”, and not simply “`text`”? I.e. why are the parentheses required in this syntax? What would happen if one leaves them out?
- Please name some examples of sequence types in XPath 2.0.

- Explain the syntax of path expressions in XPath. What are the components of a “step” in full syntax, and how are steps combined? What are the most important abbreviations?
- Compare path expressions in XPath with UNIX file paths.

  (It would also be an interesting task to think about file paths in future operating systems, and whether they could be extended with constructs borrowed from XPath.)

  (Also object oriented programming languages such as Java contain path expressions for calling methods of objects or navigating between objects with attributes, if the attributes are public. XPath might be an inspiration for path expressions for ODBMS, or for navigating in JSON structures used in some NoSQL databases.)
- Please explain exactly, how `E1/E2` is evaluated (in a given context).
- Please explain exactly, how `E1[E2]` is evaluated (in a given context).
- What is the “effective boolean value” of a sequence? Where is it used?
- Give an example for a possibly unexpected semantics of an XPath expression (a “surprise” or possible error to be avoided).

In-Class Exercises

b) The following XPath queries refer to an XML data file with data about employees, departments and salary grades. It was derived from an example relational database distributed by Oracle. The data file and its DTD can be downloaded here:

- [http://users.informatik.uni-halle.de/~brass/xml17/empdept.xml]
- [http://users.informatik.uni-halle.de/~brass/xml17/empdept.dtd]

The DTD is:

```
<!ELEMENT EMPDEPT (DEPT*, SALGRADE*)>

<!ELEMENT DEPT (EMP*)>
<!ATTLIST DEPT  DEPTNO   NMTOKEN #REQUIRED
                DNAME    CDATA   #REQUIRED
                LOC      CDATA   #REQUIRED>

<!ELEMENT EMP EMPTY>
<!ATTLIST EMP   EMPNO    NMTOKEN #REQUIRED
                ENAME    CDATA   #REQUIRED
                JOB      CDATA   #REQUIRED
                MGR      NMTOKEN #IMPLIED
                HIREDATE CDATA   #REQUIRED
                SAL      NMTOKEN #REQUIRED
                COMM     NMTOKEN #IMPLIED>

<!ELEMENT SALGRADE EMPTY>
<!ATTLIST SALGRADE
                GRADE    NMTOKEN #REQUIRED
                LOSAL    NMTOKEN #REQUIRED
                HISAL    NMTOKEN #REQUIRED>
```

Please write the following queries in XPath:

- Print the names (ENAME) of all employees in the department in “DALLAS” (LOC).
- Find all data of the employee “FORD” (ENAME).
- What is the job of the employee “SCOTT”?
- Which departments (DNAME) have an employee with job “ANALYST”?
- Who has the president of the company as direct supervisor? (The employee number EMPNO of the direct supervisor is stored in the attribute MGR, “PRESIDENT” is a job.)
- Which department has no employees?

Homework Exercises

c) Please download the following example for XPath queries:

- Data file:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.xml]
- XML Schema definition:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.xsd]
- DTD:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.dtd]

The DTD is shown here, since it is a relatively concise description of the document structure:

```
<!ELEMENT CDDB (composers, cds?, soloists?)>
<!ATTLIST CDDB  xmlns:xsi CDATA #IMPLIED
                xsi:noNamespaceSchemaLocation CDATA #IMPLIED>

<!ELEMENT composers    (composer*)>

<!ELEMENT composer     (pieceOfMusic*)>
<!ATTLIST composer     cno        NMTOKEN #REQUIRED
                       firstName  CDATA   #REQUIRED
                       name       CDATA   #REQUIRED
                       born       NMTOKEN #IMPLIED
                       died       NMTOKEN #IMPLIED>

<!ELEMENT pieceOfMusic (recording*)>
<!ATTLIST pieceOfMusic pno        NMTOKEN #REQUIRED
                       title      CDATA   #REQUIRED
                       key        CDATA   #IMPLIED
                       opus       CDATA   #IMPLIED>

<!ELEMENT recording    EMPTY>
<!ATTLIST recording    rno        NMTOKEN #REQUIRED
                       orchestra  CDATA   #IMPLIED
                       conductor  CDATA   #IMPLIED>

<!ELEMENT cds          (cd*)>

<!ELEMENT cd           (track*)>
<!ATTLIST cd           cdno       NMTOKEN #REQUIRED
                       name       CDATA   #REQUIRED
                       producer   CDATA   #REQUIRED
                       numDiscs   NMTOKEN "1"
                       totalTime  NMTOKEN #IMPLIED>

<!ELEMENT track        EMPTY>
<!ATTLIST track        rno        NMTOKEN #REQUIRED>

<!ELEMENT soloists     (soloist*)>

<!ELEMENT soloist      (performance*)>
<!ATTLIST soloist      name       CDATA   #REQUIRED>

<!ELEMENT performance  EMPTY>
<!ATTLIST performance  rno        NMTOKEN #REQUIRED
                       instrument CDATA   #IMPLIED>
```

Please write the following queries in XPath and test it with an actual XPath implementation. Mention in your homework which system you have used for the practical test and whether there were problems using that system (and possibly how you solved the problems). One possibility is to download the BaseX XML database:

[http://basex.org/home/]

However, there are also online demos like

[http://www.freeformatter.com/xpath-tester.html]

Simple XPath expressions (XPath 1.0) can also be tried with the method described in the lecture using an XSLT-stylesheet and a web browser.

Write the following queries in XPath:

- Select all comment nodes in the entire document.
- Navigate first to an element node with the name "`Recording`", and from there to the last name of the componist of the music piece. Does this query yield duplicates?
- Please find the names (attribute value) of all orchestras that appear in the document.
- Select all music pieces (element nodes) of the composer "`Bach`" in "`E major`".