Homework 3
XPath

Due date: Tuesday, October 16, in class

General instructions

You are asked to write a number of XPath expressions for specific information needs.

You will be using two XML files, `dblp-tiny.xml` and `auctions1.xml` available from the course web page:

http://www.csc.calpoly.edu/∼dekhtyar/560-Fall2007/

The direct URLs for the files are:

http://www.csc.calpoly.edu/∼dekhtyar/560-Fall2007/dblp-tiny.xml

http://www.csc.calpoly.edu/∼dekhtyar/560-Fall2007/auctions1.xml

You shall create a MonetDB database and shred these two files into it. For each query/information need specified below, you must write an XPath expression. You must test your XPath expression and ensure that it produces correct result.[1]

submission instructions

You will submit two shell script files, one for each XML file. In addition, each XPath query you write must be stored in a single file (different queries – in different files). You shell script shall consist of the following commands:

[1]Note, that because you are testing your expressions on only one instance, getting the right results is not a guarantee that your expression is 100% correct/complete.

```
> ~/MonetDB/bin/MapiClient -l xquery q1.xq
> ~/MonetDB/bin/MapiClient -l xquery q2.xq
...
```

Here `''∼/MonetDB/bin/MapiClient"` is a hardcoded path (if you use a different path to your copy of MonetDB, just rename the paths prior to submission, and `q1.xq`, etc. are the names of your files containing XPath queries. The names of the XPath query files can be any of your choosing.

Problem 1: dblp-tiny.xml

First, study the DTD for this file. The DTD is available at

http://www.informatik.uni-trier.de/ ley/db/about/dblp.dtd

A link to this file is available on the class web page.

DBLP is a computer science bibliography site, tracking various research publications in the field. DBLP database consists of a list of entries, each entry describing a single publication. `dblp-tiny.xml` file contains information about three types of publications: master's thesis, a www resource and a journal article. Within each publication record, the meanings of XML elements (such as `<author>, <title>, <year>` etc...) are straightforward.

Represent in XPath, implement and run:

1. Return all journal articles (`<article>` element.
2. Return all authors of journal articles.
3. Return the first author of each journal article.
4. Return titles of all master's theses.
5. Return all articles published after 1992.
6. Return all co-authors (co-editors) of `Jonathan Robie`.
7. Return all publications whose authors or editors are not listed.
8. Return the publication, whose description contains some element with content `''Univ.  of Wisconsin-Madison"`.
9. Return titles of all articles published in volue 33 of `Advances in Computing`.
10. Return journal names for all articles whose `mdate`[2] is `''2002-01-03"`.

[2] Date of last record modification

Problem 2: auctions1.xml

`auctions1.xml` is the smallest size XML file that can be produced by the `xmlgen` utility built for XQuery benchmark XMark. No DTD is available on-line, but an appendix to this assignment includes as tree diagram of the structure of this file.

This XML document is designed to represent data about auctions going on in an on-line auction house. The database consists of the following "top-level" objects:

- Regions of the world.
- Categories: a list of auction/object categories.
- Catgraph: a graph that lists dependencies between different categories.
- Open auctions: a lost of ongoing auctions
- Closed auctions: a list of completed auctions.

Write XPath expressions for the following queries.

1. List all closed auctions.
2. Find the author of the annotation for the second closed auction.
3. Find all regions for which shipping information is available.
4. Find all sale prices for all closed auctions.
5. Find the quantity for each item sold (closed auction) for regular auctions (auction type).
6. Find all open auctions watched by `''Jaak Tempesti"`. (returning the IDREF value is sufficient).
7. Find all elements in the document which have exactly one child.
8. Return the last time a bid was placed on the first open auction.
9. Return all `<parlist>` elements that do not have other `<parlist>` elements as their descendants.
10. For each `''Featured"` closed auction, return the contents of the `text` element inside the auction's annotation.