Exam XML & Databases [192110961]

Wednesday April 13, 2011; 08:45 – 12:15 in CR-3A

Maurice van Keulen and Djoerd Hiemstra

Allowed on exam: slides, reader, print outs, notes on paper

Not allowed on exam: electronic devices

There are 9 questions. For each question, an indication of the associated paper from the reader is given, if appropriate. Moreover, the number of points for each question is mentioned. The points add up to a total of 90 points. You receive 10 points bonus for showing up at the exam. The final grade is determined by dividing the total score by 10.

XML Data

Consider the following example XML document (parsetree.xml). It is a simplified and shortened version of a document produced by the Stanford natural language parser. It has parsed the text “35 km NW of Iguatu (0622\/3918 (USBGN)) (ICWB).”, which is a line from a historical logbook of a ship's travels. The elements are related to “part-of-speech tagging” (POS) that attempts to uncover the intended grammatical structure of a sentence. The elements have the following meanings: truecomma and trueperiod: the symbols ',' and '.'; np: noun phrase; prn: parenthetical; pp: prepositional phrase; cd: cardinal number; nn: noun singular; in: preposition or subordinating conjunction.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE parsetree [
<!ELEMENT parsetree (entry)*>
<!ELEMENT entry (np | truecomma | trueperiod)*>
<!ATTLIST entry id CDATA #REQUIRED>
<!ELEMENT truecomma EMPTY>
<!ELEMENT trueperiod EMPTY>
<!ELEMENT np (np | nn | prn | pp | cd)*>
<!ELEMENT prn (np | truecomma)*>
<!ELEMENT pp (in , np)>
<!ELEMENT cd (#PCDATA)>
<!ELEMENT nn (#PCDATA)>
<!ELEMENT in (#PCDATA)>
<!ATTLIST nn category CDATA #REQUIRED> ]>

<parsetree> <entry id="3">
 <np> <cd>35</cd> <nn category="dist_unit">km</nn> </np>
  <np> <np> <nn category="card_direction">NW</nn> </np>
   <pp>
     <in>of</in>
     <np> <np>
      <np> <nn category="placename">Iguatu</nn> </np>
      <prn>
        <np>
          <np> <cd>0622\/3918</cd> </np>
         <prn> <np> <nn category="institutename">USBGN</nn> </np> </prn>
        </np>
      </prn> </np>
     <prn> <np> <nn category="institutename">ICWB</nn> </np> </prn>
    </np>
   </pp> </np>
  <trueperiod/>
</entry> </parsetree>
```

Question 1 *(10 points)*

[XQuery] Give XQuery queries that are as short as possible for the questions below. Note that the queries need to properly produce answers for any document that is valid according to the embedded DTD, i.e., not only for the given document.

a) *Give all distinct placenames mentioned in the document.*

b) *For every distance indication (an "nn" element of category "dist_unit" immediately following a "cd" element), output the distance measured in meters. In other words, multiply the number by "1" if the unit is "m"; by "10" if the unit is "dm"; and by 1000 if the unit is "km".*

c) *For each institute name, give the number of entries it occurs in.*

Question 2 *(12 points)*

[Paper F/G]

Given the query: `//nn[@category="institutename"][.="ICWB"]/parent::np` producing the noun phrases mentioning the institute "ICWB". There are several variants of the XPath Accelerator storage schema; Adhere to a storage schema with three tables: "doc" with attributes pre/post/level/kind/name, "attr" with attributes pre/value, and "text" with attributes pre/txt.

Follow the XPath Accelerator approach to answer the following question:

a) *Draw the pre/post plane of parsetree.xml. To keep it small, restrict yourself to only the bold part of the XML document.*

b) *Give the corresponding SQL-query to evaluate the above XPath-query. Make sure that it closely adheres to XPath semantics, i.e., the result should be in document order and duplicate-free and for each axis step the appropriate test for node kind is applied (element, attribute or text node).*

c) *Suppose I would delete all "prn" elements. For each of the three tables of the above storage scheme, how many rows are being deleted from that table? Explain your answer, especially how to handle the complication that prn-elements may be nested (such as in the example document)*

Question 3 *(10 points)*

[Paper O] Suppose we have an XML message broker for which users can subscribe to queries on the Stanford parser's activities. The message broker uses the YFilter approach. We have the following three users, each with two subscriptions:

User 1: Q1.1: //np//np; Q1.2: //prn//prn
User 2: Q2.1: /parsetree/entry/trueperiod; Q2.2: //np/cd
User 3: Q3.1: //prn/np/nn; Q3.2: //prn//nn

a) *Draw the YFilter NFA for these queries.*

b) *Which match occurs first if our example parsetree.xml document would stream into the message broker? Explain your answer.*

c) *YFilter uses a stack for its execution and we'd like to calculate the maximal size of the stack. How many entries will the stack maximally have for the given document and queries? And what is the maximum size of one entry on the stack? Explain your answer.*

***Question 4** (10 points)*

[Paper V] It is possible to use stand-off annotation for natural language parsing.

a) *Describe how this can be done. Include an XML-fragment that uses stand-off annotation for parsing the example sentence "35 km NW of Iguatu (0622\3918 (USBGN)) (ICWB)."*

b) *Give two advantages or disadvantages of using stand-off annotation for natural language parsing.*

c) [Paper I] *Does Burkowski's containment model support stand-off annotation? Explain why/why not.*

***Question 5** (10 points)*

[Paper NEW] Consider the storage scheme proposed by Meier for the eXist Database System. Suppose we need to store the *parsetree.xml* document.

a) *How many unique identifiers, including the spare identifiers, would be used by the level-order numbering scheme to store the document. Explain your answer.*

b) *How many unique identifiers, again including the spare identifiers, would be used by the alternating level-order numbering scheme. Explain your answer.*

c) *Compare this approach to the XPath Accelerator storage approach. What are the advantages and disadvantages of level-order numbering?*

***Question 6** (12 points)*

[Paper K, L, M, Y and Z] Query 14 of the XMark benchmark test collection addresses full text querying:

Q 14. *Return the names of all items whose description contains the word 'gold'.*

Answer the following questions.

a) *Give the XQuery statement (without full-text extensions) for this query.*

b) *Give at least three problems with this query. Address the query semantics, the usefulness of the answer, and the efficiency of the query.*

c) *Give the XQuery Full-Text query that solves these problems.*

d) *Is the XMark benchmark well-suited for evaluating full text querying? Explain your answer. If not, what benchmark collection would be more suitable?*

2

***Question 7** (10 points)*

[Paper B/E/F] The paper by Tatarinov et al., distinguishes in *three dimensions of XML order, and several XML order encoding methods.* Explain the following questions:

a) *Which of the three dimensions of order is/are addressed by the stair case join? Explain your answer.*

b) *To which of the XML order encoding methods does pre/post encoding belong? Explain your answer.*

c) *Explain precisely in what case(s) "Inlining" can be used, and which elements can be inlined.*

Question 8 *(10 points)*

Consider the following general questions on XML.

a) *What do the acronyms XML, GML, SGML and HTML stand for? (choose a definition if you think that more than one answer is possible) How are they related to each other? Put them in a timeline (in order of historical appearance).*

b) *How are XPath, XQuery, XUpdate, and XSLT related to each other? Which of these can be used to get all np elements from parsetree.xml, and output each element between the tags <nounphrase> ... </nounphrase>.*

c) *What do the acronyms SAX and DOM, stand for? Mention for each a typical use case. What are disadvantages of using SAX and/or DOM when compared to query languages like XPath or XQuery?*

Question 9 *(6 points)*

[Paper A] Consider the following relational database

CourseCode	*Study*	*CourseName*
211096	CS	XML&DB 1
211086	CS	XML&DB 2
211074	INF	DB

StudentNum	*CourseCode*	*Grade*
123787612	211096	8
987654332	211096	7
444444444	211096	4
123787612	211086	9
444444444	211086	6

StudentNum	*Name*
123787612	Mel
987654332	Kim
444444444	Pete

And the following query:

```
SELECT XMLELEMENT(NAME "Course",
    XMLATTRIBUTES(C.CourseCode, C.Study),
    XMLELEMENT(NAME "Name", C.CourseName),
    XMLELEMENT(NAME "Description", C.Description),
    XMLAGG(NAME "Enrollment",
      XMLELEMENT(NAME "StudentName", S.Name),
      XMLELEMENT(NAME "Grade", E.Grade)
    )
  ) AS StudentRegistration
FROM Courses C, Enrollment E , Student S
WHERE C.CourseCode = E.CourseCode
AND E.StudentNum = S.StudentNum
GROUP BY C.CourseCode, C.Study, C.CourseName
```

Answer the following questions: Name

a) *How many "Course" XML elements, "Student" XML elements and "Enrollment" XML elements does the query produce? Explain your answer.*

b) *How many SQL rows does the query produce, if any? Explain your answer.*