

Please be brief and precise in your responses; you will lose points for wordy or ambiguous answers. You do *not* need to use formal notation in your proofs, although you are welcome to do so; however, your proofs must be presented in a manner that makes it easy to follow your reasoning. You are free to use any print or electronic resources. However, you must submit only text written by you, following the usual norms of quoting and citation when appropriate. You must not discuss this exam with anyone before the time it is due.

1. (20 pts) Describe how XSQ [PC03] evaluates the following XPath query on streaming data: `//cameras/digital[res>=2.0]/camera[manuf="Canon"]`. Exhibit the automaton generated for this query and trace the execution of the automaton on a suitable input of your choice until one data item is first buffered and then sent to the output. What is the shortest sequence of input events (SAX events) that produces output? What is the shortest sequence of input events that produces output that is first buffered? Is the automaton minimal? That is, is it impossible to merge any pair of existing states and retain the query answering functionality? Explain your answer.
2. (20 pts) Consider relations $R(\underline{A}, B)$, $S(\underline{B}, \underline{C}, \underline{D})$, and $T(\underline{C}, \underline{D}, E, \underline{F})$. Suppose attributes A and B are of type `varchar(40)`, C and D are of type `char(8)`, E is of type `float`, and F is of type `integer`. Describe an efficient record and file organization for the relations R , S , and T . Now suppose R , S , and T have cardinalities 1000, 5000, and 10000, respectively. Compute the number of disk blocks used by each relation, assuming a block size of 8 KB. Assume there is a B-tree index for the primary key of each relation. Compute the number of disk blocks used by each index.
3. (20 pts) Consider the following query over the relations of Question 2:

$$\gamma_{A,B, \text{sum}(E) \rightarrow G} \sigma_{F \geq 21 \wedge F \leq 55} (R \bowtie S \bowtie T)$$

List all the subexpressions and orders considered by a System-R-style optimizer. Hint: Pay attention to *interesting orders* and refer to [GMUW02, Section 16.5]. Exhibit the physical query plan that incurs the fewest disk operations, assuming 8000 KB of main memory is available for data. Assume that the only available physical join operators are nested-loop, sort-merge, and hash-join (non-hybrid).

4. (20 pts) Describe an algorithm for deciding containment of SPJRU queries. Provide pseudo-code for the algorithm, explain why it is correct, and analyze its worst-case running time. Include at least two examples: one illustrating containment and the other non-containment. Reminder: Here, as in other questions, you are free to look up prior work on this problem. However, you must cite any work you use and you must describe the algorithm *completely* and in your words. Hint: Refer to the discussion just before Theorem 4.5.2 in [AHV95] and to Section 1.1 of [Ull97].

5. (20 pts) Consider a collection of XHTML documents. (We may think of XHTML as a XML-ized version of HTML.) We assume that we have all the documents on disk and that inter-document links are traversed identically to intra-document links. That is, we use a graph model for XHTML that does not give special treatment to the primary tree structure encoded in the serialization. We wish to replace all elements of the form `c` with `c'` iff the element is reachable from a given document, d . Write a structurally recursive function for this purpose, using the syntax of [ABS99, Section 6.4]. Trace the action of this function on some sample data of your choice, using each of the three interpretations described in [ABS99, Section 6.4.4]. Make sure that your sample data has cycles reachable from a `<a>` element that is reachable from d and that it includes some `<a>` elements not reachable from d .

Submission: Please submit your work electronically as a PDF file named using the scheme `PublicJQ-fin-MNNN.pdf` (replacing PublicJQ with your last name and initials, and NNNN with a 4-digit number) by anonymous FTP upload to `ftp.cs.umd.edu`, directory `/incoming/chaw/724`. Please make sure that the PDF file you upload is viewable using the `gv` program on the `linuxlab` machines.

References

- [ABS99] Serge Abiteboul, Peter Buneman, and Dan Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, first edition, October 1999.
- [AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [GMUW02] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems: The Complete Book*. Prentice-Hall, 2002.
- [PC03] Feng Peng and Sudarshan S. Chawathe. XPath queries on streaming data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, San Diego, California, June 2003. To appear. Available at <http://www.cs.umd.edu/~pengfeng/>.
- [U1197] Jeffrey D. Ullman. Information integration using logical views. In *Proceedings of the International Conference on Database Theory*, 1997.