CSE 322: Formal Models in Computer Science Winter 2005

PROBLEM SET 1

Due Friday, January 14, 2005, in class

Reminder: If you haven't done so already, subscribe to CSE 322 mailing list ASAP by following the link off the course webpage (which is at `http://www.cs.washington.edu/322`).

Reading assignment: Sipser's book, sections 1.1 and 1.2; you should have already read Chapter 0.

Instructions: Information on the collaboration policy and honor code in solving problem sets can be found off the course webpage — **please be sure to read it carefully!**. In a nutshell, you are allowed to collaborate with fellow students taking the class to the extent of discussing solution ideas, *provided you think about each problem on your own for at least 30 minutes*. You must write down solutions on your own, and must clearly acknowledge each person with whom you discussed the solutions. You are expected to refrain from looking up solutions from web-sites, pre-existing sources from prior offerings of this course at UW or similar courses at other schools, or other literature.

This problem set has **FIVE** questions. Each question is worth 10 points unless indicated otherwise. Please be as clear as possible in your arguments and answers. Poorly written solutions, even if more or less correct, will be penalized.

---

1. (Exercise to hone your induction skills) The reversal of a string w, denoted by w^R, is the string "spelled backwards": for example, $reverse^R = esrever$.
   (a) Give a formal definition of the reverse of a string over alphabet Σ using induction on the length of the string.
   (b) Use the above definition to give a formal proof by induction that for all strings $x, y \in \Sigma^*$, $(xy)^R = y^R x^R$. (Here by xy we mean the concatenation of x and y, i.e., x followed by y.)
2. Let A, B, C be languages over an alphabet Σ.
   (a) Prove that $A \cdot (B \cup C) = A \cdot B \cup A \cdot C$.
   (b) Give an example of languages A, B, and C over some alphabet Σ such that $A \cdot (B \cap C)$ does not equal $A \cdot B \cap A \cdot C$.
3. Sipser's textbook, Exercise 1.3 (page 84)
4. (20 points; 5 points each part) Give state-diagrams for (deterministic) finite automata accepting each of the following languages over the alphabet $\{0, 1\}$. Also, give a formal description of the DFA for L_2.
   (a) $L_1 = \{w \mid w \in \{0,1\}^*,\ w$ has an even number of 0's and the number of 1's in w is a multiple of 3$\}$.
   (b) $L_2 = \{w \mid w \in \{0,1\}^*,\ w$ begins with 1, and which, interpreted as the binary representation of an integer, is divisible by 5$\}$.

(c) $L_3 = \{w \mid w \in \{0,1\}^*,\ w \text{ does } \textbf{not} \text{ have } 010 \text{ as a substring}\}$.

(d) $L_4 = \{w \mid w \in \{0,1\}^*,\ w \text{ has at least 3 symbols and ends with } 010\}$.

5. Does every regular language L have a deterministic finite automaton (DFA) D recognizing it such that D has exactly one accept state? Why, or why not? (If your answer is Yes, you must prove that for every DFA there is an equivalent one recognizing the same language that has exactly one accept state, and if your answer is No, you must present a regular language, presumably by giving a DFA for it, and *prove* that **no** other DFA could possibly recognize this language if constrained to have exactly one accept state.)