CSE 322

Intro to Formal Models in CS

Homework #4 (rev. e)

Due: Friday, 29 Jan

W. L. Ruzzo 22 Jan 10

Reminder: **Midterm**, Wednesday 2/3/10, and consequent Modified Late Policy: this assignment will *not* be accepted after 5:00PM Monday, 2/1.
Again three separate, stapled, turn-in bundles, with your name on each please: Problem(s) 1–4 in one, problem(s) 5–6 in another and problem(s) 7–8 in the third. Text problems below are on pages 83-93 of Sipser, *US second edition*; see online scanned versions if you don't have it.

1. Give regular expressions for each of the languages in exercise 1.6 a-f .
2. 1.19
3. 1.20
4. 1.22
5. 1.21. For definiteness and ease of grading, eliminate states in *de*creasing order of state number. Give the full (unoptimized) expressions that arise during intermediate steps. If desired, you may show optimized versions and carry them forward into subsequent steps, but *clearly* label these changes, and I recommend you use only the simplest optimizations. E.g., "simplifying $\emptyset \cup a$ to a we have..."
6. Give two finite sets L having the property that $L = L^2$. ($L^2 = L \circ L$, i.e., language concatenation.) Extra Credit: Give a third, or prove that no others exist. What can you say about finite languages satisfying $L = L^*$?
7. 1.51
8. Suppose L is regular and M is a DFA recognizing L.
   (a) Following the definition in 1.51, show that if two strings x and y are distinguishable by L, then the state reached my M after reading x must be different from the state reached after reading y
   (b) Prove: if there are k strings $\{x_1, \ldots, x_k\}$ every pair of which are distinguishable with respect to L, then M has at least k states. [Hint: pigeon hole principle.]
   (c) The point of the above is that it sometimes gives us a way to prove *impossibility* results — since the statements above apply to *any* DFA for L, they might allow is to prove that it is impossible for a DFA with a small number of states to recognize L. Use these ideas to prove that *any* DFA recognizing the language in Example 1.30 (3rd letter from the right end is 1) must have at least 8 states. Thus the machine in Fig. 1.32 is optimal. (Note that it is not sufficient to argue that various potential "optimizations" to fig 1.32 don't succeed in reducing its number of states, since in principle the best machine might look radically different, and "tweaking" 1.32 isn't guaranteed to find it.)
   (d) Extra Credit. Generalize part (c) to show that the "powerset construction" is nearly optimal, in the sense that for each $n \geq 3$, there is a language recognized by an $n + 1$ state NFA, but by no DFA with fewer than 2^n states.