CSE 322
Intro to Formal Models in CS
Homework #3 (Rev b)
Due: Friday, 22 Jan

W. L. Ruzzo 15 Jan 10

Three separate, stapled, turn-in bundles this week, with you name on each please: Problem(s) 1–4 in one, problem(s) 5–6 in another and problem(s) 7–8 in the third.

Note on text book editions: Problem numbers/pages are from the *US second edition* of Sipser. First/other edition users: proceed at your own risk; there may be some critical differences.

Problems below are on pages 84-89.

1. 1.7bc. (1st ed.: 1.5bc)
2. 1.8a. (1st ed.: 1.6a)
3. 1.9a. (1st ed.: 1.7a)
4. 1.10c. (1st ed.: 1.8c)
5. 1.14(b). (1st ed.: 1.10) I did part (a) in lecture 3, and briefly discussed part (b) in lecture 6, but it's worth writing out carefully. Make your example as simple as possible.
6. 1.38 (1st ed.: 1.31, but more clearly worded in 2nd ed). The components of the 5-tuple are defined exactly as in ordinary NFAs, but the definition of "acceptance" is different. For simplicity, if desired, you may assume that an "all-NFA" has no ϵ edges. You are to show that a language is regular if and only if it is recognized by some all-NFA.

   Extra Credit: Give a regular language L recognized by some all-NFA, say with n states, for which the smallest NFA acccepting L seems to require many more than n states.

   Extra extra credit: can you prove it?
7. 1.16. (1st ed.: 1.12) Show all states, transitions, etc., as specified by the construction, i.e., don't use shortcuts or "optimize" it.
8. For languages $A, B \subseteq \Sigma^*$, define $\text{SHUFFLE}(A, B)$ to be the set

   $$\{w \mid w = a_1 b_1 a_2 b_2 \cdots a_k b_k \text{ where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ with each } a_i, b_i \in \Sigma^*\}.$$

   Show that the regular languages are closed under shuffle. Give both a short, convincing, one paragraph "proof idea" similar to those in the text, and a formal proof. Hint: A variant of the "Cartesian product" construction in Theorem 1.25 may be useful. And, yes, "induction is your friend."

   Note: Read the definition carefully. It says "$a_1 \cdots a_k \in A$," not "$a_1, \ldots, a_k \in A$"; the later specifies k strings, each individually in A; the former specifies k strings, perhaps none in A, whose concatenation (in order) is a single string in A.

Example: if $A =$a*b and $B =$ even parity, then shuffle(A, B) includes strings like `aab0110` and `a01ab10` and `0a1a1b0` and `0110aab` (but not `ab00ab`). All 4 examples could be expressed using $k = 8$ and half of $a_i, b_i = \epsilon$. Alternatively, the 1st can be expressed using $k = 1$, and no ϵ's, the fourth with $k = 2$ and 2 ϵ's, etc.