CSE 322

Intro to Formal Models in CS

Homework #3

Due: Friday, 22 Oct; and *NOT* accepted after noon Tuesday 10/26

W. L. Ruzzo 15 Oct 10

As usual, three separate, stapled, turn-in bundles this week, with your name on each please: Problem(s) 1–3 in one, problem(s) 4–6 in another and problem(s) 7 (plus 8, if you do it) in the third.

Note on text book editions: Problem numbers/pages are from the *US second edition* of Sipser. If you have a different edition, consult the scanned versions online on the course web site. See very early class email for password.

Problems below are on pages 84-89.

1. 1.7bc.
2. 1.8a.
3. 1.9a.
4. 1.10c.
5. 1.14(b).
6. 1.16. Show all states, transitions, etc., as specified by the construction, i.e., don't use shortcuts or "optimize" it.
7. Give a correctness proof for the "closure under concatenation" construction given in Theorem 1.47. Give both a short, convincing, one paragraph "proof idea" similar to those in the text, and a more formal proof. For the later, give a proof roughly in the style and level of detail given for closure under star in lecture and on NFA slides 34-37. But note that the version on the slides is a little more terse than desired, just due to the limited space available on a slide, but your proof need not be more than, say, twice as long.
8. (Extra Credit) 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 $A = \texttt{a}^*\texttt{b}$ 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.