CSE 322
Intro to Formal Models in CS
Homework #3
Due: Friday 19 Oct 07

W. L. Ruzzo 12 Oct 07

Note on collaboration: on this and all homework assignments, you are encouraged to talk to your classmates about the problems, brainstorm, trade ideas, but you are *not* allowed to carry away written notes from these discussions, nor are you allowed to use or borrow from others' written solutions to the problems. This means searching the internet, your friends' old course files, etc. are *not* allowed. Violation of these rules will be treated as academic misconduct.

Note on text book editions: Problem numbers and pages here and in future assignments are from the *second edition* of Sipser. First edition users: proceed at your own risk. Where possible, I'll indicate the correspondence to the 1st edition. Based on preliminary scanning, the contents of the two editions look reasonably similar, but I can't promise that there won't be some critical differences.

Problems below are on pages 84-89.

1. 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?

2. For the language denoted by each of the following regular expressions, give two strings that are members and two strings that are non-members—a total of four strings for each part. Assume the alphabet $\Sigma = \{a, b\}$ in all parts.

   (a) a^*b^*
   (b) $a(ab)^*b$
   (c) $a^* \cup b^*$
   (d) $(aaa)^*$
   (e) $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$
   (f) $aba \cup bab$
   (g) $(\epsilon \cup a)b$
   (h) $(a \cup ba \cup bb)\Sigma^*$

3. Give regular expressions for each of the languages in exercise 1.6 a-f (1st ed.: 1.4 a-f).

4. 1.21 (1st ed.: 1.16).

5. 1.42 (1st ed.: not present).

   Note: Don't misread this. It says "$a_1 \ldots a_k \in A$," not "$a_1, \ldots, a_k \in A$"; the later specifies k (possibly different) 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 $A = \mathtt{a}^*\mathtt{b}$ 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 lots 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.