Computer Science B38 November 25, 1998

Homework Assignment #5
Due: Wednesday, December 9, 1998, in tutorial

On the cover page of your assignment, you must write ***and sign*** *the following statement: "I have read and understood the policy on collaboration in homework stated in the Course Information handout." Without such a signed statement your homework will not be marked. You must also indicate your tutorial section.*

---

1. (10 points) Let R and S be arbitrary regular expressions. For each of the following assertions, state whether it is true or false, and justify your answer.

(a) $(R+S)^* \equiv (R^*S^*)^*$.

(b) if $R^* \equiv S^*$ then $R \equiv S$.

(c) $(R+S)^*R \equiv ((\epsilon + S)R)^*$.

2. (10 points) A regular expression is called $\emptyset$*-free* if it does not contain the regular expression $\emptyset$ as a subexpression. More precisely, the set of $\emptyset$-free regular expressions is defined inductively just as the set of regular expressions (see Definition 5.6, page 152 in the notes) except that in the basis of the definition we omit the expression $\emptyset$. Prove that any regular expression is equivalent either to an $\emptyset$-free regular expression, or to the regular expression $\emptyset$.
(Hint: Use structural induction on regular expressions.)

3. (10 points) For any $x \in \{0,1\}^*$, let $\mathbf{int}(x)$ be the integer whose binary representation is x. For example, $\mathbf{int}(011010) = 26$, $\mathbf{int}(1011) = 11$, and $\mathbf{int}(1100) = 12$. Draw a DFSA M that accepts

$$L_1 = \{x \in \{0,1\}^* : \mathbf{int}(x) \bmod 3 = 0\}$$

Prove that your DFSA accepts this language.
(Hint: For any $x \in \{0,1\}^*$ and any $a \in \{0,1\}$, express $\mathbf{int}(xa)$ in terms of $\mathbf{int}(x)$.)

4. (EXTRA CREDIT — worth 5 points) Draw a DFSA that accepts

$$L_2 = \{x \in \{0,1\}^* : x \text{ does not contain either of the substrings } 010 \text{ or } 110\}$$

Prove that your DFSA accepts this language.

P.T.O.

5. (10 points) Let $L_3 = \{x \in \{0,1\}^* : \; x = y1z, \text{for some } y, z \in \{0,1\}^* \text{ s.t. } |z| = 3\}$. That is, L_3 consists of all strings with at least 4 symbols, where the 4th symbol from the end is 1.

(a) (2 points) Give a regular expression for L_3.

(b) (2 points) Give a nondeterministic FSA that accepts L_3.

(c) (6 points) Apply the subset construction on the NFSA from part (b) to get a deterministic FSA that accepts L_3.

(d) (EXTRA CREDIT — worth 5 points) Prove that *every* deterministic FSA that accepts L_3 must have at least 16 states.

(Hint: Prove that in any deterministic FSA that accepts L_3, distinct input strings of length 4 must lead (from the initial state) to *distinct* states.)

6. (10 points) For any language L over alphabet Σ, define the following operation:

$$\mathbf{min}(L) = \{x \in L : \text{ there are no } y \in L,\; z \in \Sigma^* - \{\epsilon\} \text{ such that } x = yz\}$$

In other words, $\mathbf{min}(L)$ consists of every string in L no proper prefix of which is in L. Prove that if L is accepted by a FSA, then so is $\mathbf{min}(L)$.

THAT'S IT WITH HOMEWORKS, FOLKS!