University of Washington
CSE 322: Introduction to Formal Models in Computer Science
Homework #3
Due: Friday, April 26, 2002, 10:30am

Spring 2002 April 17, 2002

Written homework is due at the *beginning* of class on the day specified. Any homework turned in after the deadline will be considered late. **Late homework policy:** You may turn in your homework after the deadline and before 5pm on the day it was due, but at a cost of a **20% penalty**. No homework will be accepted after 5pm on the due date.

Please staple all of your pages together (and order them according to the order of the problems below) and have your name on each page, just in case the pages get separated. Write legibly (or type) and organize your answers in a way that is easy to read. Neatness counts!

For each problem, make sure you have acknowledged all persons with whom you worked. Even though you are encouraged to work together on problems, the work you turn in is expected to be your own. When in doubt, invoke the *Gilligan's Island* rule (see the course organization handout) or ask the instructor.

* * *

Regular problems (to be turned in) **:**

1. Here is a DFA that accepts $L = \{ w \in \{0, 1\}^* \mid$ the 8th symbol from the right end of the string w is a 1$\}$.

   $M = (\mathrm{Q}, \{0,1\}, \delta, s, \mathrm{F})$ where:

   $\mathrm{Q} = \{q_w \mid w \in \{0, 1\}^* \text{ and } |w| = 8\}$,
   $s = q_{00000000}$,
   $\mathrm{F} = \{ q_w \mid w = 1x \text{ , where } |x| = 7\}$,
   $\delta(q_{a1\ a2\ a3\ a4\ a5\ a6\ a7\ a8},\ a) = q_{a2\ a3\ a4\ a5\ a6\ a7\ a8\ a}$, where $a \in \{0, 1\}$ and $a_i \in \{0, 1\}$,
   for all $1 \le i \le 8$.

   Each state is represented by a string of length 8, which keeps track of what the last 8 symbols the DFA has consumed. (Note that this DFA is defined slightly differently than in the solution set to Homework #2, but it accepts the same language.)

   Prove that $L(M) = L$. To do this, you need to prove by induction on the length of the string w that $(s, w) \vdash^*_{\mathrm{M}} (q_y, \varepsilon)$, where y is the last eight symbols of w, or if $|w| < 8$, then $y = 0^k w$, where $k = 8 - |w|$.

2. Consider the languages mentioned in Exercise 1.4, parts b, e, and i (page 84 in Sipser). For each language, give a regular expression that represents that language. Note that for part b, the 1s can be anywhere in the string. For part i, remember that

we index strings starting at 1, so that a string $w = w_1 w_2 w_3 \ldots w_n$, where $w_i \in \{0, 1\}$ for $1 \leq i \leq n$.

You may use any of the short cuts we have mentioned in class or that the book uses to refer to regular expressions, but please be sure you know what the actual regular expression is, according to the formal definition given in class and in the book.

3. Consider the following regular expression over the alphabet {0, 1}.

   $$(((\varepsilon \cup 0)\ 01 \cup 1^*)\ 0)^*$$

   Use the procedure described in Lemma 1.29 to convert it into a state diagram of an NFA that accepts the language that the regular expression represents. Do not skip steps or simplify your automaton. In other words, everyone who follows the procedure correctly should come up with exactly the same state diagram. If you are worried that you did one of the steps incorrectly, show one or more intermediate steps in the procedure.

4. Exercise 1.16(b). Do not skip steps or simplify your regular expression.

5. If w is a string, we define w^R to be the *reverse* of the string. That is, if $w = w_1 w_2 \ldots w_{n-1} w_n$, then $w^R = w_n w_{n-1} \ldots w_2 w_1$. Suppose L is a language. We define $L^R = \{ w^R \mid w \in L\}$.

   Now suppose r is a regular expression such that $L(r) = L$. Define a function f that takes a regular expression r and produces a regular expression such that $L(f(r)) = L^R$. Briefly justify why your procedure works. (Hint: you want to take advantage of the fact that regular expressions are defined inductively, much as the proof of Lemma 1.29 does.)

   Note: This shows that regular languages are closed under the reverse operation.

* * *

Bonus Problem (optional):

1. A regular expression is in *disjunctive normal form* if it is of the form $(r_1 \cup r_2 \cup \ldots \cup r_n)$ for some $n \geq 1$, where none of the r_i contains an occurrence of $\cup$. Show that every regular language is represented by some regular expression in disjunctive normal form. Hint: $\{a, b\}^* = \{a\}^* \bullet (\{b\} \bullet \{a\}^*)^*$, where $\bullet$ is the concatenation operation. Be sure your construction works if the alphabet Σ has more than two symbols in it.

* * *

Suggested problems (highly recommended, but not to be turned in) **:**

1. Come up with regular expressions that represent the other languages in Exercise 1.4 in Sipser. Note that although such regular expressions must exist, it is not easy to come up with them. For example, part f and part h seem particularly difficult without resorting to the procedure in Lemma 1.32.
2. Exercise 1.14. More practice with converting regular expressions to NFAs.
3. Exercise 1.16a. Good warmup problem.