This lab gives practice at constructing DFAs.

1. Design the following DFAs assuming that the alphabet is $\{0, 1\}$.

   (a) A DFA for $\{w \mid |w| \text{ is odd}\}$.

   (b) A DFA for $\{w \mid \text{every prefix } x \text{ of } w \text{ has } |\#_0(x) - \#_1(x)| \leq 2\}$. Here, $\#_0(y)$ and $\#_1(y)$ are the number of 0's and 1's respectively in the string y.

   (c) A DFA $M = (Q, \Sigma, \delta, q_0, F)$ for the intersection of the previous two languages. Specify each element of the tuple precisely. Do not draw any pictures. Label the states reasonably.

2. Design a DFA that accepts all strings over the alphabet {$,¢,0,1,2,3,4,5,6,7,8,9,.} that correspond to valid currency amounts. A valid string is either

   - a dollar sign followed by a number which has no leading 0's (unless the number is a single 0 by itself), optionally followed by a decimal point and exactly two decimal digits, OR
   - a one or two-digit number with no leading 0's (unless the number is a single 0 by itself) followed by the cent sign ¢.

   Thus, $432.63, $0, $0.02, $0.00, 47¢, 2¢, 0¢are all accepted, but $021, $4.3, $8.63¢, $0.0, $.02, 02¢, 00¢ are not accepted.

3. **To think at home:** Here are some more DFA construction exercises.

   (a) i. $(0 + 1)^*$
   ii. $\emptyset$
   iii. $\{\epsilon\}$

   (b) Every string except **000**.

   (c) All strings containing the substring **000**.

   (d) All strings *not* containing the substring **000**.

   (e) All strings in which the reverse of the string is the binary representation of a integer divisible by 3.

   (f) All strings w such that *in every prefix of* w, the number of **0**s and **1**s differ by at most 2.

4. **To think at home:** Given two regular expressions r and s we write $r = s$ if $L(r) = L(s)$. Which of the following are true?

   - $(0 + 1)^* = 0^* + 1^*$
   - $(01 + 0)^*0 = 0(10 + 0)^*$
   - $1(01 + 1)^*0 = 11^*0(11^*0)^* = (1^+0)^+$