1.21 Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expressions.

![Diagrams of finite automata](a) ![Diagrams of finite automata](b)

1.47 Let $\Sigma = \{1, \#\}$ and let

$$Y = \{ w | w = x_1\#x_2\#\cdots\#x_k \text{ for } k \geq 0, \text{ each } x_i \in 1^*, \text{ and } x_i \neq x_j \text{ for } i \neq j \}.$$ 

Prove that $Y$ is not regular.

In Either 1.47 (above) or 1.53 (below), AT YOUR CHOOSING, YOU MUST USE THE MYHILL-NERODE THEOREM TO PROVE NON-REGULARITY, while in the other you must use the pumping lemma.

1.53 Let $\Sigma = \{0, 1, +, =\}$ and

$$ADD = \{ x=y+z | x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z \}.$$ 

Show that $ADD$ is not regular.

1.54 Consider the language $F = \{ a^i b^j c^k | i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k \}$.

a. Show that $F$ is not regular.

b. Show that $F$ acts like a regular language in the pumping lemma. In other words, give a pumping length $p$ and demonstrate that $F$ satisfies the three conditions of the pumping lemma for this value of $p$. 
1.60 Let \( \Sigma = \{a, b\} \). For each \( k \geq 1 \), let \( C_k \) be the language consisting of all strings that contain an \( a \) exactly \( k \) places from the right-hand end. Thus \( C_k = \Sigma^* a \Sigma^{k-1} \). Describe an NFA with \( k + 1 \) states that recognizes \( C_k \), both in terms of a state diagram and a formal description.

1.61 Consider the languages \( C_k \) defined in Problem 1.60. Prove that for each \( k \), no DFA can recognize \( C_k \) with fewer than \( 2^k \) states.

1.64 Let \( N \) be an NFA with \( k \) states that recognizes some language \( A \).

a. Show that, if \( A \) is nonempty, \( A \) contains some string of length at most \( k \).

b. Show that, by giving an example, that part (a) is not necessarily true if you replace both \( A \)'s by \( \overline{A} \).

c. Show that, if \( \overline{A} \) is nonempty, \( \overline{A} \) contains some string of length at most \( 2^k \).

d. Show that the bound given in part (b) is nearly tight; that is, for each \( k \), demonstrate an NFA recognizing a language \( A_k \) where \( A_k \) is nonempty and where \( A_k \)'s shortest member strings are of length exponential in \( k \). Come as close to the bound in (b) as you can.

1.998 Consider the (decimal) languages defined below. For each one, either give a regular expression for its elements or prove the language is non-regular:

In all examples, a number cannot start with a 0 (unless it is 0 itself) and the empty string is NOT a number:

a) \( L_a = \{ w \mid \text{as an integer } w \text{ is a multiple of 50} \} \).

b) \( L_b = \{ w \mid \text{as an integer } w \text{ is a multiple of 40} \} \).

c) \( L_c = \{ w \mid \text{as an integer } w \text{ is a power of 10} \} \).

d) \( L_d = \{ w \mid \text{as an integer } w \text{ is a multiple of 6} \} \).

e) \( L_e = \{ w \mid \text{as an integer } w \text{ is s. t. the sum of its digits is a multiple of 10} \} \).

f) \( L_f = \{ w \mid \text{as an integer } w \text{ is a power of 2} \} \).

g) \( L_g = \{ w \mid w \text{ is a rational number} \} \).

( with \( \Sigma = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, / \} \) ) 

 Examples of such strings are -76403/3300, or 100/100 but not 1/0 or -0/0.

h) \( L_h = \{ w \mid w \text{ is a relatively prime rational number} \} \).

( with \( \Sigma = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, / \} \) )

 Examples of such strings are -76403/1117, or 100/101 but not 5/10 or 2/4.

1.999 Exhibit and explain an algorithm to find the shortest regular expression \( R_{\text{min}} \) equivalent to a given regular expression \( R \).

( Shortest means least number of symbols and operators. The alphabet is \( \{0,1\} \). )