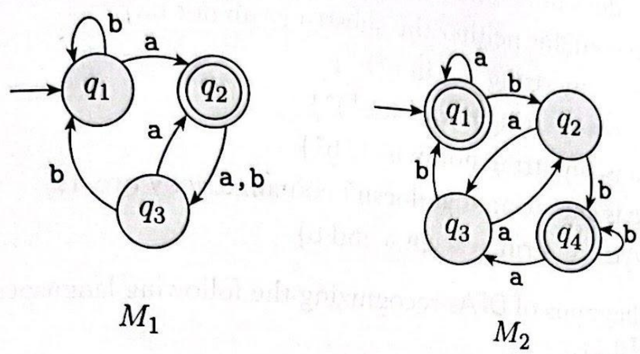
EXERCISES

[A]1.1 The following are the state diagrams of two DFAs, M_1 and M_2. Answer the following questions about each of these machines.



a. What is the start state?
b. What is the set of accept states?
c. What sequence of states does the machine go through on input aabb?
d. Does the machine accept the string aabb?
e. Does the machine accept the string ε?

[A]1.2 Give the formal description of the machines M_1 and M_2 pictured in Exercise 1.1.

1.3 The formal description of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{\text{u}, \text{d}\}, \delta, q_3, \{q_3\})$, where δ is given by the following table. Give the state diagram of this machine.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

1.4 Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{\text{a}, \text{b}\}$.

a. $\{w|\ w$ has at least three a's and at least two b's$\}$
[A]b. $\{w|\ w$ has exactly two a's and at least two b's$\}$
c. $\{w|\ w$ has an even number of a's and one or two b's$\}$
[A]d. $\{w|\ w$ has an even number of a's and each a is followed by at least one b$\}$
e. $\{w|\ w$ starts with an a and has at most one b$\}$
f. $\{w|\ w$ has an odd number of a's and ends with a b$\}$
g. $\{w|\ w$ has even length and an odd number of a's$\}$

1.5 Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{a, b\}$.

- [A]a. $\{w \mid w$ does not contain the substring ab$\}$
- [A]b. $\{w \mid w$ does not contain the substring baba$\}$
- c. $\{w \mid w$ contains neither the substrings ab nor ba$\}$
- d. $\{w \mid w$ is any string not in $a^*b^*\}$
- e. $\{w \mid w$ is any string not in $(ab^+)^*\}$
- f. $\{w \mid w$ is any string not in $a^* \cup b^*\}$
- g. $\{w \mid w$ is any string that doesn't contain exactly two a's$\}$
- h. $\{w \mid w$ is any string except a and b$\}$

1.6 Give state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $\{0,1\}$.

- a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$
- b. $\{w \mid w$ contains at least three 1s$\}$
- c. $\{w \mid w$ contains the substring 0101 (i.e., $w = x0101y$ for some x and y)$\}$
- d. $\{w \mid w$ has length at least 3 and its third symbol is a 0$\}$
- e. $\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$
- f. $\{w \mid w$ doesn't contain the substring 110$\}$
- g. $\{w \mid$ the length of w is at most 5$\}$
- h. $\{w \mid w$ is any string except 11 and 111$\}$
- i. $\{w \mid$ every odd position of w is a 1$\}$
- j. $\{w \mid w$ contains at least two 0s and at most one 1$\}$
- k. $\{\varepsilon, 0\}$
- l. $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$
- m. The empty set
- n. All strings except the empty string

1.7 Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is $\{0,1\}$.

- [A]a. The language $\{w \mid w$ ends with 00$\}$ with three states
- b. The language of Exercise 1.6c with five states
- c. The language of Exercise 1.6l with six states
- d. The language $\{0\}$ with two states
- e. Th...

1.9 Use the construction in the proof of Theorem 1.47 to give the state diagrams of NFAs recognizing the concatenation of the languages described in

a. Exercises 1.6g and 1.6i.
b. Exercises 1.6b and 1.6m.

1.10 Use the construction in the proof of Theorem 1.49 to give the state diagrams of NFAs recognizing the star of the languages described in

a. Exercise 1.6b.
b. Exercise 1.6j.
c. Exercise 1.6m.

[A]**1.11** Prove that every NFA can be converted to an equivalent one that has a single accept state.

1.12 Let $D = \{w|\ w$ contains an even number of a's and an odd number of b's and does not contain the substring ab$\}$. Give a DFA with five states that recognizes D and a regular expression that generates D. (Suggestion: Describe D more simply.)

1.13 Let F be the language of all strings over $\{0,1\}$ that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognizes F. (You may find it helpful first to find a 4-state NFA for the complement of F.)

1.14 **a.** Show that if M is a DFA that recognizes language B, swapping the accept and nonaccept states in M yields a new DFA recognizing the complement of B. Conclude that the class of regular languages is closed under complement.
b. Show by giving an example that if M is an NFA that recognizes language C, swapping the accept and nonaccept states in M doesn't necessarily yield a new NFA that recognizes the complement of C. Is the class of languages recognized by NFAs closed under complement? Explain your answer.

1.15 Give a counterexample to show that the following construction fails to prove Theorem 1.49, the closure of the class of regular languages under the star operation.[7] Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1. Construct $N = (Q_1, \Sigma, \delta, q_1, F)$ as follows. N is supposed to recognize A_1^*.

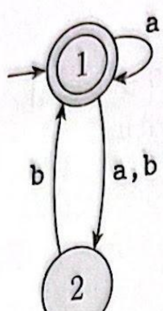
a. The states of N are the states of N_1.
b. The start state of N is the same as the start state of N_1.
c. $F = \{q_1\} \cup F_1$.
The accept states F are the old accept states plus its start state.
d. Define δ so that for any $q \in Q_1$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \notin F_1 \text{ or } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a = \varepsilon. \end{cases}$$

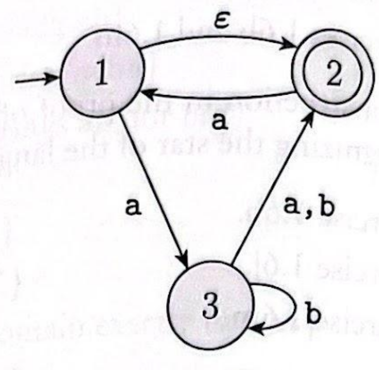
(Suggestion: Show this construction graphically, as in Figure 1.50.)

...tomaton, N_1, for which the constructed

1.16 Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.



(a)



(b)

1.17 **a.** Give an NFA recognizing the language $(01 \cup 001 \cup 010)^*$.

b. Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

1.18 Give regular expressions generating the languages of Exercise 1.6.

1.19 Use the procedure described in Lemma 1.55 to convert the following regular expressions to nondeterministic finite automata.

a. $(0 \cup 1)^*000(0 \cup 1)^*$

b. $(((00)^*(11)) \cup 01)^*$

c. $\emptyset^*$

1.20 For each of the following languages, give two strings that are members and two strings that are *not* members—a total of four strings for each part. Assume the alphabet $\Sigma = \{a,b\}$ in all parts.

a. a^*b^*

b. $a(ba)^*b$

c. $a^* \cup b^*$

d. $(aaa)^*$

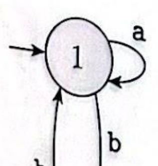
e. $\Sigma^* a \Sigma^* b \Sigma^* a \Sigma^*$

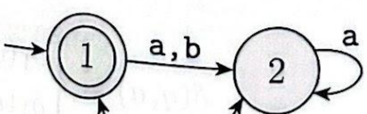
f. $aba \cup bab$

g. $(\varepsilon \cup a)b$

h. $(a \cup ba \cup bb)\Sigma^*$

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





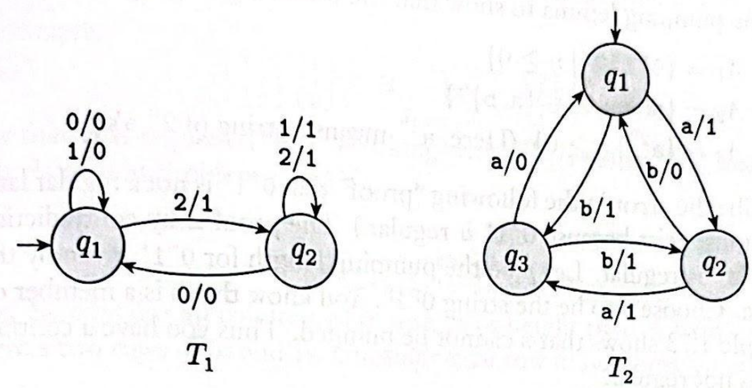
1.22 In certain programming languages, comments appear between delimiters such as `/#` and `#/`. Let C be the language of all valid delimited comment strings. A member of C must begin with `/#` and end with `#/` but have no intervening `#/`. For simplicity, assume that the alphabet for C is $\Sigma = \{\texttt{a}, \texttt{b}, /, \#\}$.

a. Give a DFA that recognizes C.

b. Give a regular expression that generates C.

[A]**1.23** Let B be any language over the alphabet Σ. Prove that $B = B^+$ iff $BB \subseteq B$.

1.24 A ***finite state transducer*** (FST) is a type of deterministic finite automaton whose output is a string and not just *accept* or *reject*. The following are state diagrams of finite state transducers T_1 and T_2.



Each transition of an FST is labeled with two symbols, one designating the input symbol for that transition and the other designating the output symbol. The two symbols are written with a slash, /, separating them. In T_1, the transition from q_1 to q_2 has input symbol 2 and output symbol 1. Some transitions may have multiple input–output pairs, such as the transition in T_1 from q_1 to itself. When an FST computes on an input string w, it takes the input symbols $w_1 \cdots w_n$ one by one and, starting at the start state, follows the transitions by matching the input labels with the sequence of symbols $w_1 \cdots w_n = w$. Every time it goes along a transition, it outputs the corresponding output symbol. For example, on input 2212011, machine T_1 enters the sequence of states $q_1, q_2, q_2, q_2, q_2, q_1, q_1, q_1$ and produces output 1111000. On input abbb, T_2 outputs 1011. Give the sequence of states entered and the output produced in each of the following parts.

a. T_1 on input 011

b. T_1 on input 211

c. T_1 on input 121

d. T_1 on input 0202

e. T_2 on input b

f. T_2 on input bbab

g. T_2 on input bbbbbb

h. T_2 on input ε

1.25 Read the informal definition of the finite state transducer given in Exercise 1.24. Give a formal definition of this model, following the pattern in Definition 1.5 (page 35). Assume that an FST has an input alphabet Σ and an output alphabet Γ but not a set of accept states. Include a formal definition of the computation of an FST. (Hint: An FST is a 5-tuple. Its transition function is of the form $\delta\colon Q\times\Sigma\longrightarrow Q\times\Gamma$.)

1.26 Using the solution you gave to Exercise 1.25, give a formal description of the machines T_1 and T_2 depicted in Exercise 1.24.

1.27 Read the informal definition of the finite state transducer given in Exercise 1.24. Give the state diagram of an FST with the following behavior. Its input and output alphabets are {0,1}. Its output string is identical to the input string on the even positions but inverted on the odd positions. For example, on input 0000111 it should output 1010010.

1.28 Convert the following regular expressions to NFAs using the procedure given in Theorem 1.54. In all parts, $\Sigma = \{\texttt{a}, \texttt{b}\}$.

a. $\texttt{a}(\texttt{abb})^* \cup \texttt{b}$
b. $\texttt{a}^+ \cup (\texttt{ab})^+$
c. $(\texttt{a} \cup \texttt{b}^+)\texttt{a}^+\texttt{b}^+$

1.29 Use the pumping lemma to show that the following languages are not regular.

[A] a. $A_1 = \{0^n 1^n 2^n \mid n \geq 0\}$
b. $A_2 = \{www \mid w \in \{\texttt{a}, \texttt{b}\}^*\}$
[A] c. $A_3 = \{\texttt{a}^{2^n} \mid n \geq 0\}$ (Here, $\texttt{a}^{2^n}$ means a string of 2^n a's.)

1.30 Describe the error in the following "proof" that 0^*1^* is not a regular language. (An error must exist because 0^*1^* *is* regular.) The proof is by contradiction. Assume that 0^*1^* is regular. Let p be the pumping length for 0^*1^* given by the pumping lemma. Choose s to be the string 0^p1^p. You know that s is a member of 0^*1^*, but Example 1.73 shows that s cannot be pumped. Thus you have a contradiction. So 0^*1^* is not regular.

PROBLEMS

1.31 For languages A and B, let the ***perfect shuffle*** of A and B be the language

$$\{w \mid w = a_1b_1 \cdots a_kb_k, \text{ where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ each } a_i, b_i \in \Sigma\}.$$

Show that the class of regular languages is closed under perfect shuffle.

1.32 For languages A and B, let the ***shuffle*** of A and B be the language

$$\{w \mid w = a_1b_1 \cdots a_kb_k, \text{ where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ each } a_i, b_i \in \Sigma^*\}.$$

Show that the class of regular languages is closed under shuffle.

1.33 Let A be any language. Define *DROP-OUT*(A) to be the language containing all strings that can be obtained by removing one symbol from a string in A. Thus, *DROP-OUT*$(A) = \{xz \mid xyz \in A \text{ where } x, z \in \Sigma^*, y \in \Sigma\}$. Show that the class of regular languages is closed under the *DROP-OUT* operation. Give both a proof by picture and a more formal proof by construction as in Theorem 1.47.

***1.35** Let $A/B = \{w \mid wx \in A \text{ for some } x \in B\}$. Show that if A is regular and B is any language, then A/B is regular.

1.36 For any string $w = w_1w_2 \cdots w_n$, the ***reverse*** of w, written $w^{\mathcal{R}}$, is the string w in reverse order, $w_n \cdots w_2w_1$. For any language A, let $A^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in A\}$. Show that if A is regular, so is $A^{\mathcal{R}}$.

1.37 Let

$$\Sigma_3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \ldots, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}.$$

Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

$$B = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the sum of the top two rows}\}.$$

For example,

$$\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} \in B, \quad \text{but} \quad \begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix} \notin B.$$

Show that B is regular. (Hint: Working with $B^{\mathcal{R}}$ is easier. You may assume the result claimed in Problem 1.36.)

1.38 Let

$$\Sigma_2 = \left\{ \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix} \right\}.$$

Here, Σ_2 contains all columns of 0s and 1s of height two. A string of symbols in Σ_2 gives two rows of 0s and 1s. Consider each row to be a binary number and let

$$C = \{w \in \Sigma_2^* \mid \text{the bottom row of } w \text{ is three times the top row}\}.$$

For example, $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \in C$, but $\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} \notin C$. Show that C is regular. (You may assume the result claimed in Problem 1.36.)

1.39 Let Σ_2 be the same as in Problem 1.38. Consider each row to be a binary number and let

$$D = \{w \in \Sigma_2^* \mid \text{the top row of } w \text{ is a larger number than is the bottom row}\}.$$

For example, $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \in D$, but $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \notin D$. Show that D is regular.

1.40 Let Σ_2 be the same as in Problem 1.38. Consider the top and bottom rows to be strings of 0s and 1s, and let

$$E = \{w \in \Sigma_2^* \mid \text{the bottom row of } w \text{ is the reverse of the top row of } w\}.$$

Show that E is not regular.

1.41 Let $B_n = \{\mathtt{a}^k \mid k \text{ is a multiple of } n\}$. Show that for each $n \geq 1$, the language B_n is regular.

1.42 Let $C_n = \{x \mid x \text{ is a binary number that is a multiple of } n\}$. Show that for each $n \geq 1$, the language C_n is regular.

1.43 An *all*-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if *every* possible state that M could be in after reading input x is a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* state among these possible states is an … Prove that all-NFAs recognize the class of regular languages.

1.44 The construction in Theorem 1.54 shows that every GNFA is equivalent to a GNFA with only two states. We can show that an opposite phenomenon occurs for DFAs. Prove that for every $k > 1$, a language $A_k \subseteq \{0,1\}^*$ exists that is recognized by a DFA with k states but not by one with only $k - 1$ states.

1.45 Recall that string x is a ***prefix*** of string y if a string z exists where $xz = y$, and that x is a ***proper prefix*** of y if in addition $x \neq y$. In each of the following parts, we define an operation on a language A. Show that the class of regular languages is closed under that operation.

- ^{A}**a.** $NOPREFIX(A) = \{w \in A|$ no proper prefix of w is a member of $A\}$.
- **b.** $NOEXTEND(A) = \{w \in A|\ w$ is not the proper prefix of any string in $A\}$.

^{A}**1.46** Read the informal definition of the finite state transducer given in Exercise 1.24. Prove that no FST can output $w^{\mathcal{R}}$ for every input w if the input and output alphabets are $\{0,1\}$.

1.47 Let x and y be strings and let L be any language. We say that x and y are ***distinguishable by L*** if some string z exists whereby exactly one of the strings xz and yz is a member of L; otherwise, for every string z, we have $xz \in L$ whenever $yz \in L$ and we say that x and y are ***indistinguishable by L***. If x and y are indistinguishable by L, we write $x \equiv_L y$. Show that $\equiv_L$ is an equivalence relation.

^{A*}**1.48** **Myhill–Nerode theorem.** Refer to Problem 1.47. Let L be a language and let X be a set of strings. Say that X is ***pairwise distinguishable by L*** if every two distinct strings in X are distinguishable by L. Define the ***index of L*** to be the maximum number of elements in any set that is pairwise distinguishable by L. The index of L may be finite or infinite.

- **a.** Show that if L is recognized by a DFA with k states, L has index at most k.
- **b.** Show that if the index of L is a finite number k, it is recognized by a DFA with k states.
- **c.** Conclude that L is regular iff it has finite index. Moreover, its index is the size of the smallest DFA recognizing it.

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

- **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.
- **c.** Explain why parts (a) and (b) do not contradict the pumping lemma.

1.50 The pumping lemma says that every regular language has a pumping length p, such that every string in the language can be pumped if it has length p or more. If p is a pumping length for language A, so is any length $p' \geq p$. The ***minimum pumping length*** for A is the smallest p that is a pumping length for A. For example, if $A = 01^*$, the minimum pumping length is 2. The reason is that the string $s = 0$ is

[A]a. 0001^*
[A]b. 0^*1^*
c. $001 \cup 0^*1^*$
[A]d. $0^*1^+0^+1^* \cup 10^*1$
e. $(01)^*$
f. ε
g. $1^*01^*01^*$
h. $10(11^*0)^*0$
i. 1011
j. Σ^*

1.51 Prove that the following languages are not regular. You may use the pumping lemma and the closure of the class of regular languages under union, intersection, and complement.

a. $\{0^n1^m0^n \mid m, n \geq 0\}$
[A]b. $\{0^m1^n \mid m \neq n\}$
c. $\{w \mid w \in \{0,1\}^*$ is not a palindrome$\}$[8]
*d. $\{wtw \mid w, t \in \{0,1\}^+\}$

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

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

Prove that Y is not regular.

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

$$D = \{w \mid w \text{ contains an equal number of occurrences of the substrings 01 and 10}\}.$$

Thus $101 \in D$ because 101 contains a single 01 and a single 10, but $1010 \notin D$ because 1010 contains two 10s and one 01. Show that D is a regular language.

1.54 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 in terms of both a state diagram and a formal description.

1.55 Consider the languages C_k defined in Problem 1.54. Prove that for each k, no DFA can recognize C_k with fewer than 2^k states.

1.56 Let $\Sigma = \{a, b\}$. For each $k \geq 1$, let D_k be the language consisting of all strings that have at least one a among the last k symbols. Thus $D_k = \Sigma^*a(\Sigma \cup \varepsilon)^{k-1}$. Describe a DFA with at most $k+1$ states that recognizes D_k in terms of both a state diagram and a formal description.

*1.57 a. Let A be an infinite regular language. Prove that A can be split into two infinite disjoint regular subsets.
b. Let B and D be two languages. Write $B \Subset D$ if $B \subseteq D$ and D contains infinitely many strings that are not in B. Show that if B and D are two regular languages where $B \Subset D$, then we can find a regular language C where $B \Subset C \Subset D$.

[8]A *palindrome* is a string that reads the same forward and backward.

1.58 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, 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 (c) is nearly tight; that is, for each k, demonstrate an NFA recognizing a language A_k where $\overline{A_k}$ is nonempty and where $\overline{A_k}$'s shortest member strings are of length exponential in k. Come as close to the bound in (c) as you can.

***1.59** Prove that for each $n > 0$, a language B_n exists where

a. B_n is recognizable by an NFA that has n states, and

b. if $B_n = A_1 \cup \cdots \cup A_k$, for regular languages A_i, then at least one of the A_i requires a DFA with exponentially many states.

1.60 A ***homomorphism*** is a function $f\colon \Sigma \longrightarrow \Gamma^*$ from one alphabet to strings over another alphabet. We can extend f to operate on strings by defining $f(w) = f(w_1)f(w_2)\cdots f(w_n)$, where $w = w_1w_2\cdots w_n$ and each $w_i \in \Sigma$. We further extend f to operate on languages by defining $f(A) = \{f(w)|\ w \in A\}$, for any language A.

a. Show, by giving a formal construction, that the class of regular languages is closed under homomorphism. In other words, given a DFA M that recognizes B and a homomorphism f, construct a finite automaton M' that recognizes $f(B)$. Consider the machine M' that you constructed. Is it a DFA in every case?

b. Show, by giving an example, that the class of non-regular languages is not closed under homomorphism.

1.61** Let the ***rotational closure of language A be $RC(A) = \{yx|\ xy \in A\}$.

a. Show that for any language A, we have $RC(A) = RC(RC(A))$.

b. Show that the class of regular languages is closed under rotational closure.

1.62 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.

***1.63** If A is a set of natural numbers and k is a natural number greater than 1, let

$$B_k(A) = \{w|\ w \text{ is the representation in base } k \text{ of some number in } A\}.$$

*1.65 If A is any language, let $A_{\frac{1}{3}-\frac{1}{3}}$ be the set of all strings in A with their middle thirds removed so that

$$A_{\frac{1}{3}-\frac{1}{3}} = \{xz \mid \text{for some } y,\ |x| = |y| = |z| \text{ and } xyz \in A\}.$$

Show that if A is regular, then $A_{\frac{1}{3}-\frac{1}{3}}$ is not necessarily regular.

*1.66 Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and let h be a state of M called its "home". A ***synchronizing sequence*** for M and h is a string $s \in \Sigma^*$ where $\delta(q, s) = h$ for every $q \in Q$. (Here we have extended δ to strings, so that $\delta(q, s)$ equals the state where M ends up when M starts at state q and reads input s.) Say that M is ***synchronizable*** if it has a synchronizing sequence for some state h. Prove that if M is a k-state synchronizable DFA, then it has a synchronizing sequence of length at most k^3. Can you improve upon this bound?

1.67 We define the ***avoids*** operation for languages A and B to be

$$A \textit{ avoids } B = \{w \mid w \in A \text{ and } w \text{ doesn't contain any string in } B \text{ as a substring}\}.$$

Prove that the class of regular languages is closed under the *avoids* operation.

1.68 Let $\Sigma = \{0,1\}$.

a. Let $A = \{0^k u 0^k \mid k \geq 1 \text{ and } u \in \Sigma^*\}$. Show that A is regular.
b. Let $B = \{0^k 1 u 0^k \mid k \geq 1 \text{ and } u \in \Sigma^*\}$. Show that B is not regular.

1.69 Let M_1 and M_2 be DFAs that have k_1 and k_2 states, respectively, and then let $U = L(M_1) \cup L(M_2)$.

a. Show that if $U \neq \emptyset$, then U contains some string s, where $|s| < \max(k_1, k_2)$.
b. Show that if $U \neq \Sigma^*$, then U excludes some string s, where $|s| < k_1 k_2$.

1.70 Let $\Sigma = \{0,1,\#\}$. Let $C = \{x\#x^{\mathcal{R}}\#x \mid x \in \{0,1\}^*\}$. Show that $\overline{C}$ is a CFL.

1.71 **a.** Let $B = \{1^k y \mid y \in \{0,1\}^* \text{ and } y \text{ contains at least } k \text{ 1s, for } k \geq 1\}$. Show that B is a regular language.
b. Let $C = \{1^k y \mid y \in \{0,1\}^* \text{ and } y \text{ contains at most } k \text{ 1s, for } k \geq 1\}$. Show that C isn't a regular language.

*1.72 In the traditional method for cutting a deck of playing cards, the deck is arbitrarily split two parts, which are exchanged before reassembling the deck. In a more complex cut, called Scarne's cut, the deck is broken into three parts and the middle part in placed first in the reassembly. We'll take Scarne's cut as the inspiration for an operation on languages. For a language A, let $CUT(A) = \{yxz \mid xyz \in A\}$.

a. Exhibit a language B for which $CUT(B) \neq CUT(CUT(B))$.
b. Show that the class of regular languages is closed under CUT.

1.73 Let $\Sigma = \{0,1\}$. Let $WW_k = \{ww \mid w \in \Sigma^* \text{ and } w \text{ is of length } k\}$.

a. Show that for each k, no DFA can recognize WW_k with fewer than 2^k states.
b. Describe a much smaller NFA for $\overline{WW_k}$, the complement of WW_k.