Chapter 5

Regular and Nonregular Languages

5.1. If there is only one equivalence class, then there cannot be two strings for which one is in L and the other isn't. Therefore, either $L = \emptyset$ or $L = \{0,1\}^*$.

5.2. There are $|x| + 2$ equivalence classes, one for each prefix of x and one containing all the nonprefixes.

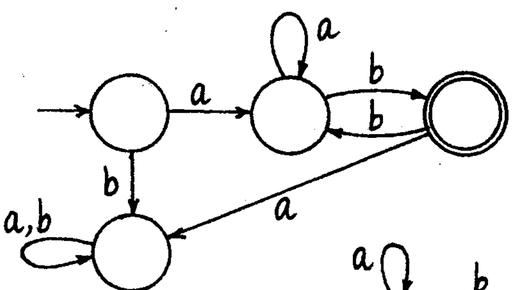
5.3. According to the proof of Theorem 3.3, the language $pal \subseteq \{0,1\}^*$ is an example.

5.4. It is enough to show that any two elements of S are equivalent, and no element of S is equivalent to any element of S'. If $x, y \in S$, then neither is a prefix of any element of L, and therefore xI_Ly. If $x \in S$, $y \notin S$, and z is a string for which $yz \in L$, then z distinguishes x from y.

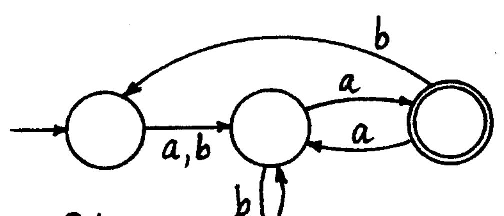
5.5. Suppose $xI_L\Lambda$ and $x \neq \Lambda$. Then for any $i \geq 1$, $x^{i+1}I_Lx^i$. (For any y, consider $x^iy = \Lambda(x^iy)$ and $x^{i+1}y = x(x^iy)$; since ΛI_Lx, either both these strings are in L or neither is.) Since $[\Lambda]$ contains all the strings x^i, it must be infinite. (Note: the same argument shows that if there are strings x and xy for which xI_Lxy and $y \neq \Lambda$, then $[x]$ is infinite.)

5.6. If there were some string x that were not a prefix of any element of L, then for every y, xy would also not be a prefix of any element of L, and so the set of strings that are not prefixes of elements of L would be infinite. By Exercise 5.4, however, this set is one of the equivalence classes. Therefore, if all equivalence classes are finite, there cannot be any such x.

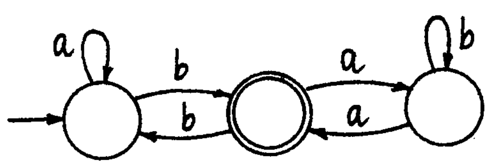
5.7.



5.8.

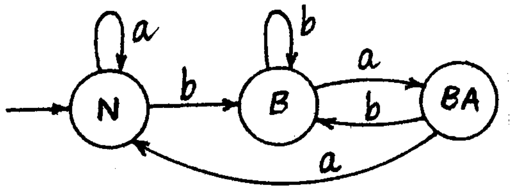


5.9.



5.10. Denote the three sets by B, BA, and N, respectively, and let $Q = \{B, BA, N\}$. If we consider an FA with state set Q, the picture looks like the one below except that the picture below doesn't indicate accepting states. There are eight different FAs, accepting eight different languages, obtained by making all possible choices of $A \subseteq Q$. However, the question asks only for languages L for which I_L has three equivalence classes, and some of the eight languages can be accepted by FAs with fewer than three states. Obvious examples are the languages $\emptyset$ and $\{a,b\}^*$, obtained by making none or all states accepting. The two

other examples are $\{a, b\}^*\{b\}$ and its complement, obtained by making B either the only accepting state or the only nonaccepting state, because both B and B' can be accepted by 2-state FAs.



There are four remaining languages, all of which have the desired property: BA, N, $B \cup BA$, and $B \cup N$.

5.11. Denote the three sets by AB, A, and B, respectively. By an argument similar to that in Exercise 5.10, there are four such languages: B, A, $B \cup AB$, and $A \cup AB$.

5.12. The partition is the same. The sets $\{\Lambda\}$ and $\{0^i1^i \mid i > 0\}$ are both subsets in the partition, just as before. The only difference is that before these were $\{\Lambda\}$ and L, now they are $\{\Lambda\}$ and $L - \{\Lambda\}$.

5.13. (a) and (b) Consider a string $x \in \{0, 1\}^*$, and let $k = n_0(x) - n_1(x)$. Saying that $k = 0$ is the same as saying that $x \in L$. If $k > 0$, then x has an excess of k 0's, so that for any z, $xz \in L$ if and only if z has an excess of k 1's. Similarly, if $k < 0$, $xz \in L$ if and only if z has an excess of $-k$ 0's. This means that if $n_0(x) - n_1(x) = n_0(y) - n_1(y)$, then x and y have the property that xz and yz will be in L for precisely the same strings z—i.e., x and y are equivalent. Conversely, if $n_0(x) - n_1(x) \neq n_0(y) - n_1(y)$, then any string z for which $n_1(z) - n_0(z) = n_0(x) - n_1(x)$ distinguishes x from y.

(c) Parts (a) and (b) say that an equivalence class containing a string x is determined by the difference $n_0(x) - n_1(x)$: another string y will be in this equivalence class precisely if $n_0(y) - n_1(y)$ is the same value as for x. This means that for every integer k, there is an equivalence class containing all those strings x for which the difference is k. Another way to say this is to say that the equivalence classes are

$$\ldots, [111], [11], [1], [\Lambda] = L, [0], [00], [000], \ldots$$

For each $k > 0$, $[1^k] = \{x \mid n_1(x) - n_0(x) = k\}$, and $[0^k] = \{x \mid n_0(x) - n_1(x) = k\}$.

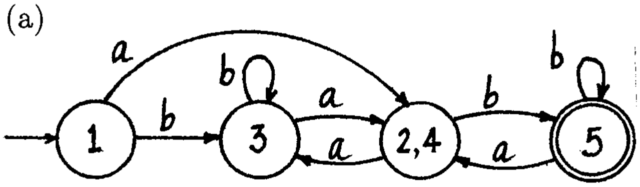
5.14. An easy induction proof shows that for every $q \in Q_1$ and every $x \in \Sigma^*$, $\delta^*(q, x) \in Q_1$. Now let $p, q \in Q_1$ and $x \in \Sigma^*$, and consider the states $\delta^*(p, x)$ and $\delta^*(q, x)$. In (a) we can say that neither is in A, and in (b) we can say that both are in A.

5.15. (a) For x and y to be distinguishable with respect to $\{0, 1\}^*\{010\}$, one of the two strings must end with a longer prefix of 010 than the other. In this case, if the longer prefix is α, and $\alpha\beta = 010$, the string β distinguishes the two strings. Therefore, the longest string that might be necessary is one of length 2. (Another way to say it is that if a string of length 3 was required to distinguish x and y, they wouldn't be distinguishable at all, since adding a string of length 3 or more would have the same effect in both cases.)

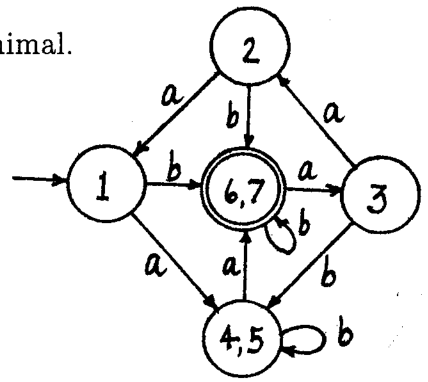
(b) If x has i more left parentheses than right, and y has j more left than right, then saying x and y are distinguishable with respect to L means that $i \neq j$. If k is the smaller of the two numbers i and j, a string of k right parentheses is the shortest string distinguishing x and y. Since i can be no larger than m, and j no larger than n, the smaller of i and j can be no larger than the smaller of m and n.

5.16. The FAs in parts (b), (c), and (f) are already minimal.

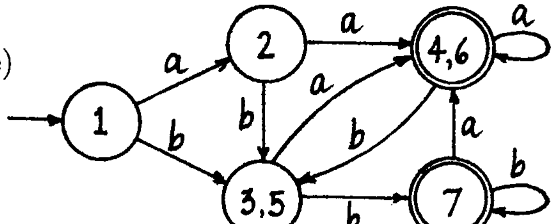
(a)



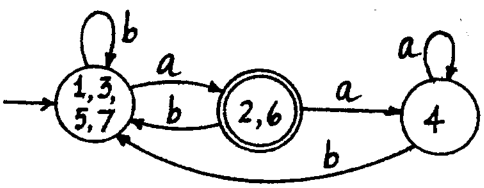
(d)



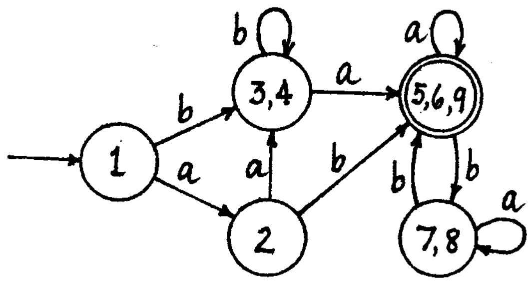
(e)



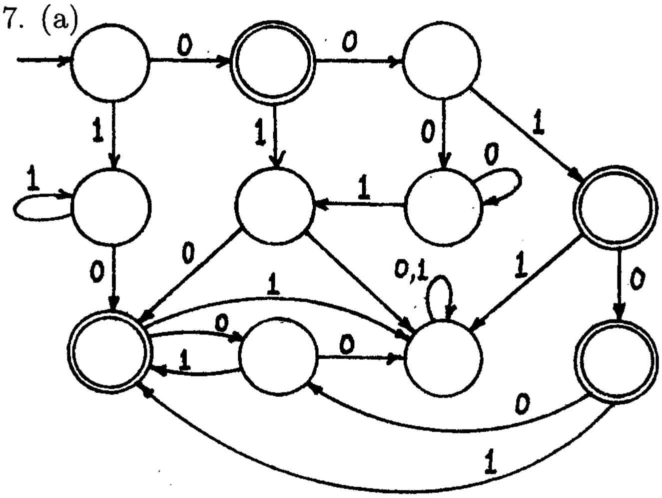
(g)



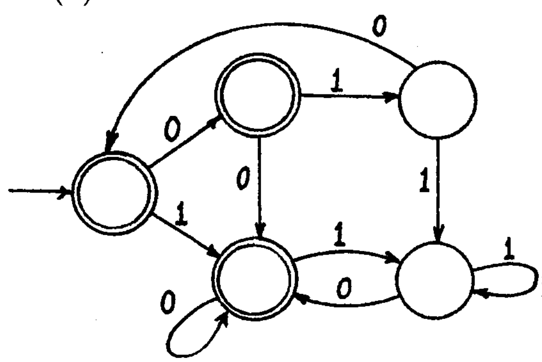
(h)



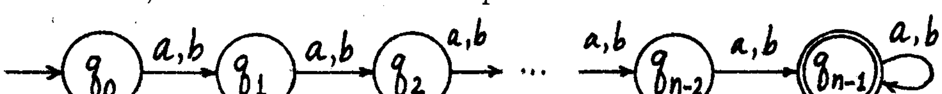
5.17. (a)



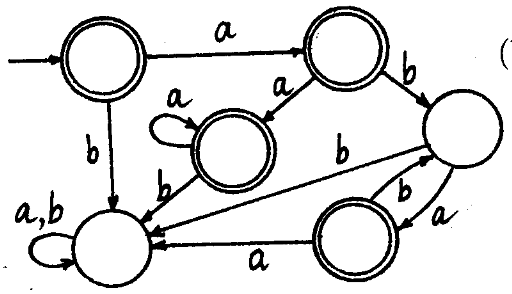
(b)

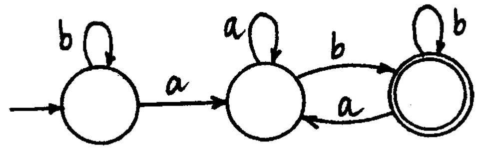


5.18. (a) Suppose the FA has n states. For the FA shown below, the only pair marked on the first pass is (q_{n-2}, q_{n-1}). If starting with pass 2 the pairs are considered in the order (q_0, q_1), (q_1, q_2), $\ldots$, (q_{n-3}, q_{n-2}), then one pair is marked on each pass. The total number of passes is $n - 1$, and no order would require more.



(b) For a pair (p, q) that is ultimately marked, let $n_{(p,q)}$ be the length of the shortest string z for which $\delta^*(p, z) \in A$ and $\delta^*(q, z) \notin A$ (or vice versa). If we process the pairs in nondecreasing order of $n_{(p,q)}$ (i.e., first all the pairs (p, q) with $n_{(p,q)} = 1$, then all the pairs (p, q) with $n_{(p,q)} = 2$, and so forth), then every pair that will ever be marked will be marked on the second pass.

5.19. (a) 

(b) 

5.20. Assume in each part that $0 \le i < j$.

(a) The string 10^{2i} distinguishes 0^i and 0^j.

(b) The string 10^{i+2} distinguishes 0^i and 0^j.

(c) The string 1^{2j} distinguishes 0^i and 0^j.

(d) The string 01^{i+1} distinguishes 0^i and 0^j.

(e) If i is odd, say $i = 2p + 1$, the string 1^{p+1} distinguishes 0^i and 0^j; if i is even, say $i = 2p$, the string 01^{p+1} distinguishes 0^i and 0^j.

(f) The string 1^j distinguishes 0^i and 0^j.

5.21. Suppose L is regular. Then by Theorem 5.3, there are strings u, v, and w so that $|v| > 0$ and $uv^mw \in L$ for every $m \ge 0$. However, this is impossible, because if $v = 0^k$ or $v = 1^k$, then for sufficiently large m the string uv^mw will have more of one symbol than the other; and if v contains both 0's and 1's, then uv^2w contains 1's before 0's. In both cases, therefore, $uv^mw \notin L$.

5.22. Consider the two strings ab^j and ab^k, where j and k are any numbers satisfying $0 \le j < k$. Then the string c^j distinguishes the two strings. This means that all the equivalence classes $[ab^j]$, where $j \ge 0$, are distinct, so that by Corollary 5.1 L cannot be regular.

5.23. (a) $L = \{0^n 1 0^{2n} \mid n \ge 0\}$. Suppose L is regular, and let n be the integer in the statement of the pumping lemma. Let $x = 0^n 1 0^{2n}$. Then $x \in L$, and $|x| \ge n$. Therefore, by the statement of the pumping lemma, $x = uvw$ for some strings u, v, and w satisfying $|uv| \le n$, $|v| > 0$, and $uv^mw \in L$ for every $m \ge 0$. The first of these three conditions implies that v is a string of 0's from the first group of 0's in x, and the second implies that $v \ne \Lambda$. Therefore, for some $j > 0$, $uv^2w = 0^{n+j} 1 0^{2n}$. However, the third condition says that uv^2w must be in L. This contradiction proves that L cannot be regular.

(b) $L = \{0^i 1^j 0^k \mid k > i + j\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n 1^n 0^{2n+1}$. Then $|x| \ge n$ and $x \in L$. Therefore, $x = uvw$ for some u, v, w satisfying the same three conditions as in the first part. As before, v must be 0^j for

some $j > 0$, and the 0's making up v come from the first group of 0's in x. Thus $uv^2w = 0^{n+j}1^n0^{2n+1}$, and this cannot be in L because $2n+j \geq 2n+1$. Again we have a contradiction, proving that L is not regular.

(c) $L = \{0^i1^j \mid j = i \text{ or } j = 2i\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n1^n$. Then $x = uvw$ for some u, v, and w satisfying the usual three conditions. As before, v must be 0^j for some $j > 0$. Therefore, $uv^2w = 0^{n+j}1^n$, and this string cannot be in L because n is neither $n + j$ nor $2(n + j)$. This contradiction implies that L is not regular.

(d) $L = \{0^i1^j \mid j \text{ is a multiple of } i\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n1^n$. Then $x = uvw$ for some u, v, and w satisfying the usual three conditions. As before, v must be 0^j for some $j > 0$, and $uv^2w = 0^{n+j}1^n$. Since n cannot be a multiple of $n + j$, this is a contradiction.

(e) $L = \{x \in \{0,1\}^* \mid n_0(x) < 2n_1(x)\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^{2n-1}1^n$. Then $x = uvw$ for some u, v, and w satisfying the usual three conditions. As before, v must be 0^j for some $j > 0$, and $uv^2w = 0^{2n+j-1}1^n$. Since $j - 1 \geq 0$, $2n + j - 1 \geq 2n$, and this contradiction implies that L is not regular.

(f) $L = \{x \in \{0,1\}^* \mid \text{no prefix of } x \text{ has more 1's than 0's}\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n1^n$. Then $x = uvw$, where u, v, and w satisfy the usual three conditions. As above, $v = 0^j$ for some $j > 0$. Therefore, $uv^0w = uw = 0^{n-j}1^n$, and this string cannot be in L because the string itself (which is a prefix of itself) has more 1's than 0's.

5.24. (a) Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n1^n0^n1^n$. Then $x = uvw$, where the usual three conditions on u, v, and w hold. Then $v = 0^j$ for some $j > 0$, and $z = uv^2w = 0^{n+j}1^n0^n1^n$. According to the pumping lemma, $z \in L$. However, $4n < |z| \leq 5n$, so that $2n < |z|/2 \leq 5n/2$. This means that even if $|z|$ is even, its midpoint is somewhere in the first string of 1's. However, this implies that the second half of z begins with 1. Since z itself begins with 0, z cannot be ww for any w, and this contradiction implies that L is not regular.

(b) $L = \{xy \mid x, y \in \{0,1\}^* \text{ and } y \text{ is either } x \text{ or } x^r\}$. Suppose L is regular, let n be the integer in the pumping lemma, and let $x = 0^n1^n0^n1^n$. Then $x = uvw$, where u, v, and w satisfy the usual three conditions. v must be 0^j for some j, and the 0's in v come from the first part of x. Therefore, $uv^2w = 0^{n+j}1^n0^n1^n$. The length of this string is $4n + j$. If j is odd we have a contradiction immediately, since all strings in L have even length; otherwise, the midpoint is between two of the 1's in the first group of 1's. This means that $0^{n+j}1^n0^n1^n$ cannot be of the form zz, since the first half starts with 0 and the second with 1. This string cannot be of the form zz^r either, because its initial and final symbols are different. This contradiction proves that L is not regular.

(c) Suppose L is regular, let n be the integer in the pumping lemma, and let $x = (^na)^n$ (that is, a preceded by n left parentheses and followed by n right parentheses). Then $x = uvw$, where the usual three conditions hold. It follows that $v = (^j$ for some $j > 0$, so that $uv^2w = (^{n+j}a)^n$. This string is not in L, and this contradiction implies that L is not regular.

5.25. No. Let L be the language of nonpalindromes over $\{0, 1\}$. L is not regular because its complement is not. However, if x begins with 0, $x1 \in L$; if x begins with 1, $x0 \in L$; and if $x = \Lambda$, $x01 \in L$. Therefore, L satisfies the condition for $k = 2$.

5.26. (a) False. Σ^* has a nonregular subset.

(b) False. Nonregular languages have finite subsets, and finite languages are regular.

(c) False. The union of any language and its complement is Σ^*, which is regular.

(d) False. The intersection of a nonregular language and its complement is empty, and the empty language is regular.

(e) True. The complement of a regular language is regular.

(f) False. L_2 could be a subset of L_1, for example.

(g) True. Since $L_1 \cap L_2$ is regular, so is $L_1 - (L_1 \cap L_2) = L_1 - L_2$. Now $L_2 = (L_1 \cup L_2) - (L_1 - L_2)$. Therefore, if $L_1 \cup L_2$ were regular, L_2 would also be regular.

(h) False. L_1 could be Σ^*, for example.

(i) False. Any language is the union of one-element languages; for example, $\{0^n 1^n \mid n \geq 0\} = \{\Lambda\} \cup \{01\} \cup \{0011\} \cup \ldots$.

(j) False. Here is a counterexample. For each $k \geq 1$, let $S_k = \{0^{2^k i} 1^{2^k i} \mid i \geq 0\}$. Thus $S_1 = \{\Lambda, 0^2 1^2, 0^4 1^4, 0^6 1^6, \ldots\}$, $S_2 = \{\Lambda, 0^4 1^4, 0^8 1^8, \ldots\}$, and so forth. Then $S_{k+1} \subseteq S_k$ for each k. The set $\cap_{k=0}^{\infty} S_k$ is $\{\Lambda\}$, which is regular, but it is easy to show that for every k, S_k is nonregular. Now let $L_k = S_k'$. It follows that L_k is nonregular and $L_k \subseteq L_{k+1}$, but $\cup_{k=1}^{\infty} L_k = (\cap_{k=1}^{\infty} S_k)' = \Sigma^* - \{\Lambda\}$, and this set is regular.

5.27. (a) Nonregular. For $i \geq 0$, let $x_i = 01^i$. Then for any i and j with $i < j$, the string 01^i distinguishes x_i and x_j, since $01^j 01^i$ has no nonnull prefix of the form ww.

(b) Regular. For any x, $x \in L$ if and only if x contains one of the substrings 00, 11, 0101, or 1010. (Reason: if x contains neither 00 nor 11, then x must consist of alternating 1's and 0's; therefore, if x contains a nonnull substring ww, it must contain either 0101 or 1010.)

(c) Nonregular. This can easily be proved using the pumping lemma, starting with a string of the form $1^n 0 1^n$.

(d) Nonregular. Let $S = \{0^{2i} \mid i \geq 0\}$. Then if $0 \leq i < j$, the strings 0^{2i} and 0^{2j} are distinguished by 1^{2i}, because the middle two symbols of $0^{2i}1^{2i}$ are unequal and the middle two symbols of $0^{2j}1^{2i}$ are both 0. S is therefore an infinite set, any two elements of which are distinguishable with respect to L.

(e) Nonregular. Suppose L is regular, and let n be the integer in the pumping lemma. Let $x = 0^n 1^n 0^n 1^n = (0^n 1^n)\Lambda(0^n 1^n)$. Then $x = uvw$, where u, v, and w satisfy the usual three conditions. As usual, v must consist of one or more 0's from the first group of 0's, say $v = 0^j$. Then $uv^2w = 0^{n+j}1^n 0^n 1^n$, and it is easy to see that this string is not of the form xyx for any x with $|x| \geq 1$.

(f) Nonregular, since the set of palindromes is nonregular.

(g) Nonregular. Let $S = \{001^i \mid i \geq 0\}$. For any i and j with $0 \leq i < j$, the two elements 001^i and 001^j of S are distinguished by the string $01^i 00$.

(h) Nonregular. First we observe that if L were regular, then $\{a^{n^2} \mid n \geq 0\}$ would be

also. Now we can use Theorem 5.4. If the set of perfect squares contained an arithmetic progression, then there would be integers p and q, with $q > 0$, so that $p + iq$ was a perfect square for every $i \geq 0$. In particular, two perfect squares could be found that were as large as we liked but differed by exactly q. However, $(n+1)^2 - n^2 = 2n + 1$, which means that if two distinct perfect squares are both $\geq q^2$, their difference is at least $2q + 1$.

5.28. (a) We have an algorithm to construct an FA accepting the complement of the language accepted by a given FA. Apply this to find M_1' and M_2' accepting $L(M_1)'$ and $L(M_2)'$, respectively. Now construct the FA accepting $L(M_1') \cap L(M_2)'$; finally, apply the algorithm on page 187 to determine whether this FA accepts any strings.

(c) On page 187 there is an algorithm to determine the states reachable from the initial state q_0 of an FA. This algorithm can be easily modified so that it determines the states reachable from q using nonnull strings. (The set T_0 is initialized to be $\emptyset$ instead of q.) The problem is then reduced to determining whether q is an element of this set.

(g) Apply the minimization algorithm described in Section 5.2 to obtain a minimum-state FA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta)$ accepting L. x and y are distinguishable with respect to L if and only if $\delta_1^*(q_1, x) \neq \delta_1^*(q_1, y)$.

(h) If $\delta^*(q_0, x) = q$, then x is a prefix of an element of L if and only if the set of states reachable from q includes at least one accepting state.

(j) The string x is a substring of an element of L if and only if there is a pair (p, q) of states in M satisfying these three conditions: i) p is reachable from q_0, the initial state; ii) $\delta^*(p, x) = q$; iii) the set of states reachable from q includes an accepting state. By testing each pair (p, q) if necessary, we can determine whether there is such a pair.

5.29. One example is the language L of all strings having equal numbers of 0's and 1's. L is nonregular and $L^* = L$.

5.30. One example is *pal*, the set of palindromes. For this L, $L^* = \{0, 1\}^*$, since Λ and all strings of length 1 are palindromes.

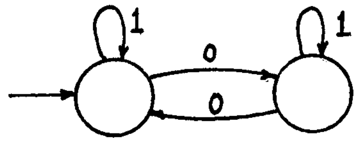
5.31. Suppose x and y are distinguishable with respect to L_1, and specifically that for some z, $xz \in L_1$ and $yz \notin L_1$. Then for some w, $xzw \in L$, since xz is a prefix of an element of L; and $yzw \notin L$, since yz is not a prefix of an element of L. Therefore, if x and y are in different equivalence classes with respect to I_{L_1}, they are in different equivalence classes with respect to I_L. This means that the partition determined by I_L is at least as fine as that determined by I_{L_1}: Any equivalence class with respect to I_{L_1} is the union of equivalence classes with respect to I_L.

5.32. One obvious way to approach this problem is to think about two-state FAs. If there are exactly two equivalence classes, then both states must be reachable from the initial state, and there must be exactly one accepting state. To answer (a), we don't need to worry about which is the accepting state; we only need to draw the transitions in every possible way. There are three ways of drawing transitions from the initial state, since the other state must be reachable, and there are four ways of drawing transitions from the

other state. Each of these twelve combinations gives a different set of strings corresponding to the initial state, and each of these sets is a possible answer to (a). They are described as follows, using regular expressions:

Λ	1^*	0^*
$((0+1)0^*1)^*$	$(1+00^*1)^*$	$(0+10^*1)^*$
$((0+1)1^*0)^*$	$(1+01^*0)^*$	$(0+11^*0)^*$
$(00+01+10+11)^*$	$(1+00+01)^*$	$(0+10+11)^*$

Each column represents a possible way of drawing transitions from the initial state, and each row represents a possible way of drawing transitions from the other state. For example, the entry in the second column, third row is obtained by drawing the FA shown here:



(b) For each of these twelve sets, there are two languages L for which the equivalence class $[\Lambda]$ is this set: one obtained by designating the initial state as the accepting state, the other obtained by designating the other state as the accepting state. In other words, the set could be either the language or the complement of the language.

5.33. It follows from Exercise 3.45 that the equivalence classes are all of the form $\{x\}$, where $x \in \{0,1\}^*$.

5.34. According to Exercise 5.4, one equivalence class is the set of all strings that are not prefixes of any element of L. These are the strings having a prefix with more right parentheses than left. Of the strings that are prefixes of balanced strings of parentheses, the equivalence class is determined by the number of excess left parentheses. (If x_1 and x_2 are both prefixes of balanced strings, and x_1 has k more left parentheses than right, and x_2 has j more left parentheses than right, then x_1 and x_2 are equivalent if $k = j$, and otherwise the string $)^k$ distinguishes x_1 and x_2 relative to L.)

This means that the equivalence classes are N, $L = [\Lambda] = [(^0]$, $[(]$, $[(^2]$, $[(^3]$, ..., where N is the set of nonprefixes of elements of L and $[(^k]$ is the set of all strings of parentheses that are prefixes of elements of L and have k more left parentheses than right.

5.35. (a) $\{\Lambda\}$, L, and the set of strings that are not prefixes of any element of L are three of the equivalence classes. We can describe the general form of all the other equivalence classes by considering a specific example. Consider a string that is a prefix of an element of L, in which there are 3 unmatched left parentheses (parentheses without the matching right parentheses). Then there are sixteen "archetypal" pairwise inequivalent strings of this type: $((($, $(((i$, $(((i+$, $(((i+i$, $((i+($, $((i+(i$, $((i+(i+$, $((i+(i+i$, $(i+(($, $(i+((i$, $(i+((i+$, $(i+((i+i$, $(i+(i+($, $(i+(i+(i$, $(i+(i+(i+$, and $(i+(i+(i+i$. It is not hard to see that any two of these are distinguishable with respect to L. (For example, consider the two strings $x = ((i+(i$ and $y = (i+((i+i$. Let $z = +i))+i)$. Then $xz \in L$ and $yz \notin L$.) Furthermore, every other string that is a prefix of an element of L and that has exactly

three unmatched left parentheses is equivalent to one of these, and looks like one of these except that one or more of the i's may be replaced by other elements of L. For example, a string in the same equivalence class as $((i + (i+$ is $(((i + i) + (((i + i) + i)+$. (The first i has been replaced by $(i + i)$ and the second by $((i + i) + i)$.)

We can enumerate these sixteen as follows. For each of the first two unmatched left parentheses, there are two possibilities: either it is followed immediately by another unmatched left parenthesis, or it is followed immediately by an element of L and a $+$. For the third unmatched left parenthesis, there are four possibilities: there is nothing after it; it is followed by an element of L and nothing else; it is followed by an element of L and a $+$; or it is followed by an element of L, a $+$, and another element of L. Altogether, then, there are $2 * 2 * 4 = 16$ combinations.

In general, for each $n > 0$, there are $2^{n-1} * 4$ equivalence classes, which can be enumerated by considering the 2 possibilities for each of the first $n - 1$ unmatched parentheses and the 4 possibilities for the last.

(b) If the expressions are not required to be "fully parenthesized", it is much easier to describe the equivalence classes. The first three are the same as in (a). In addition, for each $n > 0$, there are two equivalence classes corresponding to n: the set of all strings with n unmatched left parentheses in which the last unmatched parenthesis is followed either by nothing or by an element of L and a $+$; and the set of all strings with n unmatched left parentheses in which the last unmatched parenthesis is followed by an element of L. We can see by looking at the sixteen strings in (a) why there are fewer equivalence classes in this case: the strings $(i + ((i + i$ and $(((i$, for example, are equivalent, because of the additional options regarding parentheses.

5.36. (a) Every equivalence class has exactly one element, which means that any two distinct strings are distinguishable with respect to L. The proof is very similar to the proof of Exercise 3.45.

(b) Every equivalence class has exactly two elements, except the one containing Λ, which has only one. For any $x \neq \Lambda$, x and $x^{\sim}$ are equivalent, but no other string is equivalent to either of these.

5.37. We show that any two distinct strings are distinguishable with respect to L, so that every equivalence class has exactly one element. Suppose $x, y \in \{0,1\}^*$ and $x \neq y$. We denote $n_0(x)$ by i, $n_0(y)$ by j, $n_1(x)$ by $i + p$, and $n_1(y)$ by $j + q$.

We consider two cases. First, suppose $p = q$—i.e., x has the same excess of one symbol as y. Then since $x \neq y$, the two numbers i and j are different. In this case we choose z containing N 0's and $(2N + i - p)$ 1's, where N is yet to be described. Then $n_0(xz) = i + N$, and $n_1(xz) = i + p + 2N + i - p = 2(N + i)$, so that $n_1(xz)/n_0(xz) = 2$. On the other hand,

$$\frac{n_1(yz)}{n_0(yz)} = \frac{j + p + 2N + i - p}{j + N} = \frac{2 + \frac{i+j}{N}}{1 + \frac{j}{N}}$$

Now it is clear that since $i + j \neq 2j$, N can be chosen large enough so that this fraction is not an integer. This means that $xz \in L$ and $yz \notin L$.

In the second case, when $p \neq q$, we choose z containing N 0's and $(N-p)$ 1's. Then xz has $(i+N)$ 0's and $i+p+N-p=(i+N)$ 1's, while yz has $(j+N)$ 0's and $(j+q+N-p)$ 1's. It is easy to see that since $q-p \neq 0$, then by choosing N sufficiently large we can guarantee that $(j+q+N-p)/(j+N)$ is not an integer, so that $xz \in L$ and $yz \notin L$.

5.38. Suppose R is a right invariant equivalence relation on Σ^* so that the set of equivalence classes of R is finite and L is the union of some of the equivalence classes. If xRy, then for any z, $xzRyz$, since R is right invariant. This means that xz and yz are in the same equivalence class with respect to R. But since L is the union of some of the equivalence classes, any equivalence class that intersects L must be completely within L; therefore, for any z, $xz \in L$ if and only if $yz \in L$. Therefore, xI_Ly. This means that the partition of Σ^* determined by R is finer than that determined by I_L, which means that every equivalence class with respect to I_L is the union of equivalence classes with respect to R. Since the number of equivalence classes with respect to R is finite, the number of equivalence classes with respect to I_L must be finite.

5.39. (a) Suppose on the one hand that there is a right invariant partition of $\{0,1\}^*$, so that S is one of the subsets in the partition. Right invariant means that whenever x and y are in the same subset, then for any z, xz and yz are in the same subset. Therefore, if x and y are both in S, then for any z, xz and yz are in the same subset of the partition, so that if one of the two is in S, the other must be also.

On the other hand, suppose S has the property that for any $x, y \in S$ and any z, xz and yz are either both in S or both not in S. We consider the following sets: first, all the one-element sets $\{x\}$, where no prefix of x is in S; second, the set S; finally, all sets of the form $S\{\alpha\}$, where $S\{\alpha\} \not\subseteq S$ and $S\{\gamma\} \subseteq S$ for every proper prefix γ of α. (We summarize this last condition by saying that α is a *minimal* string so that $S\{\alpha\} \not\subseteq S$.)

A useful observation is that if $x \in S\{\alpha\}$ and $x \in S\{\beta\}$, where $S\{\alpha\} \not\subseteq S$, $S\{\beta\} \not\subseteq S$, and α and β are both minimal strings with this property, then $\alpha = \beta$. To see this, suppose $\alpha \neq \beta$ and $x = s_1\alpha = s_2\beta$, where $s_1, s_2 \in S$. The strings α and β can't be the same length; suppose $|\alpha| > |\beta|$. Then $\alpha = \gamma\beta$, for some γ, so that $s_1\gamma = s_2$. Because of the assumption on S, it follows that $S\{\gamma\} \subseteq S$, so that α can't be minimal.

Now we check that these subsets of Σ^* form a partition. Obviously, if x is a string such that no prefix of x is in S, then x cannot be in S or any of the sets $S\{\alpha\}$. If $S\{\alpha\} \not\subseteq S$, then $S\{\alpha\}$ and S must be disjoint, because of the assumption on S. Finally, as we observed in the previous paragraph, if $S\{\alpha\} \cap S\{\beta\} \neq \emptyset$, $S\{\alpha\} \not\subseteq S$, $S\{\beta\} \not\subseteq S$, and α and β are both minimal, then $\alpha = \beta$.

Finally, we check that this partition is right invariant. Because of the assumption on S, it is sufficient to show that if $x, y \in S\{\alpha\}$, and $xz \in S\{\beta\}$, where α and β are minimal as defined above, then yz is also in $S\{\beta\}$. We know that $x = s_1\alpha$ and $y = s_2\alpha$, for some $s_1, s_2 \in S$; therefore, xz and yz are both in $S\{\alpha z\}$. Let γ be the longest prefix of αz satisfying $S\{\gamma\} \subseteq S$. Then our previous discussion shows that $\alpha z = \gamma\beta$, so that $S\{\alpha z\} \subseteq S\{\beta\}$, and $y \in S\{\beta\}$.

(b) In the case where S is finite, the condition reduces to saying that no element of S is a prefix of any other element of S.

(c) If S is finite and satisfies this condition, let M be the maximum length of elements of S. We can consider the following partition of Σ^*: first, all the one-element sets $\{x\}$, where $|x| \leq M$ and no prefix of x is in S; second, the set S; third, the single set T containing all other strings. This is obviously a finite partition. It is right invariant, because of the assumption on S and the fact that for any $x \in T$ and any string y, $xy \in T$.

(d) If S satisfies the condition in (a), then S is one of the subsets of a finite right invariant partition if and only if S is regular. On the one hand, if P is a finite right invariant partition, then there is an FA M so that the states of M correspond to the subsets in P, each of which is therefore regular. On the other hand, if S is regular, let M be a minimum-state FA accepting S. The subsets L_p, for the states p of M, form a finite right invariant partition, and L is the union of all the L_p's for which p is accepting. However, the condition in A implies (by Lemma 5.2) that $p \equiv q$ for any two accepting states p and q; therefore, since M is minimal, there is only one accepting state. (Of course, at this point, we have several other ways to characterize regular sets.)

5.40. Suppose $M = (Q, \Sigma, q_0, A, \delta)$ is an FA accepting L_1. Then for a string $x \in \Sigma^*$, $x \in L_1/L_2$ if and only if there is a string $y \in L_2$ with $xy \in L$; and this is true if and only if there is a string $y \in L_2$ so that $\delta^*(q_0, xy) = \delta^*(\delta^*(q_0, x), y) \in A$. With this in mind, we define $B = \{q \in Q \mid \text{ for some } y \in L_2, \delta^*(q, y) \in A\}$. Then we have observed that for any x, $x \in L_1/L_2$ if and only if $\delta^*(q_0, x) \in B$. In other words, the FA $M' = (Q, \Sigma, q_0, B, \delta)$ accepts the language L_1/L_2.

Notice that the argument does not really use the fact that L_2 is regular. However, if L_2 is not regular, there may be no way to determine exactly which states are in the set B. If L_2 is regular, on the other hand, there is an algorithm to do this. We do not present all the details, but the idea is that for $q \in Q$, we can determine for each $r \in A$ the language $L(q, r) = \{x \in \Sigma^* \mid \delta^*(q, x) = r\}$, and we can then determine whether the intersection $L(q, r) \cap L_2$ is nonempty. (See the proof of Theorem 4.5, and the discussion in Section 5.4.)

5.41. Using the paintcan analogy, we observe that with p distinct primary colors, there are 2^p possible combinations that can be used to create distinct colors, since there are 2^p subsets of a set of p elements. If we want the number of distinct colors to be at least n, then p must satisfy $2^p \geq n$. Therefore, $p \geq \log_2 n$. We conclude that in order to distinguish all possible pairs from among n strings, at least $\log_2 n$ strings are required.

5.42. For $z = a_1 a_2 \ldots a_n$, let $p_i = \delta^*(p, a_1 \ldots a_i)$ and $q_i = \delta^*(q, a_1 \ldots a_i)$, for each i with $1 \leq i \leq n$. We observe that if z has the property we want (that is, exactly one of the two states $\delta^*(p, z)$ and $\delta^*(q, z)$ is in A), and if the pairs (p_i, q_i) and $(p_{i+d} q_{i+d})$ are identical, then we can shorten z by omitting the substring $a_{i+1} a_{i+2} a_{i+d}$, and the shortened string will still have the desired property. Therefore, we may assume that all the pairs (p_i, q_i) are distinct, for $1 \leq i \leq n$, and thus that n is no larger than the maximum number of distinct ordered pairs of states, which is s^2 (where $s = |Q|$).

5.43. If L is regular, and M is an s-state FA accepting L, then any two strings x and y distinguishable with respect to L can be distinguished by a string of length s^2 or less. (We

let $p = \delta^*(q_0, x)$ and $q = \delta^*(q_0, y)$ and use the result of Exercise 5.42.) On the other hand, if there is an n so that any two strings distinguishable with respect to L can be distinguished by a string of length n or less, then L must be regular. (According to Exercise 5.41, using only the finite set S of strings of length $\leq n$, it is impossible to have an infinite set of strings, any two of which can be distinguished by an element of S.)

5.44. As is pointed out in the statement of the problem, each state in either FA corresponds to one of the equivalence classes of I_L. This means that the L_q partition determined by one of the FA's is the same as that determined by the other. So the bijection $i : Q_1 \to Q_2$ is described as follows: if $q \in Q_1$, and $\delta_1^*(q_1, x) = q$, then $i(q) = \delta_2^*(q_2, x)$.

We must show these three things:
(i) $i(q_1) = q_2$; (ii) For any $q \in Q_1$, $q \in A_1$ if and only if $i(q) \in A_2$; and (iii) For any $q \in Q_1$ and any $a \in \Sigma$, $i(\delta_1(q, a)) = \delta_2(i(q), a)$.

Statement (i) follows from our definition of i and the fact that $\delta_1^*(q_1, \Lambda) = q_1 = \delta_2^*(q_2, \Lambda)$. To show (ii), we take a state $q \in A_1$ and a string x such that $\delta_1^*(q_1, x) = q$. Then $\delta_2^*(q_2, x) = i(q) \in A_2$, because the string x is in L (since M_1 accepts L). The argument is also reversible: If $i(q) \in A_2$, then $x \in L$ because M_2 accepts L, so that $q \in A_1$. Finally, for $q \in Q_1$ and $a \in \Sigma$, if $\delta_1^*(q_1, x) = q$, then

$$i(\delta_1(q, a)) = i(\delta_1^*(q_1, xa)) = \delta_2^*(q_2, xa) = \delta_2(\delta_2^*(q_2, x), a) = \delta_2(i(q), a)$$

5.46. (a) This follows immediately from the fact that if M is any FA accepting L relative to L_{i+1}, then M accepts L relative to L_i.

(b) Suppose there is a constant N so that $n_i \leq N$ for every i. Let M_i be an FA with n_i states that accepts L relative to L_i. We may assume that the states of each M_i are elements of the set $\{q_0, q_1, \ldots, q_{N_1}\}$. Therefore, there must be infinitely many i's for which the FAs M_i are all equal, say to M. Then M must accept L relative to L_i for every i; therefore, since $\cup_{i=1}^{\infty} L_i = \Sigma^*$, M accepts L, which is assumed to be nonregular.

5.47. Let n be the number of states in an FA $M = (Q, \Sigma, q_0, A, \delta)$ accepting L, and suppose $x \in L$ and $x = x_1x_2x_3 = x_1(a_1a_2 \ldots a_k)x_3$, where $k \geq n$. Denote by p_i the state $\delta^*(q_0, x_1a_1a_2 \ldots a_i)$ for each i with $0 \leq i \leq k$. Then just as in the proof of the pumping lemma, at least two of the states p_i must be the same; suppose $p_i = p_{i+j}$, where $j > 0$. Then let $u = a_1 \ldots a_i$, $v = a_{i+1} \ldots a_{i+j}$, and $w = a_{i+j+1} \ldots a_k$. We have $x_2 = uvw$ and $|v| > 0$, and the same argument as in the proof of the pumping lemma shows that $x_1uv^mwx_3 \in L$ for every $m \geq 0$.

5.48. The language L in Example 5.11 is such a language. Suppose L is regular, let n be the integer in the statement, and let $x = ab^nc^n$, and let $x_1 = a$, $x_2 = b^n$, and $x_3 = c^n$. The statement says that for some u, v, and w with $|uv| \leq n$ and $|v| > 0$, $b^n = uvw$ and $auv^mwc^n \in L$ for every $m \geq 0$. This would mean that there are strings ab^ic^n in L for which $i \neq n$, which contradicts the definition of L.

5.50. (a) Nonregular. We use the pumping lemma to show L' is nonregular, and it will follow that L is also. Suppose L' is regular, let n be the integer in the pumping lemma, and let x be a string in L' with $|x| \geq n$. (The fact that there is such an x follows from the parenthetical statement in the exercise.) Then $x = uvw$ for some strings u, v, w such that $|v| > 0$ and $uv^iw \in L'$ for every $i \geq 0$. In particular, $uv^3w \in L'$. However, this string contains three consecutive occurrences of v and is therefore an element of L by definition.

(b) Regular. There is an FA recognizing L having 6 states. Five of the states correspond to the five possible values of $n_0(x) - n_1(x)$ between -2 and 2, and the sixth is the "dead" state for strings not in L.

(c) Nonregular. Any two of the strings 1, 1^2, 1^3, ..., say 1^i and 1^j (where $i < j$), can be distinguished with respect to L by the string 0^{i+3}.

(d) Regular. One way to construct an FA is to have states corresponding to the possible pairs ($n_0(x)$ mod 5, $n_1(x)$ mod 5), 25 states in all. The initial state, which is the only accepting state, is $(0, 0)$.

(e) Nonregular. For each $i \geq 1$, let $x_i = 0^{p_i}$, where p_i is the ith prime ($p_1 = 2$, $p_2 = 3$, $p_3 = 5$, ...). If $i \neq j$, then x_i and x_j are distinguished by the string 1^{p_i}.

(f) Regular. Let $M = (Q, \{0, 1\}, q_0, A, \delta)$ be an FA accepting L, and let $M_1 = (Q, \{0, 1\}, q_0, A_1, \delta)$, where A_1 is the set of states q in A for which there is no string z satisfying $\delta^*(q, z) \in A$. Then M_1 accepts Max(L).

(g) Regular. Let $M = (Q, \{0, 1\}, q_0, A, \delta)$ be an FA accepting L, and let $M_1 = (Q, \{0, 1\}, q_0, A_1, \delta)$, where A_1 is the set of states q in A for which there do not exist strings w and z satisfying $|z| > 0$, $\delta^*(q_0, w) \in A$, and $\delta^*(q_0, wz) = q$. In other words, A_1 is the set of states q in A for which no path from q_0 to q reaches an accepting state until the last step. Then M_1 accepts Min(L).

5.51. (b) Notice that this definition of arithmetic progression does not require the integer p to be positive; if it did, every arithmetic progression would be an infinite set, and the result would not be true. Suppose A is accepted by the FA $M = (Q, \{0\}, q_0, F, \delta)$. Since Q is finite, there are integers m and p so that $p > 0$ and $\delta^*(q_0, 0^m) = \delta^*(q_0, 0^{m+p})$. It follows that for every $n \geq m$, $\delta^*(q_0, 0^n) = \delta^*(q_0, 0^{n+p})$. In particular, for every n satisfying $m \leq n < m + p$, if $0^n \in A$, then $0^{n+ip} \in A$ for every $i \geq 0$, and if $0^n \notin A$, then $0^{n+ip} \notin A$ for every $i \geq 0$. If n_1, n_2, ..., n_r are the values of n between m and $m + p - 1$ for which $0^n \in A$, and P_1, ..., P_r are the corresponding arithmetic progressions (that is, $P_j = \{n_j + ip \mid i \geq 0\}$), then S is the union of the sets P_1, ..., P_r and the finite set of all k with $0 \leq k < m$ for which $0^k \in A$. Therefore, S is the union of a finite number of arithmetic progressions.

5.52. (c) We show the result in the case where f is periodic—i.e., where $f(n) = f(n + p)$ for every n. It is sufficient to show that the equivalence relation I_L has only a finite number of equivalence classes, and to do that it is sufficient to find a finite set S of strings so that any string is indistinguishable with respect to L from some element of S. Let M be the maximum value taken by the function f. For each i with $0 \leq i < p$ and each j with $0 \leq j \leq M$, let $x_{i,j} = b^ia^j$, and let $x_0 = a^{M+1}$. Now let x be any string. If $n_a(x) > M$, then xz cannot be in L for any z, and so xI_Lx_0. Otherwise, let $n_b(x) = i$ and $n_a(x) = j$. Clearly $xI_Lb^ia^j$, since the order of the symbols in a string is irrelevant as far as whether

the string is in L. Moreover, since $f(n) = f(n+p)$ for every n, the only significant thing about the number of b's is the the value mod p; therefore, $xI_L x_{i_1,j}$, where $i_1 = i$ mod p.

(d) Suppose L is regular, and let n be the integer in the pumping lemma. Consider the string $x = b^m a^{f(m)}$, where m is any integer with $m \geq n$. By the pumping lemma, $x = uvw$, where $|uv| \leq n$, $|v| > 0$, and $uv^i w \in L$ for every $i \geq 0$. Since $m \geq n$, $v = b^p$ for some $p > 0$. This means that each of the strings $b^{m+ip}a^{f(m}$ is in L. It follows from the definition of L that $f(m+ip) = f(m)$ for every $i \geq 0$. To summarize: for every $m \geq n$, there is a number p_m so that $0 < p_m \leq n$ and $f(m+ip) = f(m)$ for every $i \geq 0$. Now there are only a finite number of integers p that can be p_m for some $m \geq n$, since for any such p_m, $p_m \leq n$. Let P be the least common multiple of all of them—i.e., the smallest positive integer that is evenly divisible by all of them. Then for any $m \geq n$, $P = kp_m$ for some k. Therefore, for any $i \geq 0$, $f(m+iP) = f(m+(ik)p_m) = f(m)$. Therefore, f is eventually periodic.

5.53. Let L be the set $\{0^n \mid n > 0 \text{ and } n \text{ is not prime}\}$. L^2 contains all sufficiently long strings of 0's, because every sufficiently large integer (7 or greater, say) is the sum of two positive nonprimes. (If n is even, n is the sum of two even numbers, each 4 or larger; if n is odd, n is $1+(n-1)$.) Therefore, L^2 is regular. However, L is not, by Example 5.10.

5.54. Let D be the set of integers that are lengths of strings in L^*. Then D is a subset of $\mathcal{N}$ that is closed under addition. We will show this implies that D is the union of a finite number of arithmetic progressions, and the result we want will follow (see Exercise 5.51).

Let p be the smallest positive integer so that p is an integer combination of elements of D (i.e., of the form $\sum_{i=1}^k a_i n_i$, where the a_i's are integers, either positive or negative, and the n_i's are elements of D). We show first that the set of all integer combinations of elements of D is precisely the set of all integer multiples of p. It is clear that every multiple of p has this form. On the other hand, suppose that j is an integer combination of elements of D but not a multiple of p. We can divide j by p, and get a quotient and a remainder: that is, $j = qp + r$, where $0 \leq r < p$. But since j and p are both integer combinations of elements of D, so is r. However, p is defined to be the smallest positive one, so $r = 0$; this contradicts the fact that j is not divisible by p. At this point, we have $p = \sum_{i=1}^k a_i n_i$, where each $n_i \in D$; we know that every integer combination of the n_i's is a multiple of p; and we know that every integer combination of the n_i's for which the coefficients are all nonnegative is itself in D (since D is closed under addition). Finally, since n_1 is a multiple of p, let $n_1 = m_1 p$. We can now show that all sufficiently large multiples of p are elements of D. To be specific, let $N = \sum_{i=1}^n m_1 |a_i| n_i$. (Except for the extra factor of m_1, this is just the formula for p, but with any negative coefficients replaced by their absolute values.) We show that for every $j \geq 0$, $N + jp \in D$. It is sufficient to show that $N + jp$ is an integer combination of the n_i's, with nonnegative coefficients.

Let us first consider j satisfying $0 \leq j \leq m_1$. For such a j,

$$N + jp = \sum_{i=1}^{n} (m_1 |a_i| + j a_i) n_i$$

and all the coefficients of this sum are nonnegative. This takes care of all multiples of p

with values between N and $N+n_1$. However, it is clear that the same argument will work for the multiples of p between $N+rn_1$ and $N+(r+1)n_1$, where r is any natural number. We have shown that $\{n \in D \mid n \geq N\} = \{N + jp \mid j \geq 0\}$. We conclude that D is the union of a finite set and an arithmetic progression, which means that it is the union of a finite number of arithmetic progressions.