Solution: We note that $(x, y) \in R_1 \cup R_2$ if and only if $(x, y) \in R_1$ or $(x, y) \in R_2$. Hence, $(x, y) \in R_1 \cup R_2$ if and only if $x < y$ or $x > y$. Because the condition $x < y$ or $x > y$ is the same as the condition $x \neq y$, it follows that $R_1 \cup R_2 = \{(x, y) \mid x \neq y\}$. In other words, the union of the "less than" relation and the "greater than" relation is the "not equals" relation.

Next, note that it is impossible for a pair (x, y) to belong to both R_1 and R_2 because it is impossible for $x < y$ and $x > y$. It follows that $R_1 \cap R_2 = \emptyset$. We also see that $R_1 - R_2 = R_1$, $R_2 - R_1 = R_2$, and $R_1 \oplus R_2 = R_1 \cup R_2 - R_1 \cap R_2 = \{(x, y) \mid x \neq y\}$. ◀

There is another way that relations are combined that is analogous to the composition of functions.

DEFINITION 6 Let R be a relation from a set A to a set B and S a relation from B to a set C. The *composite* of R and S is the relation consisting of ordered pairs (a, c), where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$. We denote the composite of R and S by $S \circ R$.

Computing the composite of two relations requires that we find elements that are the second element of ordered pairs in the first relation and the first element of ordered pairs in the second relation, as Examples 20 and 21 illustrate.

EXAMPLE 20 What is the composite of the relations R and S, where R is the relation from $\{1, 2, 3\}$ to $\{1, 2, 3, 4\}$ with $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$ and S is the relation from $\{1, 2, 3, 4\}$ to $\{0, 1, 2\}$ with $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$?

Solution: $S \circ R$ is constructed using all ordered pairs in R and ordered pairs in S, where the second element of the ordered pair in R agrees with the first element of the ordered pair in S. For example, the ordered pairs $(2, 3)$ in R and $(3, 1)$ in S produce the ordered pair $(2, 1)$ in $S \circ R$. Computing all the ordered pairs in the composite, we find

$$S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}.$$ ◀

EXAMPLE 21 **Composing the Parent Relation with Itself** Let R be the relation on the set of all people such that $(a, b) \in R$ if person a is a parent of person b. Then $(a, c) \in R \circ R$ if and only if there is a person b such that $(a, b) \in R$ and $(b, c) \in R$, that is, if and only if there is a person b such that a is a parent of b and b is a parent of c. In other words, $(a, c) \in R \circ R$ if and only if a is a grandparent of c. ◀

The powers of a relation R can be recursively defined from the definition of a composite of two relations.

DEFINITION 7 Let R be a relation on the set A. The powers R^n, $n = 1, 2, 3, \ldots$, are defined recursively by

$$R^1 = R \quad \text{and} \quad R^{n+1} = R^n \circ R.$$

The definition shows that $R^2 = R \circ R$, $R^3 = R^2 \circ R = (R \circ R) \circ R$, and so on.

EXAMPLE 22 Let $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$. Find the powers R^n, $n = 2, 3, 4, \ldots$.

Solution: Because $R^2 = R \circ R$, we find that $R^2 = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$. Furthermore, because $R^3 = R^2 \circ R$, $R^3 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. Additional computation shows that R^4 is the same as R^3, so $R^4 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. It also follows that $R^n = R^3$ for $n = 5, 6, 7, \ldots$. The reader should verify this. ◀

The following theorem shows that the powers of a transitive relation are subsets of this relation. It will be used in Section 8.4.

THEOREM 1 The relation R on a set A is transitive if and only if $R^n \subseteq R$ for $n = 1, 2, 3, \ldots$.

Proof: We first prove the "if" part of the theorem. We suppose that $R^n \subseteq R$ for $n = 1, 2, 3, \ldots$. In particular, $R^2 \subseteq R$. To see that this implies R is transitive, note that if $(a, b) \in R$ and $(b, c) \in R$, then by the definition of composition, $(a, c) \in R^2$. Because $R^2 \subseteq R$, this means that $(a, c) \in R$. Hence, R is transitive.

We will use mathematical induction to prove the only if part of the theorem. Note that this part of the theorem is trivially true for $n = 1$.

Assume that $R^n \subseteq R$, where n is a positive integer. This is the inductive hypothesis. To complete the inductive step we must show that this implies that R^{n+1} is also a subset of R. To show this, assume that $(a, b) \in R^{n+1}$. Then, because $R^{n+1} = R^n \circ R$, there is an element x with $x \in A$ such that $(a, x) \in R$ and $(x, b) \in R^n$. The inductive hypothesis, namely, that $R^n \subseteq R$, implies that $(x, b) \in R$. Furthermore, because R is transitive, and $(a, x) \in R$ and $(x, b) \in R$, it follows that $(a, b) \in R$. This shows that $R^{n+1} \subseteq R$, completing the proof. ◁

Exercises

1. List the ordered pairs in the relation R from $A = \{0, 1, 2, 3, 4\}$ to $B = \{0, 1, 2, 3\}$, where $(a, b) \in R$ if and only if
   a) $a = b$.
   b) $a + b = 4$.
   c) $a > b$.
   d) $a \mid b$.
   e) $\gcd(a, b) = 1$.
   f) $\operatorname{lcm}(a, b) = 2$.
2. a) List all the ordered pairs in the relation $R = \{(a, b) \mid a \text{ divides } b\}$ on the set $\{1, 2, 3, 4, 5, 6\}$.
   b) Display this relation graphically, as was done in Example 4.
   c) Display this relation in tabular form, as was done in Example 4.
3. For each of these relations on the set $\{1, 2, 3, 4\}$, decide whether it is reflexive, whether it is symmetric, whether it is antisymmetric, and whether it is transitive.
   a) $\{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$
   b) $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$
   c) $\{(2, 4), (4, 2)\}$
   d) $\{(1, 2), (2, 3), (3, 4)\}$
   e) $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
   f) $\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}$
4. Determine whether the relation R on the set of all people is reflexive, symmetric, antisymmetric, and/or transitive, where $(a, b) \in R$ if and only if
   a) a is taller than b.
   b) a and b were born on the same day.
   c) a has the same first name as b.
   d) a and b have a common grandparent.
5. Determine whether the relation R on the set of all Web pages is reflexive, symmetric, antisymmetric, and/or transitive, where $(a, b) \in R$ if and only if
   a) everyone who has visited Web page a has also visited Web page b.
   b) there are no common links found on both Web page a and Web page b.
   c) there is at least one common link on Web page a and Web page b.
   d) there is a Web page that includes links to both Web page a and Web page b.
6. Determine whether the relation R on the set of all real numbers is reflexive, symmetric, antisymmetric, and/or transitive, where $(x, y) \in R$ if and only if

a) $x + y = 0$.
b) $x = \pm y$.
c) $x - y$ is a rational number.
d) $x = 2y$.
e) $xy \geq 0$.
f) $xy = 0$.
g) $x = 1$.
h) $x = 1$ or $y = 1$.

7. Determine whether the relation R on the set of all integers is reflexive, symmetric, antisymmetric, and/or transitive, where $(x, y) \in R$ if and only if

a) $x \neq y$.
b) $xy \geq 1$.
c) $x = y + 1$ or $x = y - 1$.
d) $x \equiv y \pmod 7$.
e) x is a multiple of y.
f) x and y are both negative or both nonnegative.
g) $x = y^2$.
h) $x \geq y^2$.

8. Give an example of a relation on a set that is

a) symmetric and antisymmetric.
b) neither symmetric nor antisymmetric.

A relation R on the set A is **irreflexive** if for every $a \in A, (a, a) \notin R$. That is, R is irreflexive if no element in A is related to itself.

9. Which relations in Exercise 3 are irreflexive?

10. Which relations in Exercise 4 are irreflexive?

11. Which relations in Exercise 5 are irreflexive?

12. Which relations in Exercise 6 are irreflexive?

13. Can a relation on a set be neither reflexive nor irreflexive?

14. Use quantifiers to express what it means for a relation to be irreflexive.

15. Give an example of an irreflexive relation on the set of all people.

A relation R is called **asymmetric** if $(a, b) \in R$ implies that $(b, a) \notin R$. Exercises 16–22 explore the notion of an asymmetric relation. Exercise 20 focuses on the difference between asymmetry and antisymmetry.

16. Which relations in Exercise 3 are asymmetric?

17. Which relations in Exercise 4 are asymmetric?

18. Which relations in Exercise 5 are asymmetric?

19. Which relations in Exercise 6 are asymmetric?

20. Must an asymmetric relation also be antisymmetric? Must an antisymmetric relation be asymmetric? Give reasons for your answers.

21. Use quantifiers to express what it means for a relation to be asymmetric.

22. Give an example of an asymmetric relation on the set of all people.

23. How many different relations are there from a set with m elements to a set with n elements?

Let R be a relation from a set A to a set B. The **inverse relation** from B to A, denoted by R^{-1}, is the set of ordered pairs $\{(b, a) \mid (a, b) \in R\}$. The **complementary relation** $\overline{R}$ is the set of ordered pairs $\{(a, b) \mid (a, b) \notin R\}$.

24. Let R be the relation $R = \{(a, b) \mid a < b\}$ on the set of integers. Find

a) R^{-1}.
b) $\overline{R}$.

25. Let R be the relation $R = \{(a, b) \mid a \text{ divides } b\}$ on the set of positive integers. Find

a) R^{-1}.
b) $\overline{R}$.

26. Let R be the relation on the set of all states in the United States consisting of pairs (a, b) where state a borders state b. Find

a) R^{-1}.
b) $\overline{R}$.

27. Suppose that the function f from A to B is a one-to-one correspondence. Let R be the relation that equals the graph of f. That is, $R = \{(a, f(a)) \mid a \in A\}$. What is the inverse relation R^{-1}?

28. Let $R_1 = \{(1, 2), (2, 3), (3, 4)\}$ and $R_2 = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4)\}$ be relations from $\{1, 2, 3\}$ to $\{1, 2, 3, 4\}$. Find

a) $R_1 \cup R_2$.
b) $R_1 \cap R_2$.
c) $R_1 - R_2$.
d) $R_2 - R_1$.

29. Let A be the set of students at your school and B the set of books in the school library. Let R_1 and R_2 be the relations consisting of all ordered pairs (a, b), where student a is required to read book b in a course, and where student a has read book b, respectively. Describe the ordered pairs in each of these relations.

a) $R_1 \cup R_2$
b) $R_1 \cap R_2$
c) $R_1 \oplus R_2$
d) $R_1 - R_2$
e) $R_2 - R_1$

30. Let R be the relation $\{(1, 2), (1, 3), (2, 3), (2, 4), (3, 1)\}$, and let S be the relation $\{(2, 1), (3, 1), (3, 2), (4, 2)\}$. Find $S \circ R$.

31. Let R be the relation on the set of people consisting of pairs (a, b), where a is a parent of b. Let S be the relation on the set of people consisting of pairs (a, b), where a and b are siblings (brothers or sisters). What are $S \circ R$ and $R \circ S$?

Exercises 32–35 deal with these relations on the set of real numbers:

$R_1 = \{(a, b) \in \mathbf{R}^2 \mid a > b\}$, the "greater than" relation,

$R_2 = \{(a, b) \in \mathbf{R}^2 \mid a \geq b\}$, the "greater than or equal to" relation,

$R_3 = \{(a, b) \in \mathbf{R}^2 \mid a < b\}$, the "less than" relation,

$R_4 = \{(a, b) \in \mathbf{R}^2 \mid a \leq b\}$, the "less than or equal to" relation,

$R_5 = \{(a, b) \in \mathbf{R}^2 \mid a = b\}$, the "equal to" relation,

$R_6 = \{(a, b) \in \mathbf{R}^2 \mid a \neq b\}$, the "unequal to" relation.

32. Find

a) $R_1 \cup R_3$.
b) $R_1 \cup R_5$.
c) $R_2 \cap R_4$.
d) $R_3 \cap R_5$.
e) $R_1 - R_2$.
f) $R_2 - R_1$.
g) $R_1 \oplus R_3$.
h) $R_2 \oplus R_4$.

33. Find

a) $R_2 \cup R_4$.
b) $R_3 \cup R_6$.
c) $R_3 \cap R_6$.
d) $R_4 \cap R_6$.
e) $R_3 - R_6$.
f) $R_6 - R_3$.
g) $R_2 \oplus R_6$.
h) $R_3 \oplus R_5$.

34. Find
a) $R_1 \circ R_1$.
b) $R_1 \circ R_2$.
c) $R_1 \circ R_3$.
d) $R_1 \circ R_4$.
e) $R_1 \circ R_5$.
f) $R_1 \circ R_6$.
g) $R_2 \circ R_3$.
h) $R_3 \circ R_3$.

35. Find
a) $R_2 \circ R_1$.
b) $R_2 \circ R_2$.
c) $R_3 \circ R_5$.
d) $R_4 \circ R_1$.
e) $R_5 \circ R_3$.
f) $R_3 \circ R_6$.
g) $R_4 \circ R_6$.
h) $R_6 \circ R_6$.

36. Let R be the parent relation on the set of all people (see Example 21). When is an ordered pair in the relation R^3?

37. Let R be the relation on the set of people with doctorates such that $(a, b) \in R$ if and only if a was the thesis advisor of b. When is an ordered pair (a, b) in R^2? When is an ordered pair (a, b) in R^n, when n is a positive integer? (Note that every person with a doctorate has a thesis advisor.)

38. Let R_1 and R_2 be the "divides" and "is a multiple of" relations on the set of all positive integers, respectively. That is, $R_1 = \{(a, b) \mid a \text{ divides } b\}$ and $R_2 = \{(a, b) \mid a$ is a multiple of $b\}$. Find
a) $R_1 \cup R_2$.
b) $R_1 \cap R_2$.
c) $R_1 - R_2$.
d) $R_2 - R_1$.
e) $R_1 \oplus R_2$.

39. Let R_1 and R_2 be the "congruent modulo 3" and the "congruent modulo 4" relations, respectively, on the set of integers. That is, $R_1 = \{(a, b) \mid a \equiv b \pmod 3\}$ and $R_2 = \{(a, b) \mid a \equiv b \pmod 4\}$. Find
a) $R_1 \cup R_2$.
b) $R_1 \cap R_2$.
c) $R_1 - R_2$.
d) $R_2 - R_1$.
e) $R_1 \oplus R_2$.

40. List the 16 different relations on the set $\{0, 1\}$.

41. How many of the 16 different relations on $\{0, 1\}$ contain the pair $(0, 1)$?

42. Which of the 16 relations on $\{0, 1\}$, which you listed in Exercise 40, are
a) reflexive?
b) irreflexive?
c) symmetric?
d) antisymmetric?
e) asymmetric?
f) transitive?

43. a) How many relations are there on the set $\{a, b, c, d\}$?
b) How many relations are there on the set $\{a, b, c, d\}$ that contain the pair (a, a)?

44. Let S be a set with n elements and let a and b be distinct elements of S. How many relations are there on S such that
a) $(a, b) \in S$?
b) $(a, b) \notin S$?
c) there are no ordered pairs in the relation that have a as their first element?
d) there is at least one ordered pair in the relation that has a as its first element?
e) there are no ordered pairs in the relation that have a as their first element and there are no ordered pairs in the relation that have b as their second element?
f) there is at least one ordered pair in the relation that either has a as its first element or has b as its second element?

*45. How many relations are there on a set with n elements that are
a) symmetric?
b) antisymmetric?
c) asymmetric?
d) irreflexive?
e) reflexive and symmetric?
f) neither reflexive nor irreflexive?

*46. How many transitive relations are there on a set with n elements if
a) $n = 1$?
b) $n = 2$?
c) $n = 3$?

47. Find the error in the "proof" of the following "theorem."
"Theorem": Let R be a relation on a set A that is symmetric and transitive. Then R is reflexive.
"Proof": Let $a \in A$. Take an element $b \in A$ such that $(a, b) \in R$. Because R is symmetric, we also have $(b, a) \in R$. Now using the transitive property, we can conclude that $(a, a) \in R$ because $(a, b) \in R$ and $(b, a) \in R$.

48. Suppose that R and S are reflexive relations on a set A. Prove or disprove each of these statements.
a) $R \cup S$ is reflexive.
b) $R \cap S$ is reflexive.
c) $R \oplus S$ is irreflexive.
d) $R - S$ is irreflexive.
e) $S \circ R$ is reflexive.

49. Show that the relation R on a set A is symmetric if and only if $R = R^{-1}$, where R^{-1} is the inverse relation.

50. Show that the relation R on a set A is antisymmetric if and only if $R \cap R^{-1}$ is a subset of the diagonal relation $\Delta = \{(a, a) \mid a \in A\}$.

51. Show that the relation R on a set A is reflexive if and only if the inverse relation R^{-1} is reflexive.

52. Show that the relation R on a set A is reflexive if and only if the complementary relation $\overline{R}$ is irreflexive.

53. Let R be a relation that is reflexive and transitive. Prove that $R^n = R$ for all positive integers n.

54. Let R be the relation on the set $\{1, 2, 3, 4, 5\}$ containing the ordered pairs $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(2, 4)$, $(3, 1)$, $(3, 4)$, $(3, 5)$, $(4, 2)$, $(4, 5)$, $(5, 1)$, $(5, 2)$, and $(5, 4)$. Find
a) R^2.
b) R^3.
c) R^4.
d) R^5.

55. Let R be a reflexive relation on a set A. Show that R^n is reflexive for all positive integers n.

*56. Let R be a symmetric relation. Show that R^n is symmetric for all positive integers n.

57. Suppose that the relation R is irreflexive. Is R^2 necessarily irreflexive? Give a reason for your answer.

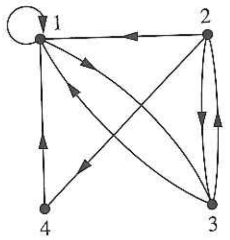


FIGURE 4 The Directed Graph of the Relation *R*.

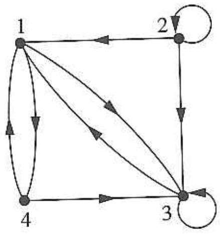
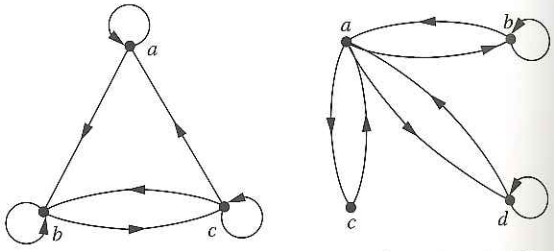


FIGURE 5 The Directed Graph of the Relation *R*.



(a) Directed graph of R (b) Directed graph of S

FIGURE 6 The Directed Graphs of the Relations *R* and *S*.

relation. Similarly, a relation is antisymmetric if and only if there are never two edges in opposite directions between distinct vertices. Finally, a relation is transitive if and only if whenever there is an edge from a vertex x to a vertex y and an edge from a vertex y to a vertex z, there is an edge from x to z (completing a triangle where each side is a directed edge with the correct direction).

Remark: Note that a symmetric relation can be represented by an undirected graph, which is a graph where edges do not have directions. We will study undirected graphs in Chapter 9.

EXAMPLE 10 Determine whether the relations for the directed graphs shown in Figure 6 are reflexive, symmetric, antisymmetric, and/or transitive.

Solution: Because there are loops at every vertex of the directed graph of R, it is reflexive. R is neither symmetric nor antisymmetric because there is an edge from a to b but not one from b to a, but there are edges in both directions connecting b and c. Finally, R is not transitive because there is an edge from a to b and an edge from b to c, but no edge from a to c.

Because loops are not present at all the vertices of the directed graph of S, this relation is not reflexive. It is symmetric and not antisymmetric, because every edge between distinct vertices is accompanied by an edge in the opposite direction. It is also not hard to see from the directed graph that S is not transitive, because (c, a) and (a, b) belong to S, but (c, b) does not belong to S. ◀

Exercises

1. Represent each of these relations on $\{1, 2, 3\}$ with a matrix (with the elements of this set listed in increasing order).

a) $\{(1, 1), (1, 2), (1, 3)\}$
b) $\{(1, 2), (2, 1), (2, 2), (3, 3)\}$
c) $\{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$
d) $\{(1, 3), (3, 1)\}$

2. Represent each of these relations on $\{1, 2, 3, 4\}$ with a matrix (with the elements of this set listed in increasing order).

a) $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$
b) $\{(1, 1), (1, 4), (2, 2), (3, 3), (4, 1)\}$
c) $\{(1, 2), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (3, 4), (4, 1), (4, 2), (4, 3)\}$
d) $\{(2, 4), (3, 1), (3, 2), (3, 4)\}$

3. List the ordered pairs in the relations on $\{1, 2, 3\}$ corresponding to these matrices (where the rows and columns correspond to the integers listed in increasing order).

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

4. List the ordered pairs in the relations on $\{1, 2, 3, 4\}$ corresponding to these matrices (where the rows and columns correspond to the integers listed in increasing order).

a) $\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

5. How can the matrix representing a relation R on a set A be used to determine whether the relation is irreflexive?
6. How can the matrix representing a relation R on a set A be used to determine whether the relation is asymmetric?
7. Determine whether the relations represented by the matrices in Exercise 3 are reflexive, irreflexive, symmetric, antisymmetric, and/or transitive.
8. Determine whether the relations represented by the matrices in Exercise 4 are reflexive, irreflexive, symmetric, antisymmetric, and/or transitive.
9. How many nonzero entries does the matrix representing the relation R on $A = \{1, 2, 3, \ldots, 100\}$ consisting of the first 100 positive integers have if R is
   a) $\{(a, b) \mid a > b\}$? b) $\{(a, b) \mid a \neq b\}$?
   c) $\{(a, b) \mid a = b + 1\}$? d) $\{(a, b) \mid a = 1\}$?
   e) $\{(a, b) \mid ab = 1\}$?
10. How many nonzero entries does the matrix representing the relation R on $A = \{1, 2, 3, \ldots, 1000\}$ consisting of the first 1000 positive integers have if R is
   a) $\{(a, b) \mid a \leq b\}$?
   b) $\{(a, b) \mid a = b \pm 1\}$?
   c) $\{(a, b) \mid a + b = 1000\}$?
   d) $\{(a, b) \mid a + b \leq 1001\}$?
   e) $\{(a, b) \mid a \neq 0\}$?
11. How can the matrix for $\overline{R}$, the complement of the relation R, be found from the matrix representing R, when R is a relation on a finite set A?
12. How can the matrix for R^{-1}, the inverse of the relation R, be found from the matrix representing R, when R is a relation on a finite set A?
13. Let R be the relation represented by the matrix

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

   Find the matrix representing
   a) R^{-1}. b) $\overline{R}$. c) R^2.
14. Let R_1 and R_2 be relations on a set A represented by the matrices

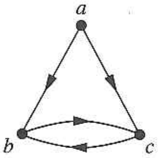
$$\mathbf{M}_{R_1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{M}_{R_2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

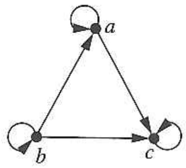
   Find the matrices that represent
   a) $R_1 \cup R_2$. b) $R_1 \cap R_2$. c) $R_2 \circ R_1$.
   d) $R_1 \circ R_1$. e) $R_1 \oplus R_2$.
15. Let R be the relation represented by the matrix

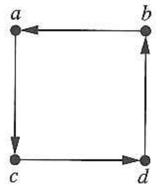
$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

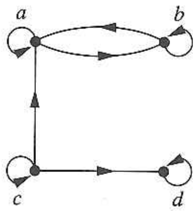
   Find the matrices that represent
   a) R^2. b) R^3. c) R^4.
16. Let R be a relation on a set A with n elements. If there are k nonzero entries in $\mathbf{M}_R$, the matrix representing R, how many nonzero entries are there in $\mathbf{M}_{R^{-1}}$, the matrix representing R^{-1}, the inverse of R?
17. Let R be a relation on a set A with n elements. If there are k nonzero entries in $\mathbf{M}_R$, the matrix representing R, how many nonzero entries are there in $\mathbf{M}_{\overline{R}}$, the matrix representing $\overline{R}$, the complement of R?
18. Draw the directed graphs representing each of the relations from Exercise 1.
19. Draw the directed graphs representing each of the relations from Exercise 2.
20. Draw the directed graph representing each of the relations from Exercise 3.
21. Draw the directed graph representing each of the relations from Exercise 4.
22. Draw the directed graph that represents the relation $\{(a, a), (a, b), (b, c), (c, b), (c, d), (d, a), (d, b)\}$.

In Exercises 23–28 list the ordered pairs in the relations represented by the directed graphs.

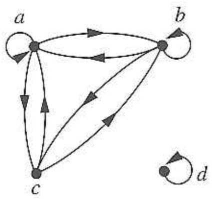
23.


24.


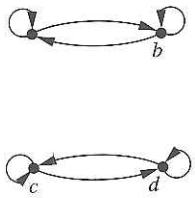
25.


26.


27.



28.



29. How can the directed graph of a relation R on a finite set A be used to determine whether a relation is asymmetric?

30. How can the directed graph of a relation R on a finite set A be used to determine whether a relation is irreflexive?

31. Determine whether the relations represented by the directed graphs shown in Exercises 23–25 are reflexive, irreflexive, symmetric, antisymmetric, and/or transitive.

32. Determine whether the relations represented by the directed graphs shown in Exercises 26–28 are reflexive, irreflexive, symmetric, antisymmetric, asymmetric, and/or transitive.

33. Let R be a relation on a set A. Explain how to use the directed graph representing R to obtain the directed graph representing the inverse relation R^{-1}.

34. Let R be a relation on a set A. Explain how to use the directed graph representing R to obtain the directed graph representing the complementary relation $\overline{R}$.

35. Show that if $\mathbf{M}_R$ is the matrix representing the relation R, then $\mathbf{M}_R^{[n]}$ is the matrix representing the relation R^n.

36. Given the directed graphs representing two relations, how can the directed graph of the union, intersection, symmetric difference, difference, and composition of these relations be found?

8.4 Closures of Relations

Introduction

A computer network has data centers in Boston, Chicago, Denver, Detroit, New York, and San Diego. There are direct, one-way telephone lines from Boston to Chicago, from Boston to Detroit, from Chicago to Detroit, from Detroit to Denver, and from New York to San Diego. Let R be the relation containing (a, b) if there is a telephone line from the data center in a to that in b. How can we determine if there is some (possibly indirect) link composed of one or more telephone lines from one center to another? Because not all links are direct, such as the link from Boston to Denver that goes through Detroit, R cannot be used directly to answer this. In the language of relations, R is not transitive, so it does not contain all the pairs that can be linked. As we will show in this section, we can find all pairs of data centers that have a link by constructing a transitive relation S containing R such that S is a subset of every transitive relation containing R. Here, S is the smallest transitive relation that contains R. This relation is called the **transitive closure** of R.

In general, let R be a relation on a set A. R may or may not have some property **P**, such as reflexivity, symmetry, or transitivity. If there is a relation S with property **P** containing R such that S is a subset of every relation with property **P** containing R, then S is called the **closure** of R with respect to **P**. (Note that the closure of a relation with respect to a property may not exist; see Exercises 15 and 35 at the end of this section.) We will show how reflexive, symmetric, and transitive closures of relations can be found.

Closures

The relation $R = \{(1, 1), (1, 2), (2, 1), (3, 2)\}$ on the set $A = \{1, 2, 3\}$ is not reflexive. How can we produce a reflexive relation containing R that is as small as possible? This can be done by adding $(2, 2)$ and $(3, 3)$ to R, because these are the only pairs of the form (a, a) that are not in R. Clearly, this new relation contains R. Furthermore, *any* reflexive relation that contains R must also contain $(2, 2)$ and $(3, 3)$. Because this relation contains R, is reflexive, and is contained within every reflexive relation that contains R, it is called the **reflexive closure** of R.

As this example illustrates, given a relation R on a set A, the reflexive closure of R can be formed by adding to R all pairs of the form (a, a) with $a \in A$, not already in R. The

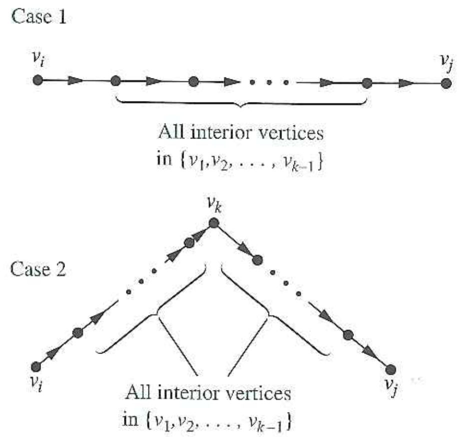


FIGURE 4 Adding v_k to the Set of Allowable Interior Vertices.

Lemma 2 gives us the means to compute efficiently the matrices $\mathbf{W}_k$, $k = 1, 2, \ldots, n$. We display the pseudocode for Warshall's algorithm, using Lemma 2, as Algorithm 2.

ALGORITHM 2 **Warshall Algorithm.**

procedure *Warshall* ($\mathbf{M}_R$: $n \times n$ zero–one matrix)
$\mathbf{W} := \mathbf{M}_R$
for $k := 1$ **to** n
begin
  for $i := 1$ **to** n
  begin
    for $j := 1$ **to** n
    $w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$
  end
end $\{\mathbf{W} = [w_{ij}]$ is $\mathbf{M}_{R^*}\}$

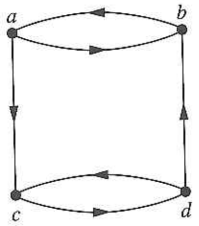
The computational complexity of Warshall's algorithm can easily be computed in terms of bit operations. To find the entry $w_{ij}^{[k]}$ from the entries $w_{ij}^{[k-1]}$, $w_{ik}^{[k-1]}$, and $w_{kj}^{[k-1]}$ using Lemma 2 requires two bit operations. To find all n^2 entries of $\mathbf{W}_k$ from those of $\mathbf{W}_{k-1}$ requires $2n^2$ bit operations. Because Warshall's algorithm begins with $\mathbf{W}_0 = \mathbf{M}_R$ and computes the sequence of n zero–one matrices $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_n = \mathbf{M}_{R^*}$, the total number of bit operations used is $n \cdot 2n^2 = 2n^3$.

Exercises

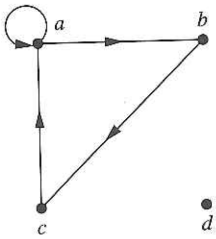
1. Let R be the relation on the set $\{0, 1, 2, 3\}$ containing the ordered pairs $(0, 1)$, $(1, 1)$, $(1, 2)$, $(2, 0)$, $(2, 2)$, and $(3, 0)$. Find the
   a) reflexive closure of R. **b)** symmetric closure of R.
2. Let R be the relation $\{(a, b) \mid a \neq b\}$ on the set of integers. What is the reflexive closure of R?
3. Let R be the relation $\{(a, b) \mid a \text{ divides } b\}$ on the set of integers. What is the symmetric closure of R?
4. How can the directed graph representing the reflexive closure of a relation on a finite set be constructed from the directed graph of the relation?

In Exercises 5–7 draw the directed graph of the reflexive closure of the relations with the directed graph shown.

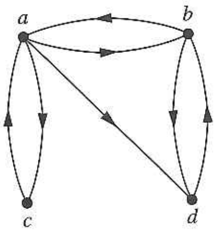
5.



6.



7.



8. How can the directed graph representing the symmetric closure of a relation on a finite set be constructed from the directed graph for this relation?

9. Find the directed graphs of the symmetric closures of the relations with directed graphs shown in Exercises 5–7.

10. Find the smallest relation containing the relation in Example 2 that is both reflexive and symmetric.

11. Find the directed graph of the smallest relation that is both reflexive and symmetric for each of the relations with directed graphs shown in Exercises 5–7.

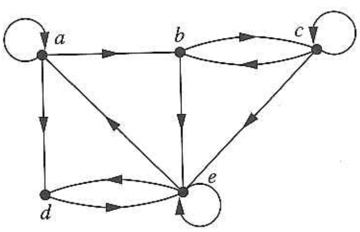
12. Suppose that the relation R on the finite set A is represented by the matrix $\mathbf{M}_R$. Show that the matrix that represents the reflexive closure of R is $\mathbf{M}_R \vee \mathbf{I}_n$.

13. Suppose that the relation R on the finite set A is represented by the matrix $\mathbf{M}_R$. Show that the matrix that represents the symmetric closure of R is $\mathbf{M}_R \vee \mathbf{M}_R^t$.

14. Show that the closure of a relation R with respect to a property **P**, if it exists, is the intersection of all the relations with property **P** that contain R.

15. When is it possible to define the "irreflexive closure" of a relation R, that is, a relation that contains R, is irreflexive, and is contained in every irreflexive relation that contains R?

16. Determine whether these sequences of vertices are paths in this directed graph.



a) a, b, c, e
b) b, e, c, b, e
c) a, a, b, e, d, e
d) b, c, e, d, a, a, b
e) b, c, c, b, e, d, e, d
f) $a, a, b, b, c, c, b, e, d$

17. Find all circuits of length three in the directed graph in Exercise 16.

18. Determine whether there is a path in the directed graph in Exercise 16 beginning at the first vertex given and ending at the second vertex given.

a) a, b **b)** b, a **c)** b, b
d) a, e **e)** b, d **f)** c, d
g) d, d **h)** e, a **i)** e, c

19. Let R be the relation on the set $\{1, 2, 3, 4, 5\}$ containing the ordered pairs $(1, 3)$, $(2, 4)$, $(3, 1)$, $(3, 5)$, $(4, 3)$, $(5, 1)$, $(5, 2)$, and $(5, 4)$. Find

a) R^2. **b)** R^3. **c)** R^4.
d) R^5. **e)** R^6. **f)** R^*.

20. Let R be the relation that contains the pair (a, b) if a and b are cities such that there is a direct non-stop airline flight from a to b. When is (a, b) in

a) R^2? **b)** R^3? **c)** R^*?

21. Let R be the relation on the set of all students containing the ordered pair (a, b) if a and b are in at least one common class and $a \neq b$. When is (a, b) in

a) R^2? **b)** R^3? **c)** R^*?

22. Suppose that the relation R is reflexive. Show that R^* is reflexive.

23. Suppose that the relation R is symmetric. Show that R^* is symmetric.

24. Suppose that the relation R is irreflexive. Is the relation R^2 necessarily irreflexive?

25. Use Algorithm 1 to find the transitive closures of these relations on $\{1, 2, 3, 4\}$.

a) $\{(1, 2), (2, 1), (2, 3), (3, 4), (4, 1)\}$
b) $\{(2, 1), (2, 3), (3, 1), (3, 4), (4, 1), (4, 3)\}$
c) $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$
d) $\{(1, 1), (1, 4), (2, 1), (2, 3), (3, 1), (3, 2), (3, 4), (4, 2)\}$

26. Use Algorithm 1 to find the transitive closures of these relations on $\{a, b, c, d, e\}$.

a) $\{(a, c), (b, d), (c, a), (d, b), (e, d)\}$
b) $\{(b, c), (b, e), (c, e), (d, a), (e, b), (e, c)\}$
c) $\{(a, b), (a, c), (a, e), (b, a), (b, c), (c, a), (c, b), (d, a), (e, d)\}$
d) $\{(a, e), (b, a), (b, d), (c, d), (d, a), (d, c), (e, a), (e, b), (e, c), (e, e)\}$

27. Use Warshall's algorithm to find the transitive closures of the relations in Exercise 25.

28. Use Warshall's algorithm to find the transitive closures of the relations in Exercise 26.

29. Find the smallest relation containing the relation $\{(1, 2), (1, 4), (3, 3), (4, 1)\}$ that is

a) reflexive and transitive.

b) symmetric and transitive.

c) reflexive, symmetric, and transitive.

30. Finish the proof of the case when $a \neq b$ in Lemma 1.

31. Algorithms have been devised that use $O(n^{2.8})$ bit operations to compute the Boolean product of two $n \times n$ zero–one matrices. Assuming that these algorithms can be used, give big-O estimates for the number of bit operations using Algorithm 1 and using Warshall's algorithm to find the transitive closure of a relation on a set with n elements.

***32.** Devise an algorithm using the concept of interior vertices in a path to find the length of the shortest path between two vertices in a directed graph, if such a path exists.

33. Adapt Algorithm 1 to find the reflexive closure of the transitive closure of a relation on a set with n elements.

34. Adapt Warshall's algorithm to find the reflexive closure of the transitive closure of a relation on a set with n elements.

35. Show that the closure with respect to the property **P** of the relation $R = \{(0, 0), (0, 1), (1, 1), (2, 2)\}$ on the set $\{0, 1, 2\}$ does not exist if **P** is the property

a) "is not reflexive."

b) "has an odd number of elements."

8.5 Equivalence Relations

Introduction

In some programming languages the names of variables can contain an unlimited number of characters. However, there is a limit on the number of characters that are checked when a compiler determines whether two variables are equal. For instance, in traditional C, only the first eight characters of a variable name are checked by the compiler. (These characters are uppercase or lowercase letters, digits, or underscores.) Consequently, the compiler considers strings longer than eight characters that agree in their first eight characters the same. Let R be the relation on the set of strings of characters such that sRt, where s and t are two strings, if s and t are at least eight characters long and the first eight characters of s and t agree, or $s = t$. It is easy to see that R is reflexive, symmetric, and transitive. Moreover, R divides the set of all strings into classes, where all strings in a particular class are considered the same by a compiler for traditional C.

The integers a and b are related by the "congruence modulo 4" relation when 4 divides $a - b$. We will show later that this relation is reflexive, symmetric, and transitive. It is not hard to see that a is related to b if and only if a and b have the same remainder when divided by 4. It follows that this relation splits the set of integers into four different classes. When we care only what remainder an integer leaves when it is divided by 4, we need only know which class it is in, not its particular value.

These two relations, R and congruence modulo 4, are examples of equivalence relations, namely, relations that are reflexive, symmetric, and transitive. In this section we will show that such relations split sets into disjoint classes of equivalent elements. Equivalence relations arise whenever we care only whether an element of a set is in a certain class of elements, instead of caring about its particular identity.

Equivalence Relations

Links

In this section we will study relations with a particular combination of properties that allows them to be used to relate objects that are similar in some way.

DEFINITION 1 A relation on a set A is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

possible when an integer is divided by m. These m congruence classes are denoted by $[0]_m, [1]_m, \ldots, [m-1]_m$. They form a partition of the set of integers.

EXAMPLE 14 What are the sets in the partition of the integers arising from congruence modulo 4?

Solution: There are four congruence classes, corresponding to $[0]_4$, $[1]_4$, $[2]_4$, and $[3]_4$. They are the sets

$$[0]_4 = \{\ldots, -8, -4, 0, 4, 8, \ldots\},$$
$$[1]_4 = \{\ldots, -7, -3, 1, 5, 9, \ldots\},$$
$$[2]_4 = \{\ldots, -6, -2, 2, 6, 10, \ldots\},$$
$$[3]_4 = \{\ldots, -5, -1, 3, 7, 11, \ldots\}.$$

These congruence classes are disjoint, and every integer is in exactly one of them. In other words, as Theorem 2 says, these congruence classes form a partition. ◀

We now provide an example of a partition of the set of all strings arising from an equivalence relation on this set.

EXAMPLE 15 Let R_3 be the relation from Example 5. What are the sets in the partition of the set of all bit strings arising from the relation R_3 on the set of all bit strings? (Recall that $s \, R_3 \, t$, where s and t are bit strings, if $s = t$ or s and t are bit strings with at least three bits that agree in their first three bits.)

Solution: Note that every bit string of length less than three is equivalent only to itself. Hence $[\lambda]_{R_3} = \{\lambda\}$, $[0]_{R_3} = \{0\}$, $[1]_{R_3} = \{1\}$, $[00]_{R_3} = \{00\}$, $[01]_{R_3} = \{01\}$, $[10]_{R_3} = \{10\}$, and $[11]_{R_3} = \{11\}$. Note that every bit string of length three or more is equivalent to one of the eight bit strings 000, 001, 010, 011, 100, 101, 110, and 111. We have

$$[000]_{R_3} = \{000, 0000, 0001, 00000, 00001, 00010, 00011, \ldots\},$$
$$[001]_{R_3} = \{001, 0010, 0011, 00100, 00101, 00110, 00111, \ldots\},$$
$$[010]_{R_3} = \{010, 0100, 0101, 01000, 01001, 01010, 01011, \ldots\},$$
$$[011]_{R_3} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \ldots\},$$
$$[100]_{R_3} = \{100, 1000, 1001, 10000, 10001, 10010, 10011, \ldots\},$$
$$[101]_{R_3} = \{101, 1010, 1011, 10100, 10101, 10110, 10111, \ldots\},$$
$$[110]_{R_3} = \{110, 1100, 1101, 11000, 11001, 11010, 11011, \ldots\}, \text{ and}$$
$$[111]_{R_3} = \{111, 1110, 1111, 11100, 11101, 11110, 11111, \ldots\}.$$

These 15 equivalence classes are disjoint and every bit string is in exactly one of them. As Theorem 2 tells us, these equivalence classes partition the set of all bit strings. ◀

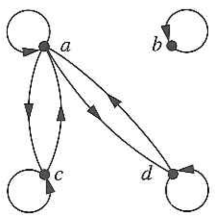
Exercises

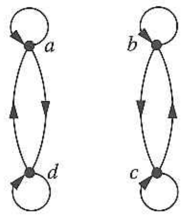
1. Which of these relations on $\{0, 1, 2, 3\}$ are equivalence relations? Determine the properties of an equivalence relation that the others lack.
   a) $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
   b) $\{(0, 0), (0, 2), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
   c) $\{(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3)\}$

d) $\{(0,0), (1,1), (1,3), (2,2), (2,3), (3,1), (3,2), (3,3)\}$
e) $\{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,2), (3,3)\}$

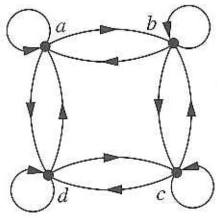
2. Which of these relations on the set of all people are equivalence relations? Determine the properties of an equivalence relation that the others lack.
   a) $\{(a, b) \mid a$ and b are the same age$\}$
   b) $\{(a, b) \mid a$ and b have the same parents$\}$
   c) $\{(a, b) \mid a$ and b share a common parent$\}$
   d) $\{(a, b) \mid a$ and b have met$\}$
   e) $\{(a, b) \mid a$ and b speak a common language$\}$
3. Which of these relations on the set of all functions from $\mathbf{Z}$ to $\mathbf{Z}$ are equivalence relations? Determine the properties of an equivalence relation that the others lack.
   a) $\{(f, g) \mid f(1) = g(1)\}$
   b) $\{(f, g) \mid f(0) = g(0)$ or $f(1) = g(1)\}$
   c) $\{(f, g) \mid f(x) - g(x) = 1$ for all $x \in \mathbf{Z}\}$
   d) $\{(f, g) \mid$ for some $C \in \mathbf{Z}$, for all $x \in \mathbf{Z}$, $f(x) - g(x) = C\}$
   e) $\{(f, g) \mid f(0) = g(1)$ and $f(1) = g(0)\}$
4. Define three equivalence relations on the set of students in your discrete mathematics class different from the relations discussed in the text. Determine the equivalence classes for each of these equivalence relations.
5. Define three equivalence relations on the set of buildings on a college campus. Determine the equivalence classes for each of these equivalence relations.
6. Define three equivalence relations on the set of classes offered at your school. Determine the equivalence classes for each of these equivalence relations.
7. Show that the relation of logical equivalence on the set of all compound propositions is an equivalence relation. What are the equivalence classes of **F** and of **T**?
8. Let R be the relation on the set of all sets of real numbers such that $S\,R\,T$ if and only if S and T have the same cardinality. Show that R is an equivalence relation. What are the equivalence classes of the sets $\{0, 1, 2\}$ and $\mathbf{Z}$?
9. Suppose that A is a nonempty set, and f is a function that has A as its domain. Let R be the relation on A consisting of all ordered pairs (x, y) such that $f(x) = f(y)$.
   a) Show that R is an equivalence relation on A.
   b) What are the equivalence classes of R?
10. Suppose that A is a nonempty set and R is an equivalence relation on A. Show that there is a function f with A as its domain such that $(x, y) \in R$ if and only if $f(x) = f(y)$.
11. Show that the relation R consisting of all pairs (x, y) such that x and y are bit strings of length three or more that agree in their first three bits is an equivalence relation on the set of all bit strings of length three or more.
12. Show that the relation R consisting of all pairs (x, y) such that x and y are bit strings of length three or more that agree except perhaps in their first three bits is an equivalence relation on the set of all bit strings of length three or more.
13. Show that the relation R consisting of all pairs (x, y) such that x and y are bit strings that agree in their first and third bits is an equivalence relation on the set of all bit strings of length three or more.
14. Let R be the relation consisting of all pairs (x, y) such that x and y are strings of uppercase and lowercase English letters with the property that for every positive integer n, the nth characters in x and y are the same letter, either uppercase or lowercase. Show that R is an equivalence relation.
15. Let R be the relation on the set of ordered pairs of positive integers such that $((a, b), (c, d)) \in R$ if and only if $a + d = b + c$. Show that R is an equivalence relation.
16. Let R be the relation on the set of ordered pairs of positive integers such that $((a, b), (c, d)) \in R$ if and only if $ad = bc$. Show that R is an equivalence relation.
17. (*Requires calculus*)
    a) Show that the relation R on the set of all differentiable functions from $\mathbf{R}$ to $\mathbf{R}$ consisting of all pairs (f, g) such that $f'(x) = g'(x)$ for all real numbers x is an equivalence relation.
    b) Which functions are in the same equivalence class as the function $f(x) = x^2$?
18. (*Requires calculus*)
    a) Let n be a positive integer. Show that the relation R on the set of all polynomials with real-valued coefficients consisting of all pairs (f, g) such that $f^{(n)}(x) = g^{(n)}(x)$ is an equivalence relation. [Here $f^{(n)}(x)$ is the nth derivative of $f(x)$.]
    b) Which functions are in the same equivalence class as the function $f(x) = x^4$, where $n = 3$?
19. Let R be the relation on the set of all URLs (or Web addresses) such that $x\,R\,y$ if and only if the Web page at x is the same as the Web page at y. Show that R is an equivalence relation.
20. Let R be the relation on the set of all people who have visited a particular Web page such that $x\,R\,y$ if and only if person x and person y have followed the same set of links starting at this Web page (going from Web page to Web page until they stop using the Web). Show that R is an equivalence relation.

In Exercises 21–23 determine whether the relation with the directed graphs shown is an equivalence relation.

21.


22.


23.



24. Determine whether the relations represented by these zero–one matrices are equivalence relations.

a) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

25. Show that the relation R on the set of all bit strings such that $s\,R\,t$ if and only if s and t contain the same number of 1s is an equivalence relation.

26. What are the equivalence classes of the equivalence relations in Exercise 1?

27. What are the equivalence classes of the equivalence relations in Exercise 2?

28. What are the equivalence classes of the equivalence relations in Exercise 3?

29. What is the equivalence class of the bit string 011 for the equivalence relation in Exercise 25?

30. What are the equivalence classes of these bit strings for the equivalence relation in Exercise 11?

a) 010 **b)** 1011 **c)** 11111 **d)** 01010101

31. What are the equivalence classes of the bit strings in Exercise 30 for the equivalence relation from Exercise 11?

32. What are the equivalence classes of the bit strings in Exercise 30 for the equivalence relation from Exercise 12?

33. What are the equivalence classes of the bit strings in Exercise 30 for the equivalence relation R_4 from Example 5 on the set of all bit strings? (Recall that bit strings s and t are equivalent under R_4 if and only if they are equal or they are both at least four bits long and agree in their first four bits.)

34. What are the equivalence classes of the bit strings in Exercise 30 for the equivalence relation R_5 from Example 5 on the set of all bit strings? (Recall that bit strings s and t are equivalent under R_5 if and only if they are equal or they are both at least five bits long and agree in their first five bits.)

35. What is the congruence class $[n]_5$ (that is, the equivalence class of n with respect to congruence modulo 5) when n is

a) 2? **b)** 3? **c)** 6? **d)** -3?

36. What is the congruence class $[4]_m$ when m is

a) 2? **b)** 3? **c)** 6? **d)** 8?

37. Give a description of each of the congruence classes modulo 6.

38. What is the equivalence class of the strings with respect to the equivalence relation in Exercise 14?

a) *No* **b)** *Yes* **c)** *Help*

39. a) What is the equivalence class of $(1, 2)$ with respect to the equivalence relation in Exercise 15?

b) Give an interpretation of the equivalence classes for the equivalence relation R in Exercise 15. [*Hint:* Look at the difference $a - b$ corresponding to (a, b).]

40. a) What is the equivalence class of $(1, 2)$ with respect to the equivalence relation in Exercise 16?

b) Give an interpretation of the equivalence classes for the equivalence relation R in Exercise 16. [*Hint:* Look at the ratio a/b corresponding to (a, b).]

41. Which of these collections of subsets are partitions of $\{1, 2, 3, 4, 5, 6\}$?

a) $\{1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}$ **b)** $\{1\}, \{2, 3, 6\}, \{4\}, \{5\}$

c) $\{2, 4, 6\}, \{1, 3, 5\}$ **d)** $\{1, 4, 5\}, \{2, 6\}$

42. Which of these collections of subsets are partitions of $\{-3, -2, -1, 0, 1, 2, 3\}$?

a) $\{-3, -1, 1, 3\}, \{-2, 0, 2\}$

b) $\{-3, -2, -1, 0\}, \{0, 1, 2, 3\}$

c) $\{-3, 3\}, \{-2, 2\}, \{-1, 1\}, \{0\}$

d) $\{-3, -2, 2, 3\}, \{-1, 1\}$

43. Which of these collections of subsets are partitions on the set of bit strings of length 8?

a) the set of bit strings that begin with 1, the set of bit strings that begin with 00, and the set of bit strings that begin with 01

b) the set of bit strings that contain the string 00, the set of bit strings that contain the string 01, the set of bit strings that contain the string 10, and the set of bit strings that contain the string 11

c) the set of bit strings that end with 00, the set of bit strings that end with 01, the set of bit strings that end with 10, and the set of bit strings that end with 11

d) the set of bit strings that end with 111, the set of bit strings that end with 011, and the set of bit strings that end with 00

e) the set of bit strings that have $3k$ ones, where k is a nonnegative integer; the set of bit strings that contain $3k + 1$ ones, where k is a nonnegative integer; and the set of bit strings that contain $3k + 2$ ones, where k is a nonnegative integer

44. Which of these collections of subsets are partitions of the set of integers?

a) the set of even integers and the set of odd integers

b) the set of positive integers and the set of negative integers

c) the set of integers divisible by 3, the set of integers leaving a remainder of 1 when divided by 3, and the set of integers leaving a remainder of 2 when divided by 3

d) the set of integers less than -100, the set of integers with absolute value not exceeding 100, and the set of integers greater than 100

e) the set of integers not divisible by 3, the set of even integers, and the set of integers that leave a remainder of 3 when divided by 6

45. Which of these are partitions of the set $\mathbf{Z} \times \mathbf{Z}$ of ordered pairs of integers?

a) the set of pairs (x, y), where x or y is odd; the set of pairs (x, y), where x is even; and the set of pairs (x, y), where y is even

b) the set of pairs (x, y), where both x and y are odd; the set of pairs (x, y), where exactly one of x and y is odd; and the set of pairs (x, y), where both x and y are even

c) the set of pairs (x, y), where x is positive; the set of pairs (x, y), where y is positive; and the set of pairs (x, y), where both x and y are negative

d) the set of pairs (x, y), where $3 \mid x$ and $3 \mid y$; the set of pairs (x, y), where $3 \mid x$ and $3 \nmid y$; the set of pairs (x, y), where $3 \nmid x$ and $3 \mid y$; and the set of pairs (x, y), where $3 \nmid x$ and $3 \nmid y$

e) the set of pairs (x, y), where $x > 0$ and $y > 0$; the set of pairs (x, y), where $x > 0$ and $y \leq 0$; the set of pairs (x, y), where $x \leq 0$ and $y > 0$; and the set of pairs (x, y), where $x \leq 0$ and $y \leq 0$

f) the set of pairs (x, y), where $x \neq 0$ and $y \neq 0$; the set of pairs (x, y), where $x = 0$ and $y \neq 0$; and the set of pairs (x, y), where $x \neq 0$ and $y = 0$

46. Which of these are partitions of the set of real numbers?

a) the negative real numbers, $\{0\}$, the positive real numbers

b) the set of irrational numbers, the set of rational numbers

c) the set of intervals $[k, k+1]$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

d) the set of intervals $(k, k+1)$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

e) the set of intervals $(k, k+1]$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

f) the sets $\{x + n \mid n \in \mathbf{Z}\}$ for all $x \in [0, 1)$

47. List the ordered pairs in the equivalence relations produced by these partitions of $\{0, 1, 2, 3, 4, 5\}$.

a) $\{0\}, \{1, 2\}, \{3, 4, 5\}$

b) $\{0, 1\}, \{2, 3\}, \{4, 5\}$

c) $\{0, 1, 2\}, \{3, 4, 5\}$

d) $\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

48. List the ordered pairs in the equivalence relations produced by these partitions of $\{a, b, c, d, e, f, g\}$.

a) $\{a, b\}, \{c, d\}, \{e, f, g\}$

b) $\{a\}, \{b\}, \{c, d\}, \{e, f\}, \{g\}$

c) $\{a, b, c, d\}, \{e, f, g\}$

d) $\{a, c, e, g\}, \{b, d\}, \{f\}$

A partition P_1 is called a **refinement** of the partition P_2 if every set in P_1 is a subset of one of the sets in P_2.

49. Show that the partition formed from congruence classes modulo 6 is a refinement of the partition formed from congruence classes modulo 3.

50. Show that the partition of the set of people living in the United States consisting of subsets of people living in the same county (or parish) and same state is a refinement of the partition consisting of subsets of people living in the same state.

51. Show that the partition of the set of bit strings of length 16 formed by equivalence classes of bit strings that agree on the last eight bits is a refinement of the partition formed from the equivalence classes of bit strings that agree on the last four bits.

In Exercises 52 and 53, R_n refers to the family of equivalence relations defined in Example 5. Recall that $s\ R_n\ t$, where s and t are two strings if $s = t$ or s and t are strings with at least n characters that agree in their first n characters.

52. Show that the partition of the set of all bit strings formed by equivalence classes of bit strings with respect to the equivalence relation R_4 is a refinement of the partition formed by equivalence classes of bit strings with respect to the equivalence relation R_3.

53. Show that the partition of the set of all identifiers in C formed by the equivalence classes of identifiers with respect to the equivalence relation R_{31} is a refinement of the partition formed by equivalence classes of identifiers with respect to the equivalence relation R_8. (Compilers for "old" C consider identifiers the same when their names agree in their first eight characters, while compilers in standard C consider identifiers the same when their names agree in their first 31 characters.)

54. Suppose that R_1 and R_2 are equivalence relations on a set A. Let P_1 and P_2 be the partitions that correspond to R_1 and R_2, respectively. Show that $R_1 \subseteq R_2$ if and only if P_1 is a refinement of P_2.

55. Find the smallest equivalence relation on the set $\{a, b, c, d, e\}$ containing the relation $\{(a, b), (a, c), (d, e)\}$.

56. Suppose that R_1 and R_2 are equivalence relations on the set S. Determine whether each of these combinations of R_1 and R_2 must be an equivalence relation.

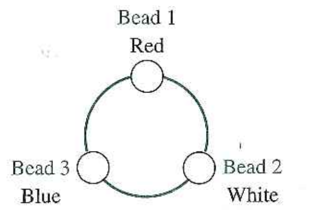
a) $R_1 \cup R_2$ **b)** $R_1 \cap R_2$ **c)** $R_1 \oplus R_2$

57. Consider the equivalence relation from Example 3, namely, $R = \{(x, y) \mid x - y \text{ is an integer}\}$.

a) What is the equivalence class of 1 for this equivalence relation?

b) What is the equivalence class of 1/2 for this equivalence relation?

***58.** Each bead on a bracelet with three beads is either red, white, or blue, as illustrated in the figure shown.



Define the relation R between bracelets as: (B_1, B_2), where B_1 and B_2 are bracelets, belongs to R if and only if B_2 can be obtained from B_1 by rotating it or rotating it and then reflecting it.

a) Show that R is an equivalence relation.

b) What are the equivalence classes of R?

*59. Let R be the relation on the set of all colorings of the 2×2 checkerboard where each of the four squares is colored either red or blue so that (C_1, C_2), where C_1 and C_2 are 2×2 checkerboards with each of their four squares colored blue or red, belongs to R if and only if C_2 can be obtained from C_1 either by rotating the checkerboard or by rotating it and then reflecting it.

a) Show that R is an equivalence relation.

b) What are the equivalence classes of R?

60. a) Let R be the relation on the set of functions from $\mathbf{Z}^+$ to $\mathbf{Z}^+$ such that (f, g) belongs to R if and only if f is $\Theta(g)$ (see Section 3.2). Show that R is an equivalence relation.

b) Describe the equivalence class containing $f(n) = n^2$ for the equivalence relation of part (a).

61. Determine the number of different equivalence relations on a set with three elements by listing them.

62. Determine the number of different equivalence relations on a set with four elements by listing them.

*63. Do we necessarily get an equivalence relation when we form the transitive closure of the symmetric closure of the reflexive closure of a relation?

*64. Do we necessarily get an equivalence relation when we form the symmetric closure of the reflexive closure of the transitive closure of a relation?

65. Suppose we use Theorem 2 to form a partition P from an equivalence relation R. What is the equivalence relation R' that results if we use Theorem 2 again to form an equivalence relation from P?

66. Suppose we use Theorem 2 to form an equivalence relation R from a partition P. What is the partition P' that results if we use Theorem 2 again to form a partition from R?

67. Devise an algorithm to find the smallest equivalence relation containing a given relation.

*68. Let $p(n)$ denote the number of different equivalence relations on a set with n elements (and by Theorem 2 the number of partitions of a set with n elements). Show that $p(n)$ satisfies the recurrence relation $p(n) = \sum_{j=0}^{n-1} C(n-1, j)p(n-j-1)$ and the initial condition $p(0) = 1$. (*Note:* The numbers $p(n)$ are called **Bell numbers** after the American mathematician E. T. Bell.)

69. Use Exercise 68 to find the number of different equivalence relations on a set with n elements, where n is a positive integer not exceeding 10.

8.6 Partial Orderings

Introduction

We often use relations to order some or all of the elements of sets. For instance, we order words using the relation containing pairs of words (x, y), where x comes before y in the dictionary. We schedule projects using the relation consisting of pairs (x, y), where x and y are tasks in a project such that x must be completed before y begins. We order the set of integers using the relation containing the pairs (x, y), where x is less than y. When we add all of the pairs of the form (x, x) to these relations, we obtain a relation that is reflexive, antisymmetric, and transitive. These are properties that characterize relations used to order the elements of sets.

Links

DEFINITION 1 A relation R on a set S is called a *partial ordering* or *partial order* if it is reflexive, antisymmetric, and transitive. A set S together with a partial ordering R is called a *partially ordered set,* or *poset,* and is denoted by (S, R). Members of S are called *elements* of the poset.

We give examples of posets in Examples 1–3.

EXAMPLE 1 Show that the "greater than or equal" relation ($\geq$) is a partial ordering on the set of integers.

Extra Examples

Solution: Because $a \geq a$ for every integer a, $\geq$ is reflexive. If $a \geq b$ and $b \geq a$, then $a = b$. Hence, $\geq$ is antisymmetric. Finally, $\geq$ is transitive because $a \geq b$ and $b \geq c$ imply that $a \geq c$. It follows that $\geq$ is a partial ordering on the set of integers and $(\mathbf{Z}, \geq)$ is a poset. ◀

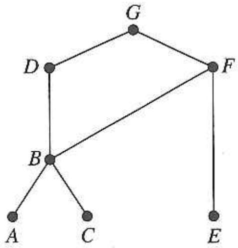


FIGURE 10 The Hasse Diagram for Seven Tasks.

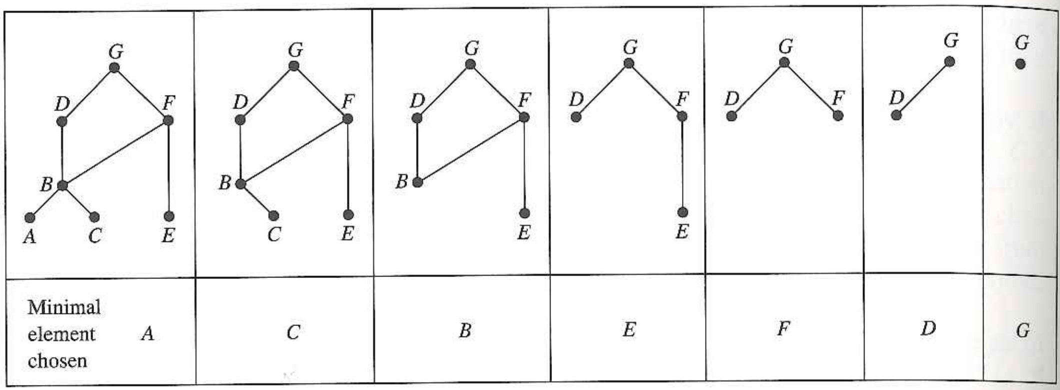


FIGURE 11 A Topological Sort of the Tasks.

EXAMPLE 27 A development project at a computer company requires the completion of seven tasks. Some of these tasks can be started only after other tasks are finished. A partial ordering on tasks is set up by considering task $X \prec$ task Y if task Y cannot be started until task X has been completed. The Hasse diagram for the seven tasks, with respect to this partial ordering, is shown in Figure 10. Find an order in which these tasks can be carried out to complete the project.

Solution: An ordering of the seven tasks can be obtained by performing a topological sort. The steps of a sort are illustrated in Figure 11. The result of this sort, $A \prec C \prec B \prec E \prec F \prec D \prec G$, gives one possible order for the tasks. ◀

Exercises

1. Which of these relations on $\{0, 1, 2, 3\}$ are partial orderings? Determine the properties of a partial ordering that the others lack.

a) $\{(0,0), (1,1), (2,2), (3,3)\}$
b) $\{(0,0), (1,1), (2,0), (2,2), (2,3), (3,2), (3,3)\}$
c) $\{(0,0), (1,1), (1,2), (2,2), (3,3)\}$
d) $\{(0,0), (1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\}$
e) $\{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,2), (3,3)\}$

2. Which of these relations on $\{0, 1, 2, 3\}$ are partial orderings? Determine the properties of a partial ordering that the others lack.

a) $\{(0,0), (2,2), (3,3)\}$
b) $\{(0,0), (1,1), (2,0), (2,2), (2,3), (3,3)\}$
c) $\{(0,0), (1,1), (1,2), (2,2), (3,1), (3,3)\}$
d) $\{(0,0), (1,1), (1,2), (1,3), (2,0), (2,2), (2,3), (3,0), (3,3)\}$
e) $\{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (1,3), (2,0), (2,2), (3,3)\}$

3. Is (S, R) a poset if S is the set of all people in the world and $(a, b) \in R$, where a and b are people, if

a) a is taller than b?
b) a is not taller than b?
c) $a = b$ or a is an ancestor of b?
d) a and b have a common friend?

4. Is (S, R) a poset if S is the set of all people in the world and $(a, b) \in R$, where a and b are people, if

a) a is no shorter than b?
b) a weighs more than b?
c) $a = b$ or a is a descendant of b?
d) a and b do not have a common friend?

5. Which of these are posets?

a) $(\mathbf{Z}, =)$ **b)** $(\mathbf{Z}, \neq)$ **c)** $(\mathbf{Z}, \geq)$ **d)** $(\mathbf{Z}, \nmid)$

6. Which of these are posets?

a) $(\mathbf{R}, =)$ **b)** $(\mathbf{R}, <)$ **c)** $(\mathbf{R}, \leq)$ **d)** $(\mathbf{R}, \neq)$

7. Determine whether the relations represented by these zero–one matrices are partial orders.

a) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

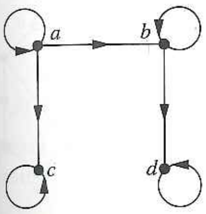
8. Determine whether the relations represented by these zero–one matrices are partial orders.

a) $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

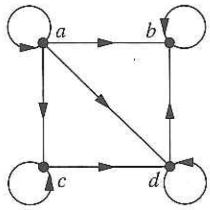
c) $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

In Exercises 9–11 determine whether the relation with the directed graph shown is a partial order.

9.



10.



11.

12. Let (S, R) be a poset. Show that (S, R^{-1}) is also a poset, where R^{-1} is the inverse of R. The poset (S, R^{-1}) is called the **dual** of (S, R).

13. Find the duals of these posets.

a) $(\{0, 1, 2\}, \leq)$ b) $(\mathbf{Z}, \geq)$
c) $(P(\mathbf{Z}), \supseteq)$ d) $(\mathbf{Z}^+, |)$

14. Which of these pairs of elements are comparable in the poset $(\mathbf{Z}^+, |)$?

a) 5, 15 b) 6, 9 c) 8, 16 d) 7, 7

15. Find two incomparable elements in these posets.

a) $(P(\{0, 1, 2\}), \subseteq)$ b) $(\{1, 2, 4, 6, 8\}, |)$

16. Let $S = \{1, 2, 3, 4\}$. With respect to the lexicographic order based on the usual "less than" relation,

a) find all pairs in $S \times S$ less than $(2, 3)$.
b) find all pairs in $S \times S$ greater than $(3, 1)$.
c) draw the Hasse diagram of the poset $(S \times S, \preccurlyeq)$.

17. Find the lexicographic ordering of these n-tuples:

a) $(1, 1, 2), (1, 2, 1)$ b) $(0, 1, 2, 3), (0, 1, 3, 2)$
c) $(1, 0, 1, 0, 1), (0, 1, 1, 1, 0)$

18. Find the lexicographic ordering of these strings of lowercase English letters:

a) *quack, quick, quicksilver, quicksand, quacking*
b) *open, opener, opera, operand, opened*
c) *zoo, zero, zoom, zoology, zoological*

19. Find the lexicographic ordering of the bit strings 0, 01, 11, 001, 010, 011, 0001, and 0101 based on the ordering $0 < 1$.

20. Draw the Hasse diagram for the "greater than or equal to" relation on $\{0, 1, 2, 3, 4, 5\}$.

21. Draw the Hasse diagram for the "less than or equal to" relation on $\{0, 2, 5, 10, 11, 15\}$.

22. Draw the Hasse diagram for divisibility on the set

a) $\{1, 2, 3, 4, 5, 6\}$. b) $\{3, 5, 7, 11, 13, 16, 17\}$.
c) $\{2, 3, 5, 10, 11, 15, 25\}$. d) $\{1, 3, 9, 27, 81, 243\}$.

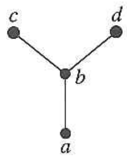
23. Draw the Hasse diagram for divisibility on the set

a) $\{1, 2, 3, 4, 5, 6, 7, 8\}$. b) $\{1, 2, 3, 5, 7, 11, 13\}$.
c) $\{1, 2, 3, 6, 12, 24, 36, 48\}$.
d) $\{1, 2, 4, 8, 16, 32, 64\}$.

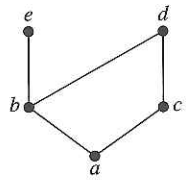
24. Draw the Hasse diagram for inclusion on the set $P(S)$, where $S = \{a, b, c, d\}$.

In Exercises 25–27 list all ordered pairs in the partial ordering with the accompanying Hasse diagram.

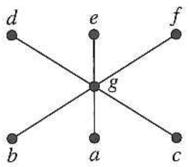
25.



26.



27.



Let $(S, \preccurlyeq)$ be a poset. We say that an element $y \in S$ **covers** an element $x \in S$ if $x \prec y$ and there is no element $z \in S$ such that $x \prec z \prec y$. The set of pairs (x, y) such that y covers x is called the **covering relation** of $(S, \preccurlyeq)$.

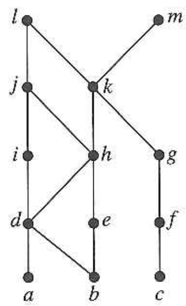
28. What is the covering relation of the partial ordering $\{(a, b) \mid a \text{ divides } b\}$ on $\{1, 2, 3, 4, 6, 12\}$?

29. What is the covering relation of the partial ordering $\{(A, B) \mid A \subseteq B\}$ on the power set of S, where $S = \{a, b, c\}$?

30. Show that the pair (x, y) belongs to the covering relation of the finite poset $(S, \preccurlyeq)$ if and only if x is lower than y and there is an edge joining x and y in the Hasse diagram of this poset.

31. Show that a finite poset can be reconstructed from its covering relation. [*Hint:* Show that the poset is the reflexive transitive closure of its covering relation.]

32. Answer these questions for the partial order represented by this Hasse diagram.



a) Find the maximal elements.
b) Find the minimal elements.
c) Is there a greatest element?
d) Is there a least element?
e) Find all upper bounds of $\{a, b, c\}$.
f) Find the least upper bound of $\{a, b, c\}$, if it exists.
g) Find all lower bounds of $\{f, g, h\}$.
h) Find the greatest lower bound of $\{f, g, h\}$, if it exists.

33. Answer these questions for the poset $(\{3, 5, 9, 15, 24, 45\}, |)$.
a) Find the maximal elements.
b) Find the minimal elements.
c) Is there a greatest element?
d) Is there a least element?
e) Find all upper bounds of $\{3, 5\}$.
f) Find the least upper bound of $\{3, 5\}$, if it exists.
g) Find all lower bounds of $\{15, 45\}$.
h) Find the greatest lower bound of $\{15, 45\}$, if it exists.

34. Answer these questions for the poset $(\{2, 4, 6, 9, 12, 18, 27, 36, 48, 60, 72\}, |)$.
a) Find the maximal elements.
b) Find the minimal elements.
c) Is there a greatest element?
d) Is there a least element?
e) Find all upper bounds of $\{2, 9\}$.
f) Find the least upper bound of $\{2, 9\}$, if it exists.
g) Find all lower bounds of $\{60, 72\}$.
h) Find the greatest lower bound of $\{60, 72\}$, if it exists.

35. Answer these questions for the poset $(\{\{1\}, \{2\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}, \subseteq)$.
a) Find the maximal elements.
b) Find the minimal elements.
c) Is there a greatest element?
d) Is there a least element?
e) Find all upper bounds of $\{\{2\}, \{4\}\}$.
f) Find the least upper bound of $\{\{2\}, \{4\}\}$, if it exists.
g) Find all lower bounds of $\{\{1, 3, 4\}, \{2, 3, 4\}\}$.
h) Find the greatest lower bound of $\{\{1, 3, 4\}, \{2, 3, 4\}\}$, if it exists.

36. Give a poset that has
a) a minimal element but no maximal element.
b) a maximal element but no minimal element.
c) neither a maximal nor a minimal element.

37. Show that lexicographic order is a partial ordering on the Cartesian product of two posets.

38. Show that lexicographic order is a partial ordering on the set of strings from a poset.

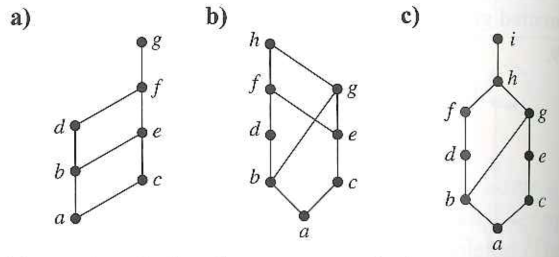
39. Suppose that $(S, \preccurlyeq_1)$ and $(T, \preccurlyeq_2)$ are posets. Show that $(S \times T, \preccurlyeq)$ is a poset where $(s, t) \preccurlyeq (u, v)$ if and only if $s \preccurlyeq_1 u$ and $t \preccurlyeq_2 v$.

40. a) Show that there is exactly one greatest element of a poset, if such an element exists.
b) Show that there is exactly one least element of a poset, if such an element exists.

41. a) Show that there is exactly one maximal element in a poset with a greatest element.
b) Show that there is exactly one minimal element in a poset with a least element.

42. a) Show that the least upper bound of a set in a poset is unique if it exists.
b) Show that the greatest lower bound of a set in a poset is unique if it exists.

43. Determine whether the posets with these Hasse diagrams are lattices.



44. Determine whether these posets are lattices.
a) $(\{1, 3, 6, 9, 12\}, |)$ b) $(\{1, 5, 25, 125\}, |)$
c) $(\mathbf{Z}, \geq)$
d) $(P(S), \supseteq)$, where $P(S)$ is the power set of a set S

45. Show that every nonempty finite subset of a lattice has a least upper bound and a greatest lower bound.

46. Show that if the poset (S, R) is a lattice then the dual poset (S, R^{-1}) is also a lattice.

47. In a company, the lattice model of information flow is used to control sensitive information with security classes represented by ordered pairs (A, C). Here A is an authority level, which may be nonproprietary (0), proprietary (1), restricted (2), or registered (3). A category C is a subset of the set of all projects {*Cheetah, Impala, Puma*}. (Names of animals are often used as code names for projects in companies.)
a) Is information permitted to flow from (*Proprietary*, {*Cheetah, Puma*}) into (*Restricted*, {*Puma*})?
b) Is information permitted to flow from (*Restricted*, {*Cheetah*}) into (*Registered*, {*Cheetah, Impala*})?
c) Into which classes is information from (*Proprietary*, {*Cheetah, Puma*}) permitted to flow?
d) From which classes is information permitted to flow into the security class (*Restricted*, {*Impala, Puma*})?

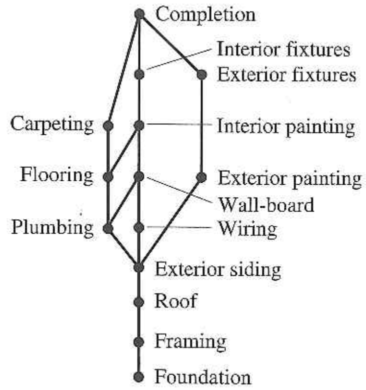
48. Show that the set S of security classes (A, C) is a lattice, where A is a positive integer representing an authority class and C is a subset of a finite set of compartments, with $(A_1, C_1) \preccurlyeq (A_2, C_2)$ if and only if $A_1 \leq A_2$ and $C_1 \subseteq C_2$. [*Hint*: First show that $(S, \preccurlyeq)$ is a poset and then show that the least upper bound and greatest lower bound of (A_1, C_1) and (A_2, C_2) are $(\max(A_1, A_2), C_1 \cup C_2)$ and $(\min(A_1, A_2), C_1 \cap C_2)$, respectively.]

***49.** Show that the set of all partitions of a set S with the relation $P_1 \preccurlyeq P_2$ if the partition P_1 is a refinement of the partition P_2 is a lattice. (See the preamble to Exercise 49 of Section 8.5.)

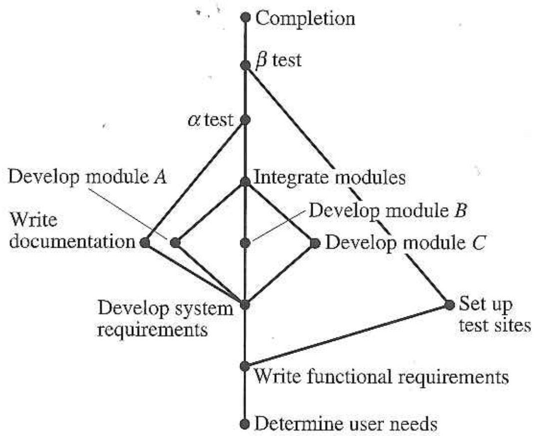
50. Show that every totally ordered set is a lattice.
51. Show that every finite lattice has a least element and a greatest element.
52. Give an example of an infinite lattice with
    a) neither a least nor a greatest element.
    b) a least but not a greatest element.
    c) a greatest but not a least element.
    d) both a least and a greatest element.
53. Verify that $(\mathbf{Z}^+ \times \mathbf{Z}^+, \preccurlyeq)$ is a well-ordered set, where $\preccurlyeq$ is lexicographic order, as claimed in Example 8.
54. Determine whether each of these posets is well-ordered.
    a) $(S, \leq)$, where $S = \{10, 11, 12, \ldots\}$
    b) $(\mathbf{Q} \cap [0, 1], \leq)$ (the set of rational numbers between 0 and 1 inclusive)
    c) $(S, \leq)$, where S is the set of positive rational numbers with denominators not exceeding 3
    d) $(\mathbf{Z}^-, \geq)$, where $\mathbf{Z}^-$ is the set of negative integers

A poset $(R, \preccurlyeq)$ is **well-founded** if there is no infinite decreasing sequence of elements in the poset, that is, elements $x_1, x_2, \ldots, x_n$ such that $\cdots \prec x_n \prec \cdots \prec x_2 \prec x_1$. A poset $(R, \preccurlyeq)$ is **dense** if for all $x \in S$ and $y \in S$ with $x \prec y$, there is an element $z \in R$ such that $x \prec z \prec y$.

55. Show that the poset $(\mathbf{Z}, \preccurlyeq)$, where $x \prec y$ if and only if $|x| < |y|$ is well-founded but is not a totally ordered set.
56. Show that a dense poset with at least two elements that are comparable is not well-founded.
57. Show that the poset of rational numbers with the usual "less than or equal to" relation, $(\mathbf{Q}, \leq)$, is a dense poset.
*58. Show that the set of strings of lowercase English letters with lexicographic order is neither well-founded nor dense.
59. Show that a poset is well-ordered if and only if it is totally ordered and well-founded.
60. Show that a finite nonempty poset has a maximal element.
61. Find a compatible total order for the poset with the Hasse diagram shown in Exercise 32.
62. Find a compatible total order for the divisibility relation on the set $\{1, 2, 3, 6, 8, 12, 24, 36\}$.
63. Find an order different from that constructed in Example 27 for completing the tasks in the development project.
64. Schedule the tasks needed to build a house, by specifying their order, if the Hasse diagram representing these tasks is as shown in the figure.



65. Find an ordering of the tasks of a software project if the Hasse diagram for the tasks of the project is as shown.



Key Terms and Results

TERMS

binary relation from A to B: a subset of $A \times B$

relation on A: a binary relation from A to itself (i.e., a subset of $A \times A$)

$S \circ R$: composite of R and S

R^{-1}: inverse relation of R

R^n: nth power of R

reflexive: a relation R on A is reflexive if $(a, a) \in R$ for all $a \in A$

symmetric: a relation R on A is symmetric if $(b, a) \in R$ whenever $(a, b) \in R$

antisymmetric: a relation R on A is antisymmetric if $a = b$ whenever $(a, b) \in R$ and $(b, a) \in R$

transitive: a relation R on A is transitive if $(a, b) \in R$ and $(b, c) \in R$ implies that $(a, c) \in R$

n-ary relation on $A_1, A_2, \ldots, A_n$: a subset of $A_1 \times A_2 \times \cdots \times A_n$

relational data model: a model for representing databases using n-ary relations

primary key: a domain of an n-ary relation such that an n-tuple is uniquely determined by its value for this domain

4. a) How many reflexive relations are there on a set with n elements?
   b) How many symmetric relations are there on a set with n elements?
   c) How many antisymmetric relations are there on a set with n elements?
5. a) Explain how an n-ary relation can be used to represent information about students at a university.
   b) How can the 5-ary relation containing names of students, their addresses, telephone numbers, majors, and grade point averages be used to form a 3-ary relation containing the names of students, their majors, and their grade point averages?
   c) How can the 4-ary relation containing names of students, their addresses, telephone numbers, and majors and the 4-ary relation containing names of students, their student numbers, majors, and numbers of credit hours be combined into a single n-ary relation?
6. a) Explain how to use a zero–one matrix to represent a relation on a finite set.
   b) Explain how to use the zero–one matrix representing a relation to determine whether the relation is reflexive, symmetric, and/or antisymmetric.
7. a) Explain how to use a directed graph to represent a relation on a finite set.
   b) Explain how to use the directed graph representing a relation to determine whether a relation is reflexive, symmetric, and/or antisymmetric.
8. a) Define the reflexive closure and the symmetric closure of a relation.
   b) How can you construct the reflexive closure of a relation?
   c) How can you construct the symmetric closure of a relation?
   d) Find the reflexive closure and the symmetric closure of the relation $\{(1, 2), (2, 3), (2, 4), (3, 1)\}$ on the set $\{1, 2, 3, 4\}$.
9. a) Define the transitive closure of a relation.
   b) Can the transitive closure of a relation be obtained by including all pairs (a, c) such that (a, b) and (b, c) belong to the relation?
   c) Describe two algorithms for finding the transitive closure of a relation.
   d) Find the transitive closure of the relation $\{(1,1), (1,3), (2,1), (2,3), (2,4), (3,2), (3,4), (4,1)\}$.
10. a) Define an equivalence relation.
    b) Which relations on the set $\{a, b, c, d\}$ are equivalence relations and contain (a, b) and (b, d)?
11. a) Show that congruence modulo m is an equivalence relation whenever m is a positive integer.
    b) Show that the relation $\{(a, b) \mid a \equiv \pm b \pmod 7\}$ is an equivalence relation on the set of integers.
12. a) What are the equivalence classes of an equivalence relation?
    b) What are the equivalence classes of the "congruent modulo 5" relation?
    c) What are the equivalence classes of the equivalence relation in Question 11(b)?
13. Explain the relationship between equivalence relations on a set and partitions of this set.
14. a) Define a partial ordering.
    b) Show that the divisibility relation on the set of positive integers is a partial order.
15. Explain how partial orderings on the sets A_1 and A_2 can be used to define a partial ordering on the set $A_1 \times A_2$.
16. a) Explain how to construct the Hasse diagram of a partial order on a finite set.
    b) Draw the Hasse diagram of the divisibility relation on the set $\{2, 3, 5, 9, 12, 15, 18\}$.
17. a) Define a maximal element of a poset and the greatest element of a poset.
    b) Give an example of a poset that has three maximal elements.
    c) Give an example of a poset with a greatest element.
18. a) Define a lattice.
    b) Give an example of a poset with five elements that is a lattice and an example of a poset with five elements that is not a lattice.
19. a) Show that every finite subset of a lattice has a greatest lower bound and a least upper bound.
    b) Show that every lattice with a finite number of elements has a least element and a greatest element.
20. a) Define a well-ordered set.
    b) Describe an algorithm for producing a well-ordered set from a partially ordered set.
    c) Explain how the algorithm from (b) can be used to order the tasks in a project if each task can be done only after one or more of the other tasks have been completed.

Supplementary Exercises

1. Let S be the set of all strings of English letters. Determine whether these relations are reflexive, irreflexive, symmetric, antisymmetric, and/or transitive.
   a) $R_1 = \{(a, b) \mid a$ and b have no letters in common$\}$
   b) $R_2 = \{(a, b) \mid a$ and b are not the same length$\}$
   c) $R_3 = \{(a, b) \mid a$ is longer than $b\}$
2. Construct a relation on the set $\{a, b, c, d\}$ that is
   a) reflexive, symmetric, but not transitive.
   b) irreflexive, symmetric, and transitive.
   c) irreflexive, antisymmetric, and not transitive.
   d) reflexive, neither symmetric nor antisymmetric, and transitive.
   e) neither reflexive, irreflexive, symmetric, antisymmetric, nor transitive.

***19.** Devise an algorithm, based on the concept of interior vertices, that finds the length of the longest path between two vertices in a directed graph, or determines that there are arbitrarily long paths between these vertices.

20. Which of these are equivalence relations on the set of all people?

a) $\{(x, y) \mid x \text{ and } y \text{ have the same sign of the zodiac}\}$

b) $\{(x, y) \mid x \text{ and } y \text{ were born in the same year}\}$

c) $\{(x, y) \mid x \text{ and } y \text{ have been in the same city}\}$

***21.** How many different equivalence relations with exactly three different equivalence classes are there on a set with five elements?

22. Show that $\{(x, y) \mid x - y \in \mathbf{Q}\}$ is an equivalence relation on the set of real numbers, where $\mathbf{Q}$ denotes the set of rational numbers. What are $[1]$, $[\frac{1}{2}]$, and $[\pi]$?

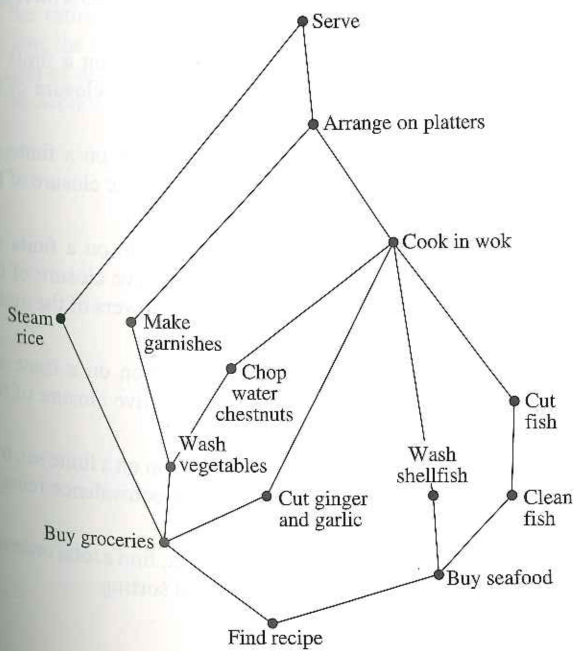
23. Suppose that $P_1 = \{A_1, A_2, \ldots, A_m\}$ and $P_2 = \{B_1, B_2, \ldots, B_n\}$ are both partitions of the set S. Show that the collection of nonempty subsets of the form $A_i \cap B_j$ is a partition of S that is a refinement of both P_1 and P_2 (see the preamble to Exercise 49 of Section 8.5).

***24.** Show that the transitive closure of the symmetric closure of the reflexive closure of a relation R is the smallest equivalence relation that contains R.

25. Let $\mathbf{R}(S)$ be the set of all relations on a set S. Define the relation $\preccurlyeq$ on $\mathbf{R}(S)$ by $R_1 \preccurlyeq R_2$ if $R_1 \subseteq R_2$, where R_1 and R_2 are relations on S. Show that $(\mathbf{R}(S), \preccurlyeq)$ is a poset.

26. Let $\mathbf{P}(S)$ be the set of all partitions of the set S. Define the relation $\preccurlyeq$ on $\mathbf{P}(S)$ by $P_1 \preccurlyeq P_2$ if P_1 is a refinement of P_2 (see Exercise 49 of Section 8.5). Show that $(\mathbf{P}(S), \preccurlyeq)$ is a poset.

27. Schedule the tasks needed to cook a Chinese meal by specifying their order, if the Hasse diagram representing these tasks is as shown here.



A subset of a poset such that every two elements of this subset are comparable is called a **chain.** A subset of a poset is called an **antichain** if every two elements of this subset are incomparable.

28. Find all chains in the posets with the Hasse diagrams shown in Exercises 25–27 in Section 8.6.

29. Find all antichains in the posets with the Hasse diagrams shown in Exercises 25–27 in Section 8.6.

30. Find an antichain with the greatest number of elements in the poset with the Hasse diagram of Exercise 32 in Section 8.6.

31. Show that every maximal chain in a finite poset $(S, \preccurlyeq)$ contains a minimal element of S. (A maximal chain is a chain that is not a subset of a larger chain.)

****32.** Show that every finite poset can be partitioned into k chains, where k is the largest number of elements in an antichain in this poset.

***33.** Show that in any group of $mn + 1$ people there is either a list of $m + 1$ people where a person in the list (except for the first person listed) is a descendant of the previous person on the list, or there are $n + 1$ people such that none of these people is a descendant of any of the other n people. [*Hint:* Use Exercise 32.]

Suppose that $(S, \preccurlyeq)$ is a well-founded partially ordered set. The *principle of well-founded induction* states that $P(x)$ is true for all $x \in S$ if $\forall x(\forall y(y \prec x \rightarrow P(y)) \rightarrow P(x))$.

34. Show that no separate basis case is needed for the principle of well-founded induction. That is, $P(u)$ is true for all minimal elements u in S if $\forall x(\forall y(y \prec x \rightarrow P(y)) \rightarrow P(x))$.

***35.** Show that the principle of well-founded induction is valid.

A relation R on a set A is a **quasi-ordering** on A if R is reflexive and transitive.

36. Let R be the relation on the set of all functions from $\mathbf{Z}^+$ to $\mathbf{Z}^+$ such that (f, g) belongs to R if and only if f is $O(g)$. Show that R is a quasi-ordering.

37. Let R be a quasi-ordering on a set A. Show that $R \cap R^{-1}$ is an equivalence relation.

***38.** Let R be a quasi-ordering and let S be the relation on the set of equivalence classes of $R \cap R^{-1}$ such that (C, D) belongs to S, where C and D are equivalence classes of R, if and only if there are elements c of C and d of D such that (c, d) belongs to R. Show that S is a partial ordering.

Let L be a lattice. Define the **meet** $(\wedge)$ and **join** $(\vee)$ operations by $x \wedge y = \text{glb}(x, y)$ and $x \vee y = \text{lub}(x, y)$.

39. Show that the following properties hold for all elements x, y, and z of a lattice L.

a) $x \wedge y = y \wedge x$ and $x \vee y = y \vee x$ **(commutative laws)**

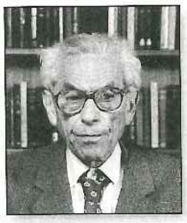
b) $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ and $(x \vee y) \vee z = x \vee (y \vee z)$ **(associative laws)**

c) $x \wedge (x \vee y) = x$ and $x \vee (x \wedge y) = x$ **(absorption laws)**

d) $x \wedge x = x$ and $x \vee x = x$ **(idempotent laws)**

3. Show that the relation R on $\mathbf{Z} \times \mathbf{Z}$ defined by $(a, b)\, R\, (c, d)$ if and only if $a + d = b + c$ is an equivalence relation.
4. Show that a subset of an antisymmetric relation is also antisymmetric.
5. Let R be a reflexive relation on a set A. Show that $R \subseteq R^2$.
6. Suppose that R_1 and R_2 are reflexive relations on a set A. Show that $R_1 \oplus R_2$ is irreflexive.
7. Suppose that R_1 and R_2 are reflexive relations on a set A. Is $R_1 \cap R_2$ also reflexive? Is $R_1 \cup R_2$ also reflexive?
8. Suppose that R is a symmetric relation on a set A. Is $\overline{R}$ also symmetric?
9. Let R_1 and R_2 be symmetric relations. Is $R_1 \cap R_2$ also symmetric? Is $R_1 \cup R_2$ also symmetric?
10. A relation R is called **circular** if aRb and bRc imply that cRa. Show that R is reflexive and circular if and only if it is an equivalence relation.
11. Show that a primary key in an n-ary relation is a primary key in any projection of this relation that contains this key as one of its fields.
12. Is the primary key in an n-ary relation also a primary key in a larger relation obtained by taking the join of this relation with a second relation?
13. Show that the reflexive closure of the symmetric closure of a relation is the same as the symmetric closure of its reflexive closure.
14. Let R be the relation on the set of all mathematicians that contains the ordered pair (a, b) if and only if a and b have written a paper together.
    a) Describe the relation R^2.
    b) Describe the relation R^*.
    c) The **Erdős number** of a mathematician is 1 if this mathematician wrote a paper with the prolific Hungarian mathematician Paul Erdős, it is 2 if this mathematician did not write a joint paper with Erdős but wrote a joint paper with someone who wrote a joint paper with Erdős, and so on (except that the Erdős number of Erdős himself is 0). Give a definition of the Erdős number in terms of paths in R.
15. a) Give an example to show that the transitive closure of the symmetric closure of a relation is not necessarily the same as the symmetric closure of the transitive closure of this relation.
    b) Show, however, that the transitive closure of the symmetric closure of a relation must contain the symmetric closure of the transitive closure of this relation.
16. a) Let S be the set of subroutines of a computer program. Define the relation R by $\mathbf{P}\, R\, \mathbf{Q}$ if subroutine $\mathbf{P}$ calls subroutine $\mathbf{Q}$ during its execution. Describe the transitive closure of R.
    b) For which subroutines $\mathbf{P}$ does $(\mathbf{P}, \mathbf{P})$ belong to the transitive closure of R?
    c) Describe the reflexive closure of the transitive closure of R.
17. Suppose that R and S are relations on a set A with $R \subseteq S$ such that the closures of R and S with respect to a property $\mathbf{P}$ both exist. Show that the closure of R with respect to $\mathbf{P}$ is a subset of the closure of S with respect to $\mathbf{P}$.
18. Show that the symmetric closure of the union of two relations is the union of their symmetric closures.

Links



PAUL ERDŐS (1913–1996) Paul Erdős, born in Budapest, Hungary, was the son of two high school mathematics teachers. He was a child prodigy; at age 3 he could multiply three-digit numbers in his head, and at 4 he discovered negative numbers on his own. Because his mother did not want to expose him to contagious diseases, he was mostly home-schooled. At 17 Erdős entered Eötvös University, graduating four years later with a Ph.D. in mathematics. After graduating he spent four years at Manchester, England, on a postdoctoral fellowship. In 1938 he went to the United States because of the difficult political situation in Hungary, especially for Jews. He spent much of his time in the United States, except for 1954 to 1962, when he was banned as part of the paranoia of the McCarthy era. He also spent considerable time in Israel.

Erdős made many significant contributions to combinatorics and to number theory. One of the discoveries of which he was most proud is his elementary proof (in the sense that it does not use any complex analysis) of the Prime Number Theorem, which provides an estimate for the number of primes not exceeding a fixed positive integer. He also participated in the modern development of the Ramsey theory.

Erdős traveled extensively throughout the world to work with other mathematicians, visiting conferences, universities, and research laboratories. He had no permanent home. He devoted himself almost entirely to mathematics, traveling from one mathematician to the next, proclaiming "My brain is open." Erdős was the author or coauthor of more than 1500 papers and had more than 500 coauthors. Copies of his articles are kept by Ron Graham, a famous discrete mathematician with whom he collaborated extensively and who took care of many of his worldly needs.

Erdős offered rewards, ranging from \$10 to \$10,000, for the solution of problems that he found particularly interesting, with the size of the reward depending on the difficulty of the problem. He paid out close to \$4000. Erdős had his own special language, using such terms as "epsilon" (child), "boss" (woman), "slave" (man), "captured" (married), "liberated" (divorced), "Supreme Fascist" (God), "Sam" (United States), and "Joe" (Soviet Union). Although he was curious about many things, he concentrated almost all his energy on mathematical research. He had no hobbies and no full-time job. He never married and apparently remained celibate. Erdős was extremely generous, donating much of the money he collected from prizes, awards, and stipends for scholarships and to worthwhile causes. He traveled extremely lightly and did not like having many material possessions.

40. Show that if x and y are elements of a lattice L, then $x \vee y = y$ if and only if $x \wedge y = x$.

A lattice L is **bounded** if it has both an **upper bound**, denoted by 1, such that $x \preccurlyeq 1$ for all $x \in L$ and a **lower bound**, denoted by 0, such that $0 \preccurlyeq x$ for all $x \in L$.

41. Show that if L is a bounded lattice with upper bound 1 and lower bound 0 then these properties hold for all elements $x \in L$.

a) $x \vee 1 = 1$ **b)** $x \wedge 1 = x$
c) $x \vee 0 = x$ **d)** $x \wedge 0 = 0$

42. Show that every finite lattice is bounded.

A lattice is called **distributive** if $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ for all x, y, and z in L.

***43.** Give an example of a lattice that is not distributive.

44. Show that the lattice $(P(S), \subseteq)$ where $P(S)$ is the power set of a finite set S is distributive.

45. Is the lattice $(\mathbf{Z}^+, |)$ distributive?

The **complement** of an element a of a bounded lattice L with upper bound 1 and lower bound 0 is an element b such that $a \vee b = 1$ and $a \wedge b = 0$. Such a lattice is **complemented** if every element of the lattice has a complement.

46. Give an example of a finite lattice where at least one element has more than one complement and at least one element has no complement.

47. Show that the lattice $(P(S), \subseteq)$ where $P(S)$ is the power set of a finite set S is complemented.

***48.** Show that if L is a finite distributive lattice, then an element of L has at most one complement.

The game of Chomp, introduced in Example 12 in Section 1.7, can be generalized for play on any finite partially ordered set $(S, \preceq)$ with a least element a. In this game, a move consists of selecting an element x in S and removing x and all elements larger than it from S. The loser is the player who is forced to select the least element a.

49. Show that the game of Chomp with cookies arranged in an $m \times n$ rectangular grid, described in Example 12 in Section 1.7, is the same as the game of Chomp on the poset $(S, |)$, where S is the set of all positive integers that divide $p^{m-1}q^{n-1}$, where p and q are distinct primes.

50. Show that if $(S, \preceq)$ has a greatest element b, then a winning strategy for Chomp on this poset exists. [*Hint:* Generalize the argument in Example 12 in Section 1.7.]

Computer Projects

Write programs with these input and output.

1. Given the matrix representing a relation on a finite set, determine whether the relation is reflexive and/or irreflexive.

2. Given the matrix representing a relation on a finite set, determine whether the relation is symmetric and/or antisymmetric.

3. Given the matrix representing a relation on a finite set, determine whether the relation is transitive.

4. Given a positive integer n, display all the relations on a set with n elements.

***5.** Given a positive integer n, determine the number of transitive relations on a set with n elements.

***6.** Given a positive integer n, determine the number of equivalence relations on a set with n elements.

***7.** Given a positive integer n, display all the equivalence relations on the set of the n smallest positive integers.

8. Given an n-ary relation, find the projection of this relation when specified fields are deleted.

9. Given an m-ary relation and an n-ary relation, and a set of common fields, find the join of these relations with respect to these common fields.

10. Given the matrix representing a relation on a finite set, find the matrix representing the reflexive closure of this relation.

11. Given the matrix representing a relation on a finite set, find the matrix representing the symmetric closure of this relation.

12. Given the matrix representing a relation on a finite set, find the matrix representing the transitive closure of this relation by computing the join of the powers of the matrix representing the relation.

13. Given the matrix representing a relation on a finite set, find the matrix representing the transitive closure of this relation using Warshall's algorithm.

14. Given the matrix representing a relation on a finite set, find the matrix representing the smallest equivalence relation containing this relation.

15. Given a partial ordering on a finite set, find a total ordering compatible with it using topological sorting.

EXAMPLE 13 Extra Examples In ALGOL 60 an identifier (which is the name of an entity such as a variable) consists of a string of alphanumeric characters (that is, letters and digits) and must begin with a letter. We can use these rules in Backus–Naur to describe the set of allowable identifiers:

$\langle identifier\rangle ::= \langle letter\rangle \mid \langle identifier\rangle\langle letter\rangle \mid \langle identifier\rangle\langle digit\rangle$
$\langle letter\rangle ::= a \mid b \mid \cdots \mid y \mid z$ the ellipsis indicates that all 26 letters are included
$\langle digit\rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

For example, we can produce the valid identifier $x99a$ by using the first rule to replace $\langle identifier\rangle$ by $\langle identifier\rangle\langle letter\rangle$, the second rule to obtain $\langle identifier\rangle a$, the first rule twice to obtain $\langle identifier\rangle\langle digit\rangle\langle digit\rangle a$, the third rule twice to obtain $\langle identifier\rangle 99a$, the first rule to obtain $\langle letter\rangle 99a$, and finally the second rule to obtain $x99a$. ◀

EXAMPLE 14 What is the Backus–Naur form of the grammar for the subset of English described in the introduction to this section?

Solution: The Backus–Naur form of this grammar is

$\langle sentence\rangle ::= \langle noun\ phrase\rangle\langle verb\ phrase\rangle$
$\langle noun\ phrase\rangle ::= \langle article\rangle\langle adjective\rangle\langle noun\rangle \mid \langle article\rangle\langle noun\rangle$
$\langle verb\ phrase\rangle ::= \langle verb\rangle\langle adverb\rangle \mid \langle verb\rangle$
$\langle article\rangle ::= a \mid the$
$\langle adjective\rangle ::= large \mid hungry$
$\langle noun\rangle ::= rabbit \mid mathematician$
$\langle verb\rangle ::= eats \mid hops$
$\langle adverb\rangle ::= quickly \mid wildly$ ◀

EXAMPLE 15 Give the Backus–Naur form for the production of signed integers in decimal notation. (A **signed integer** is a nonnegative integer preceded by a plus sign or a minus sign.)

Solution: The Backus–Naur form for a grammar that produces signed integers is

$\langle signed\ integer\rangle ::= \langle sign\rangle\langle integer\rangle$
$\langle sign\rangle ::= + \mid -$
$\langle integer\rangle ::= \langle digit\rangle \mid \langle digit\rangle\langle integer\rangle$
$\langle digit\rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ ◀

The Backus–Naur form, with a variety of extensions, is used extensively to specify the syntax of programming languages, such as Java and LISP; database languages, such as SQL; and markup languages, such as XML. Some extensions of the Backus–Naur form that are commonly used in the description of programming languages are introduced in the preamble to Exercise 34 at the end of this section.

Exercises

Exercises 1–3 refer to the grammar with start symbol **sentence**, set of terminals $T = \{the, sleepy, happy, tortoise, hare, passes, runs, quickly, slowly\}$, set of nonterminals $N =$ {**noun phrase, transitive verb phrase, intransitive verb phrase, article, adjective, noun, verb, adverb**}, and productions:

sentence → **noun phrase transitive verb phrase noun phrase**
sentence → **noun phrase intransitive verb phrase**
noun phrase → **article adjective noun**
noun phrase → **article noun**
transitive verb phrase → **transitive verb**
intransitive verb phrase → **intransitive verb adverb**
intransitive verb phrase → **intransitive verb**
article → *the*

adjective $\rightarrow$ *sleepy*
adjective $\rightarrow$ *happy*
noun $\rightarrow$ *tortoise*
noun $\rightarrow$ *hare*
transitive verb $\rightarrow$ *passes*
intransitive verb $\rightarrow$ *runs*
adverb $\rightarrow$ *quickly*
adverb $\rightarrow$ *slowly*

1. Use the set of productions to show that each of these sentences is a valid sentence.
  a) *the happy hare runs*
  b) *the sleepy tortoise runs quickly*
  c) *the tortoise passes the hare*
  d) *the sleepy hare passes the happy tortoise*

2. Find five other valid sentences, besides those given in Exercise 1.

3. Show that *the hare runs the sleepy tortoise* is not a valid sentence.

4. Let $G = (V, T, S, P)$ be the phrase-structure grammar with $V = \{0, 1, A, S\}$, $T = \{0, 1\}$, and set of productions P consisting of $S \rightarrow 1S$, $S \rightarrow 00A$, $A \rightarrow 0A$, and $A \rightarrow 0$.
  a) Show that 111000 belongs to the language generated by G.
  b) Show that 11001 does not belong to the language generated by G.
  c) What is the language generated by G?

5. Let $G = (V, T, S, P)$ be the phrase-structure grammar with $V = \{0, 1, A, B, S\}$, $T = \{0, 1\}$, and set of productions P consisting of $S \rightarrow 0A$, $S \rightarrow 1A$, $A \rightarrow 0B$, $B \rightarrow 1A$, $B \rightarrow 1$.
  a) Show that 10101 belongs to the language generated by G.
  b) Show that 10110 does not belong to the language generated by G.
  c) What is the language generated by G?

***6.** Let $V = \{S, A, B, a, b\}$ and $T = \{a, b\}$. Find the language generated by the grammar (V, T, S, P) when the set P of productions consists of
  a) $S \rightarrow AB$, $A \rightarrow ab$, $B \rightarrow bb$.
  b) $S \rightarrow AB$, $S \rightarrow aA$, $A \rightarrow a$, $B \rightarrow ba$.
  c) $S \rightarrow AB$, $S \rightarrow AA$, $A \rightarrow aB$, $A \rightarrow ab$, $B \rightarrow b$.
  d) $S \rightarrow AA$, $S \rightarrow B$, $A \rightarrow aaA$, $A \rightarrow aa$, $B \rightarrow bB$, $B \rightarrow b$.
  e) $S \rightarrow AB$, $A \rightarrow aAb$, $B \rightarrow bBa$, $A \rightarrow \lambda$, $B \rightarrow \lambda$.

7. Construct a derivation of $0^3 1^3$ using the grammar given in Example 5.

8. Show that the grammar given in Example 5 generates the set $\{0^n 1^n \mid n = 0, 1, 2, \ldots\}$.

9. a) Construct a derivation of $0^2 1^4$ using the grammar G_1 in Example 6.
  b) Construct a derivation of $0^2 1^4$ using the grammar G_2 in Example 6.

10. a) Show that the grammar G_1 given in Example 6 generates the set $\{0^m 1^n \mid m, n = 0, 1, 2, \ldots\}$.
  b) Show that the grammar G_2 in Example 6 generates the same set.

11. Construct a derivation of $0^2 1^2 2^2$ in the grammar given in Example 7.

***12.** Show that the grammar given in Example 7 generates the set $\{0^n 1^n 2^n \mid n = 0, 1, 2, \ldots\}$.

13. Find a phrase-structure grammar for each of these languages.
  a) the set consisting of the bit strings 0, 1, and 11
  b) the set of bit strings containing only 1s
  c) the set of bit strings that start with 0 and end with 1
  d) the set of bit strings that consist of a 0 followed by an even number of 1s

14. Find a phrase-structure grammar for each of these languages.
  a) the set consisting of the bit strings 10, 01, and 101
  b) the set of bit strings that start with 00 and end with one or more 1s
  c) the set of bit strings consisting of an even number of 1s followed by a final 0
  d) the set of bit strings that have neither two consecutive 0s nor two consecutive 1s

***15.** Find a phrase-structure grammar for each of these languages.
  a) the set of all bit strings containing an even number of 0s and no 1s
  b) the set of all bit strings made up of a 1 followed by an odd number of 0s
  c) the set of all bit strings containing an even number of 0s and an even number of 1s
  d) the set of all strings containing 10 or more 0s and no 1s
  e) the set of all strings containing more 0s than 1s
  f) the set of all strings containing an equal number of 0s and 1s
  g) the set of all strings containing an unequal number of 0s and 1s

16. Construct phrase-structure grammars to generate each of these sets.
  a) $\{1^n \mid n \geq 0\}$ **b)** $\{10^n \mid n \geq 0\}$
  c) $\{(11)^n \mid n \geq 0\}$

17. Construct phrase-structure grammars to generate each of these sets.
  a) $\{0^n \mid n \geq 0\}$ **b)** $\{1^n 0 \mid n \geq 0\}$
  c) $\{(000)^n \mid n \geq 0\}$

18. Construct phrase-structure grammars to generate each of these sets.
  a) $\{01^{2n} \mid n \geq 0\}$
  b) $\{0^n 1^{2n} \mid n \geq 0\}$
  c) $\{0^n 1^m 0^n \mid m \geq 0 \text{ and } n \geq 0\}$

19. Let $V = \{S, A, B, a, b\}$ and $T = \{a, b\}$. Determine whether $G = (V, T, S, P)$ is a type 0 grammar but not a type 1 grammar, a type 1 grammar but not a type 2 grammar, or a type 2 grammar but not a type 3 grammar if P, the set of productions, is

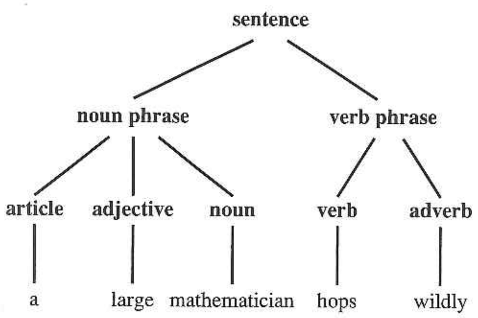
a) $S \to aAB$, $A \to Bb$, $B \to \lambda$.
b) $S \to aA$, $A \to a$, $A \to b$.
c) $S \to ABa$, $AB \to a$.
d) $S \to ABA$, $A \to aB$, $B \to ab$.
e) $S \to bA$, $A \to B$, $B \to a$.
f) $S \to aA$, $aA \to B$, $B \to aA$, $A \to b$.
g) $S \to bA$, $A \to b$, $S \to \lambda$.
h) $S \to AB$, $B \to aAb$, $aAb \to b$.
i) $S \to aA$, $A \to bB$, $B \to b$, $B \to \lambda$.
j) $S \to A$, $A \to B$, $B \to \lambda$.

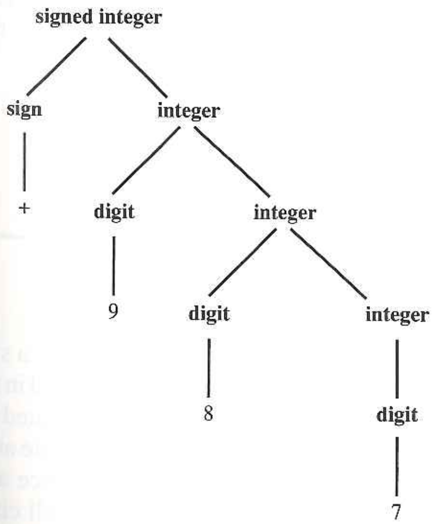
20. A **palindrome** is a string that reads the same backward as it does forward, that is, a string w, where $w = w^R$, where w^R is the reversal of the string w. Find a context-free grammar that generates the set of all palindromes over the alphabet $\{0, 1\}$.

***21.** Let G_1 and G_2 be context-free grammars, generating the languages $L(G_1)$ and $L(G_2)$, respectively. Show that there is a context-free grammar generating each of these sets.

a) $L(G_1) \cup L(G_2)$ b) $L(G_1)L(G_2)$
c) $L(G_1)^*$

22. Find the strings constructed using the derivation trees shown here.





23. Construct derivation trees for the sentences in Exercise 1.

24. Let G be the grammar with $V = \{a, b, c, S\}$; $T = \{a, b, c\}$; starting symbol S; and productions $S \to abS$, $S \to bcS$, $S \to bbS$, $S \to a$, and $S \to cb$. Construct derivation trees for

a) $bcbba$.
b) $bbbcbba$.
c) $bcabbbbbcb$.

***25.** Use top-down parsing to determine whether each of the following strings belongs to the language generated by the grammar in Example 12.

a) $baba$ b) $abab$
c) $cbaba$ d) $bbbcba$

***26.** Use bottom-up parsing to determine whether the strings in Exercise 25 belong to the language generated by the grammar in Example 12.

27. Construct a derivation tree for -109 using the grammar given in Example 15.

28. a) Explain what the productions are in a grammar if the Backus–Naur form for productions is as follows:

$$\begin{aligned}\langle expression\rangle &::= (\langle expression\rangle) \mid \\ &\quad\ \langle expression\rangle + \langle expression\rangle \mid \\ &\quad\ \langle expression\rangle * \langle expression\rangle \mid \\ &\quad\ \langle variable\rangle \\ \langle variable\rangle &::= x \mid y\end{aligned}$$

b) Find a derivation tree for $(x * y) + x$ in this grammar.

29. a) Construct a phrase-structure grammar that generates all signed decimal numbers, consisting of a sign, either $+$ or $-$; a nonnegative integer; and a decimal fraction that is either the empty string or a decimal point followed by a positive integer, where initial zeros in an integer are allowed.
b) Give the Backus–Naur form of this grammar.
c) Construct a derivation tree for -31.4 in this grammar.

30. a) Construct a phrase-structure grammar for the set of all fractions of the form a/b, where a is a signed integer in decimal notation and b is a positive integer.
b) What is the Backus–Naur form for this grammar?
c) Construct a derivation tree for $+311/17$ in this grammar.

31. Give production rules in Backus–Naur form for an identifier if it can consist of

a) one or more lowercase letters.
b) at least three but no more than six lowercase letters.
c) one to six uppercase or lowercase letters beginning with an uppercase letter.
d) a lowercase letter, followed by a digit or an underscore, followed by three or four alphanumeric characters (lower or uppercase letters and digits).

32. Give production rules in Backus–Naur form for the name of a person if this name consists of a first name, which is a string of letters, where only the first letter is uppercase; a middle initial; and a last name, which can be any string of letters.

33. Give production rules in Backus–Naur form that generate all identifiers in the C programming language. In C an identifier starts with a letter or an underscore (_) that is followed by one or more lowercase letters, uppercase letters, underscores, and digits.

Several extensions to Backus–Naur form are commonly used to define phrase-structure grammars. In one such extension, a question mark (?) indicates that the symbol, or group of symbols inside parentheses, to its left can appear zero or once (that is, it is optional), an asterisk (*) indicates that the symbol to its left can appear zero or more times, and a plus (+) indicates that the symbol to its left can appear one or more times. These extensions are part of **extended Backus–Naur form (EBNF),** and the symbols ?, *, and + are called **metacharacters.** In EBNF the brackets used to denote nonterminals are usually not shown.

34. Describe the set of strings defined by each of these sets of productions in EBNF.

a) $string ::= L{+}D?L{+}$
$L ::= a \mid b \mid c$
$D ::= 0 \mid 1$

b) $string ::= sign\ D{+} \mid D{+}$
$sign ::= + \mid -$
$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

c) $string ::= L{*}(D{+})?L{*}$
$L ::= x \mid y$
$D ::= 0 \mid 1$

35. Give production rules in extended Backus–Naur form that generate all decimal numerals consisting of an optional sign, a nonnegative integer, and a decimal fraction that is either the empty string or a decimal point followed by an optional positive integer optionally preceded by some number of zeros.

36. Give production rules in extended Backus–Naur form that generate a sandwich if a sandwich consists of a lower slice of bread; mustard or mayonnaise; optional lettuce; an optional slice of tomato; one or more slices of either turkey, chicken, or roast beef (in any combination); optionally some number of slices of cheese; and a top slice of bread.

37. Give production rules in extended Backus–Naur form for identifiers in the C programming language (see Exercise 33).

38. Describe how productions for a grammar in extended Backus–Naur form can be translated into a set of productions for the grammar in Backus–Naur form.

This is the Backus–Naur form that describes the syntax of expressions in postfix (or reverse Polish) notation.

$\langle expression\rangle ::= \langle term\rangle \mid \langle term\rangle\langle term\rangle\langle addOperator\rangle$
$\langle addOperator\rangle ::= + \mid -$
$\langle term\rangle ::= \langle factor\rangle \mid \langle factor\rangle\langle factor\rangle\langle mulOperator\rangle$
$\langle mulOperator\rangle ::= * \mid /$
$\langle factor\rangle ::= \langle identifier\rangle \mid \langle expression\rangle$
$\langle identifier\rangle ::= a \mid b \mid \cdots \mid z$

39. For each of these strings, determine whether it is generated by the grammar given for postfix notation. If it is, find the steps used to generate the string

a) $abc{*}{+}$ **b)** $xy{+}{+}$ **c)** $xy{-}z{*}$
d) $wxyz{-}{*}/$ **e)** $ade{-}{*}$

40. Use Backus–Naur form to describe the syntax of expressions in infix notation, where the set of operators and identifiers is the same as in the BNF for postfix expressions given in the preamble to Exercise 39, but parentheses must surround expressions being used as factors.

41. For each of these strings, determine whether it is generated by the grammar for infix expressions from Exercise 40. If it is, find the steps used to generate the string.

a) $x + y + z$ **b)** $a/b + c/d$
c) $m * (n + p)$ **d)** $+m - n + p - q$
e) $(m + n) * (p - q)$

42. Let G be a grammar and let R be the relation containing the ordered pair (w_0, w_1) if and only if w_1 is directly derivable from w_0 in G. What is the reflexive transitive closure of R?

12.2 Finite-State Machines with Output

Introduction

Links

Many kinds of machines, including components in computers, can be modeled using a structure called a finite-state machine. Several types of finite-state machines are commonly used in models. All these versions of finite-state machines include a finite set of states, with a designated starting state, an input alphabet, and a transition function that assigns a next state to every state and input pair. Finite-state machines are used extensively in applications in computer science and data networking. For example, finite-state machines are the basis for programs for spell checking, grammar checking, indexing or searching large bodies of text, recognizing speech, transforming text using markup languages such as XML and HTML, and network protocols that specify how computers communicate.

Exercises

1. Draw the state diagrams for the finite-state machines with these state tables.

   a)

	f		g	
	Input		*Input*	
State	0	1	0	1
s_0	s_1	s_0	0	1
s_1	s_0	s_2	0	1
s_2	s_1	s_1	0	0

   b)

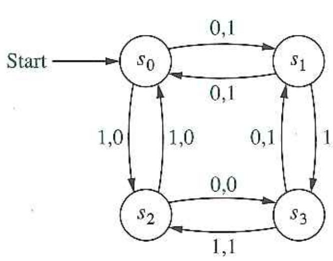
	f		g	
	Input		*Input*	
State	0	1	0	1
s_0	s_1	s_0	0	0
s_1	s_2	s_0	1	1
s_2	s_0	s_3	0	1
s_3	s_1	s_2	1	0

   c)

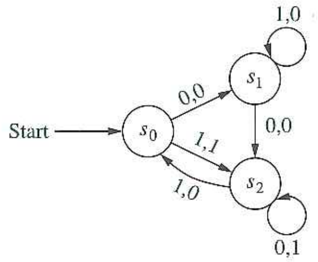
	f		g	
	Input		*Input*	
State	0	1	0	1
s_0	s_0	s_4	1	1
s_1	s_0	s_3	0	1
s_2	s_0	s_2	0	0
s_3	s_1	s_1	1	1
s_4	s_1	s_0	1	0

2. Give the state tables for the finite-state machines with these state diagrams.

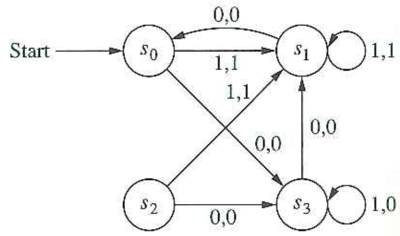
   a)



   b)



   c)



3. Find the output generated from the input string 01110 for the finite-state machine with the state table in
   a) Exercise 1(a).
   b) Exercise 1(b).
   c) Exercise 1(c).
4. Find the output generated from the input string 10001 for the finite-state machine with the state diagram in
   a) Exercise 2(a).
   b) Exercise 2(b).
   c) Exercise 2(c).
5. Find the output for each of these input strings when given as input to the finite-state machine in Example 2.
   a) 0111 b) 11011011
   c) 01010101010
6. Find the output for each of these input strings when given as input to the finite-state machine in Example 3.
   a) 0000 b) 101010
   c) 11011100010
7. Construct a finite-state machine that models an old-fashioned soda machine that accepts nickels, dimes, and quarters. The soda machine accepts change until 35 cents has been put in. It gives change back for any amount greater than 35 cents. Then the customer can push buttons to receive either a cola, a root beer, or a ginger ale.
8. Construct a finite-state machine that models a newspaper vending machine that has a door that can be opened only after either three dimes (and any number of other coins) or a quarter and a nickel (and any number of other coins) have been inserted. Once the door can be opened, the customer opens it and takes a paper, closing the door. No change is ever returned no matter how much extra money has been inserted. The next customer starts with no credit.

9. Construct a finite-state machine that delays an input string two bits, giving 00 as the first two bits of output.

10. Construct a finite-state machine that changes every other bit, starting with the second bit, of an input string, and leaves the other bits unchanged.

11. Construct a finite-state machine for the log-on procedure for a computer, where the user logs on by entering a user identification number, which is considered to be a single input, and then a password, which is considered to be a single input. If the password is incorrect, the user is asked for the user identification number again.

12. Construct a finite-state machine for a combination lock that contains numbers 1 through 40 and that opens only when the correct combination, 10 right, 8 second left, 37 right, is entered. Each input is a triple consisting of a number, the direction of the turn, and the number of times the lock is turned in that direction.

13. Construct a finite-state machine for a toll machine that opens a gate after 25 cents, in nickels, dimes, or quarters, has been deposited. No change is given for overpayment, and no credit is given to the next driver when more than 25 cents has been deposited.

14. Construct a finite-state machine for entering a security code into an automatic teller machine (ATM) that implements these rules: A user enters a string of four digits, one digit at a time. If the user enters the correct four digits of the password, the ATM displays a welcome screen. When the user enters an incorrect string of four digits, the ATM displays a screen that informs the user that an incorrect password was entered. If a user enters the incorrect password three times, the account is locked.

15. Construct a finite-state machine for a restricted telephone switching system that implements these rules. Only calls to the telephone numbers 0, 911, and the digit 1 followed by 10-digit telephone numbers that begin with 212, 800, 866, 877, and 888 are sent to the network. All other strings of digits are blocked by the system and the user hears an error message.

16. Construct a finite-state machine that gives an output of 1 if the number of input symbols read so far is divisible by 3 and an output of 0 otherwise.

17. Construct a finite-state machine that determines whether the input string has a 1 in the last position and a 0 in the third to the last position read so far.

18. Construct a finite-state machine that determines whether the input string read so far ends in at least five consecutive 1s.

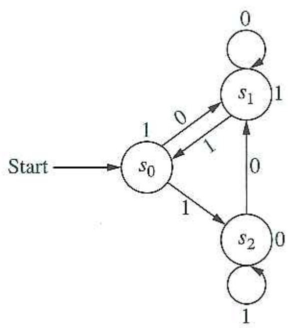
19. Construct a finite-state machine that determines whether the word *computer* has been read as the last eight characters in the input read so far, where the input can be any string of English letters.

A **Moore machine** $M = (S, I, O, f, g, s_0)$ consists of a finite set of states, an input alphabet I, an output alphabet O, a transition function f that assigns a next state to every pair of a state and an input, an output function g that assigns an output to every state, and a starting state s_0. A Moore machine can be represented either by a table listing the transitions for each pair of state and input and the outputs for each state, or by a state diagram that displays the states, the transitions between states, and the output for each state. In the diagram, transitions are indicated with arrows labeled with the input, and the outputs are shown next to the states.

20. Construct the state diagram for the Moore machine with this state table.

	f		
	Input		
State	0	1	*g*
s_0	s_0	s_2	0
s_1	s_3	s_0	1
s_2	s_2	s_1	1
s_3	s_2	s_0	1

21. Construct the state table for the Moore machine with the state diagram shown here. Each input string to a Moore machine M produces an output string. In particular, the output corresponding to the input string $a_1a_2\ldots a_k$ is the string $g(s_0)g(s_1)\ldots g(s_k)$, where $s_i = f(s_{i-1}, a_i)$ for $i = 1, 2, \ldots, k$.



22. Find the output string generated by the Moore machine in Exercise 20 with each of these input strings.

a) 0101 **b)** 111111

c) 11101110111

23. Find the output string generated by the Moore machine in Exercise 21 with each of the input strings in Exercise 22.

24. Construct a Moore machine that gives an output of 1 whenever the number of symbols in the input string read so far is divisible by 4.

25. Construct a Moore machine that determines whether an input string contains an even or odd number of 1s. The machine should give 1 as output if an even number of 1s are in the string and 0 as output if an odd number of 1s are in the string.

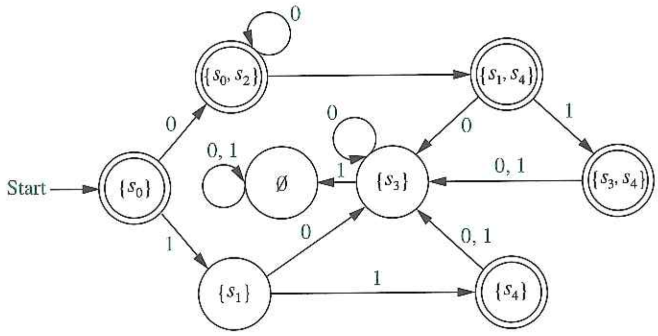


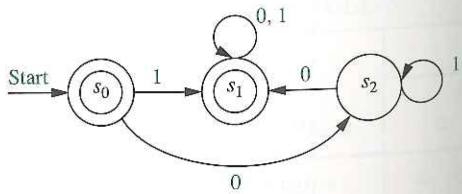
FIGURE 8 A Deterministic Automaton Equivalent to the Nondeterministic Automaton in Example 10.

Exercises

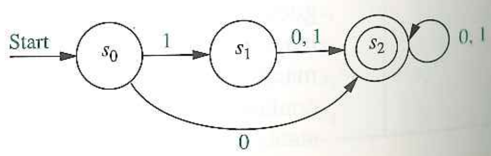
1. Let $A = \{0, 11\}$ and $B = \{00, 01\}$. Find each of these sets.
   a) AB b) BA c) A^2 d) B^3
2. Show that if A is a set of strings, then $A\emptyset = \emptyset A = \emptyset$.
3. Find all pairs of sets of strings A and B for which $AB = \{10, 111, 1010, 1000, 10111, 101000\}$.
4. Show that these equalities hold.
   a) $\{\lambda\}^* = \{\lambda\}$
   b) $(A^*)^* = A^*$ for every set of strings A
5. Describe the elements of the set A^* for these values of A.
   a) $\{10\}$ b) $\{111\}$ c) $\{0, 01\}$ d) $\{1, 101\}$
6. Let V be an alphabet, and let A and B be subsets of V^*. Show that $|AB| \leq |A||B|$.
7. Let V be an alphabet, and let A and B be subsets of V^* with $A \subseteq B$. Show that $A^* \subseteq B^*$.
8. Suppose that A is a subset of V^*, where V is an alphabet. Prove or disprove each of these statements.
   a) $A \subseteq A^2$ b) if $A = A^2$, then $\lambda \in A$
   c) $A\{\lambda\} = A$ d) $(A^*)^* = A^*$
   e) $A^*A = A^*$ f) $|A^n| = |A|^n$
9. Determine whether the string 11101 is in each of these sets.
   a) $\{0, 1\}^*$ b) $\{1\}^*\{0\}^*\{1\}^*$
   c) $\{11\}\{0\}^*\{01\}$ d) $\{11\}^*\{01\}^*$
   e) $\{111\}^*\{0\}^*\{1\}$ f) $\{11, 0\}\{00, 101\}$
10. Determine whether the string 01001 is in each of these sets.
    a) $\{0, 1\}^*$ b) $\{0\}^*\{10\}\{1\}^*$
    c) $\{010\}^*\{0\}^*\{1\}$ d) $\{010, 011\}\{00, 01\}$
    e) $\{00\}\{0\}^*\{01\}$ f) $\{01\}^*\{01\}^*$
11. Determine whether each of these strings is recognized by the deterministic finite-state automaton in Figure 1.
    a) 111 b) 0011 c) 1010111 d) 011011011
12. Determine whether each of these strings is recognized by the deterministic finite-state automaton in Figure 1.
    a) 010 b) 1101 c) 1111110 d) 010101010
13. Determine whether all the strings in each of these sets are recognized by the deterministic finite-state automaton in Figure 1.
    a) $\{0\}^*$ b) $\{0\}\{0\}^*$ c) $\{1\}\{0\}^*$
    d) $\{01\}^*$ e) $\{0\}^*\{1\}^*$ f) $\{1\}\{0, 1\}^*$
14. Show that if $M = (S, I, f, s_0, F)$ is a deterministic finite-state automaton and $f(s, x) = s$ for the state $s \in S$ and the input string $x \in I^*$, then $f(s, x^n) = s$ for every nonnegative integer n. (Here x^n is the concatenation of n copies of the string x, defined recursively in Exercise 37 in Section 4.3.)
15. Given a deterministic finite-state automaton $M = (S, I, f, s_0, F)$, use structural induction and the recursive definition of the extended transition function f to prove that $f(s, xy) = f(f(s, x), y)$ for all states $s \in S$ and all strings $x \in I^*$ and $y \in I^*$.

In Exercises 16–22 find the language recognized by the given deterministic finite-state automaton.

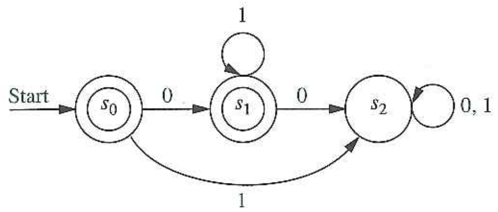
16.



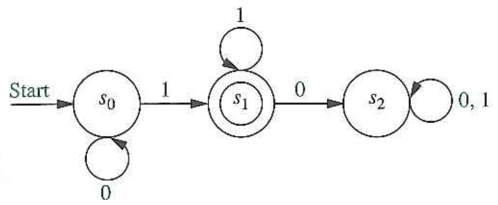
17.



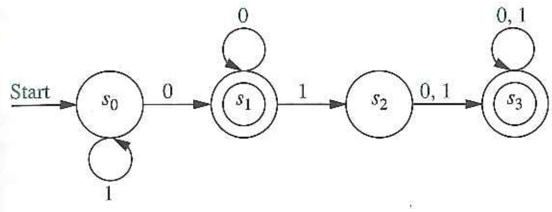
18.



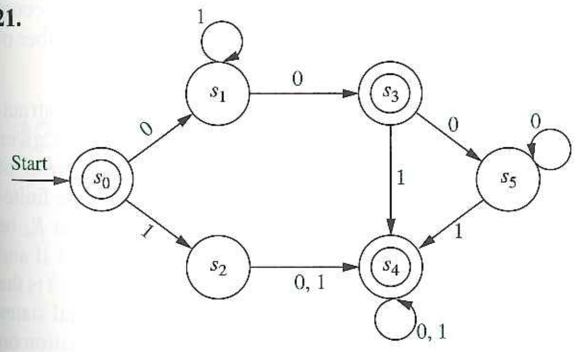
19.



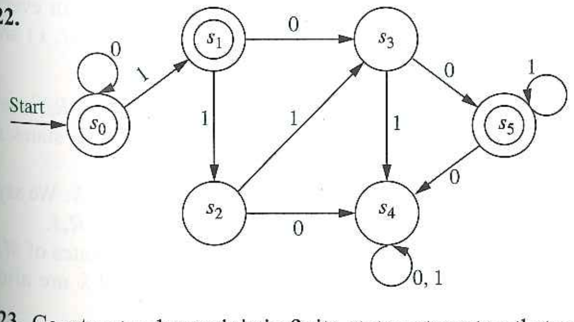
20.



21.



22.



23. Construct a deterministic finite-state automaton that recognizes the set of all bit strings beginning with 01.

24. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that end with 10.

25. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain the string 101.

26. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that do not contain three consecutive 0s.

27. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain exactly three 0s.

28. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain at least three 0s.

29. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain three consecutive 1s.

30. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that begin with 0 or with 11.

31. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that begin and end with 11.

32. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain an even number of 1s.

33. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain an odd number of 0s.

34. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain an even number of 0s and an odd number of 1s.

35. Construct a finite-state automaton that recognizes the set of bit strings consisting of a 0 followed by a string with an odd number of 1s.

36. Construct a finite-state automaton with four states that recognizes the set of bit strings containing an even number of 1s and an odd number of 0s.

37. Show that there is no finite-state automaton with two states that recognizes the set of all bit strings that have one or more 1 bits and end with a 0.

38. Show that there is no finite-state automaton with three states that recognizes the set of bit strings containing an even number of 1s and an even number of 0s.

39. Explain how you can change the deterministic finite-state automaton M so that the changed automaton recognizes the set $I^* - L(M)$.

40. Use Exercise 39 and finite-state automata constructed in Example 6 to find deterministic finite-state automata that recognize each of these sets.

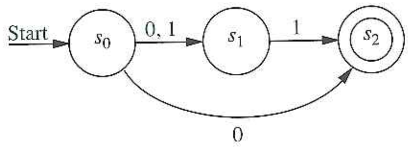
a) the set of bit strings that do not begin with two 0s
b) the set of bit strings that do not end with two 0s
c) the set of bit strings that contain at most one 0 (that is, that do not contain at least two 0s)

41. Use the procedure you described in Exercise 39 and the finite-state automata you constructed in Exercise 25 to find a deterministic finite-state automaton that recognizes the set of all bit strings that do not contain the string 101.

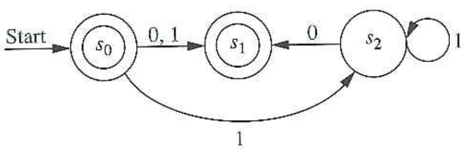
42. Use the procedure you described in Exercise 39 and the finite-state automaton you constructed in Exercise 29 to find a deterministic finite-state automaton that recognizes the set of all bit strings that do not contain three consecutive 1s.

In Exercises 43–49 find the language recognized by the given nondeterministic finite-state automaton.

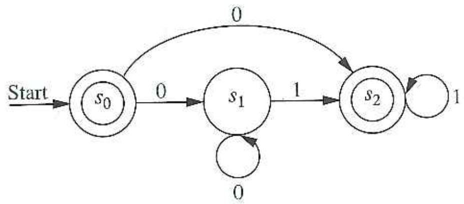
43.



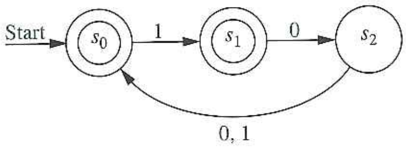
44.



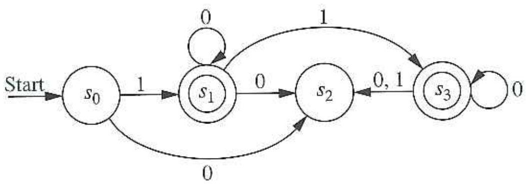
45.



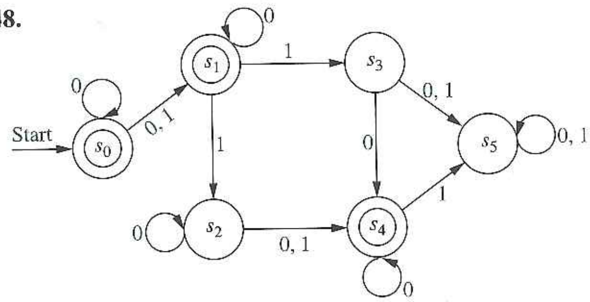
46.



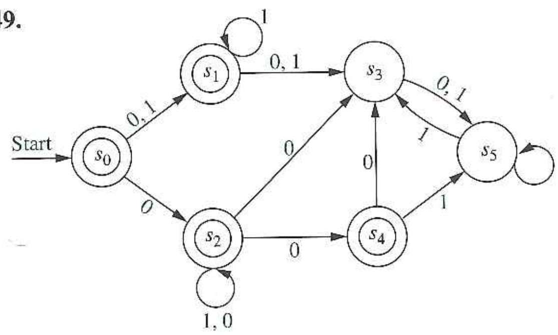
47.



48.



49.



50. Find a deterministic finite-state automaton that recognizes the same language as the nondeterministic finite-state automaton in Exercise 43.

51. Find a deterministic finite-state automaton that recognizes the same language as the nondeterministic finite-state automaton in Exercise 44.

52. Find a deterministic finite-state automaton that recognizes the same language as the nondeterministic finite-state automaton in Exercise 45.

53. Find a deterministic finite-state automaton that recognizes the same language as the nondeterministic finite-state automaton in Exercise 46.

54. Find a deterministic finite-state automaton that recognizes the same language as the nondeterministic finite-state automaton in Exercise 47.

55. Find a deterministic finite-state automaton that recognizes each of these sets.

a) $\{0\}$ **b)** $\{1, 00\}$ **c)** $\{1^n \mid n = 2, 3, 4, \ldots\}$

56. Find a nondeterministic finite-state automaton that recognizes each of the languages in Exercise 27, and has fewer states, if possible, than the deterministic automaton you found in that exercise.

***57.** Show that there is no finite-state automaton that recognizes the set of bit strings containing an equal number of 0s and 1s.

In Exercises 58–62 we introduce a technique for constructing a deterministic finite-state machine equivalent to a given deterministic finite-state machine with the least number of states possible. Suppose that $M = (S, I, f, s_0, F)$ is a finite-state automaton and that k is a nonnegative integer. Let R_k be the relation on the set S of states of M such that $s R_k t$ if and only if for every input string x with $l(x) \leq k$ [where $l(x)$ is the length of x, as usual], $f(s, x)$ and $f(t, x)$ are both final states or both not final states. Furthermore, let R_* be the relation on the set of states of M such that $s R_* t$ if and only if for every input string x, regardless of length, $f(s, x)$ and $f(t, x)$ are both final states or both not final states.

***58. a)** Show that for every nonnegative integer k, R_k is an equivalence relation on S. We say that two states s and t are ***k*-equivalent** if $s R_k t$.

b) Show that R_* is an equivalence relation on S. We say that two states s and t are **$*$-equivalent** if $s R_* t$.

c) Show that if s and t are two k-equivalent states of M, where k is a positive integer, then s and k are also $(k-1)$-equivalent

d) Show that the equivalence classes of R_k are a refinement of the equivalence classes of R_{k-1} if k is a positive integer. (The refinement of a partition of a set is defined in the preamble to Exercise 49 in Section 8.5.)

e) Show that if s and t are k-equivalent for every nonnegative integer k, then they are $*$-equivalent.

f) Show that all states in a given R_*-equivalence class are final states or all are not final states.

g) Show that if s and t are R_*-equivalent, then $f(s, a)$ and $f(t, a)$ are also R_*-equivalent for all $a \in I$.

***59.** Show that there is a nonnegative integer n such that the set of n-equivalence classes of states of M is the same as the set of $(n+1)$-equivalence classes of states of M. Then show for this integer n, the set of n-equivalence classes of states of M equals the set of $*$-equivalence classes of states of M.

The **quotient automaton** $\overline{M}$ of the deterministic finite-state automaton $M = (S, I, f, s_0, F)$ is the finite-state automaton $(\overline{S}, I, \overline{f}, [s_0]_{R_*}, \overline{F})$, where the set of states $\overline{S}$ is the set of $*$-equivalence classes of S, the transition function $\overline{f}$ is defined by $\overline{f}([s]_{R_*}, a) = [f(s, a)]_{R_*}$ for all states $[s]$ of $\overline{M}$ and input symbols $a \in I$, and $\overline{F}$ is the set consisting of R_*-equivalence classes of final states of M.

***60.** **a)** Show that s and t are 0-equivalent if and only if either both s and t are final states or neither s nor t is a final state. Conclude that each final state of $\overline{M}$, which is an R_*-equivalence class, contains only final states of M.

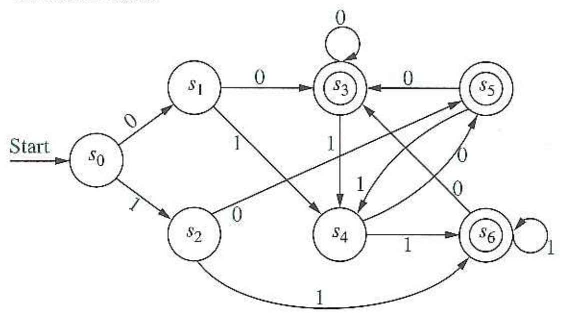
b) Show that if k is a positive integer, then s and t are k-equivalent if and only if s and t are $(k-1)$-equivalent and for every input symbol $a \in I$, $f(s, a)$ and $f(t, a)$ are $(k-1)$-equivalent. Conclude that the transition function $\overline{f}$ is well-defined.

c) Describe a procedure that can be used to construct the quotient automaton of a finite-automaton M.

****61.** **a)** Show that if M is a finite-state automaton, then the quotient automaton $\overline{M}$ recognizes the same language as M.

b) Show that if M is a finite-state automaton with the property that for every state s of M there is a string $x \in I^*$ such that $f(s_0, x) = s$, then the quotient automaton $\overline{M}$ has the minimum number of states of any finite-state automaton equivalent to M.

62. Answer these questions about the finite-state automaton M shown here.



a) Find the k-equivalence classes of M for $k = 0, 1, 2$, and 3. Also, find the $*$-equivalence classes of M.

b) Construct the quotient automaton $\overline{M}$ of M.

12.4 Language Recognition

Introduction

We have seen that finite-state automata can be used as language recognizers. What sets can be recognized by these machines? Although this seems like an extremely difficult problem, there is a simple characterization of the sets that can be recognized by finite state automata. This problem was first solved in 1956 by the American mathematician Stephen Kleene. He showed that there is a finite-state automaton that recognizes a set if and only if this set can be built up from the null set, the empty string, and singleton strings by taking concatenations, unions, and Kleene closures, in arbitrary order. Sets that can be built up in this way are called **regular sets.**

Regular grammars were defined in Section 12.1. Because of the terminology used, it is not surprising that there is a connection between regular sets, which are the sets recognized by finite-state automata, and regular grammars. In particular, a set is regular if and only if it is generated by a regular grammar.

Finally, there are sets that cannot be recognized by any finite-state automata. We will give an example of such a set. We will briefly discuss more powerful models of computation, such as pushdown automata and Turing machines, at the end of this section.

Regular Sets

The regular sets are those that can be formed using the operations of concatenation, union, and Kleene closure in arbitrary order, starting with the empty set, the empty string, and singleton sets. We will see that the regular sets are those that can be recognized using a finite-state automaton. To define regular sets we first need to define regular expressions.

mathematics as the Hilbert problems did for twentieth century mathematics. (See [De02] for more about the Millennium Problems.) As mentioned in Section 3.3, there is an important class of problems, the class of NP-complete problems, such that a problem is in this class if it is in the class NP and if it can be shown that if it is also in the class P, then *every* problem in the class NP must also be in the class P. That is, a problem is NP-complete if the existence of a polynomial-time algorithm for solving it implies the existence of a polynomial-time algorithm for every problem in NP. In this book we have discussed several different NP-complete problems, such as determining whether a simple graph has a Hamilton circuit and determining whether a proposition in n-variables is a tautology.

Exercises

1. Let T be the Turing machine defined by the five-tuples: $(s_0, 0, s_1, 1, R)$, $(s_0, 1, s_1, 0, R)$, $(s_0, B, s_1, 0, R)$, $(s_1, 0, s_2, 1, L)$, $(s_1, 1, s_1, 0, R)$, and $(s_1, B, s_2, 0, L)$. For each of these initial tapes, determine the final tape when T halts, assuming that T begins in initial position.

a)

...	B	B	0	0	1	1	B	B	...

b)

...	B	B	1	0	1	B	B	B	...

c)

...	B	B	1	1	B	0	1	B	...

d)

...	B	B	B	B	B	B	B	B	...

2. Let T be the Turing machine defined by the five-tuples: $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, L)$, $(s_0, B, s_1, 1, R)$, $(s_1, 0, s_2, 1, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, B, s_2, 0, R)$, and $(s_2, B, s_3, 0, R)$. For each of these initial tapes, determine the final tape when T halts, assuming that T begins in initial position.

a)

...	B	B	0	1	0	1	B	B	...

b)

...	B	B	1	1	1	B	B	B	...

c)

...	B	B	0	0	B	0	0	B	...

d)

...	B	B	B	B	B	B	B	B	...

3. What does the Turing machine described by the five-tuples $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 0, R)$, (s_0, B, s_2, B, R), $(s_1, 0, s_1, 0, R)$, $(s_1, 1, s_0, 1, R)$, and (s_1, B, s_2, B, R) do when given

a) 11 as input?

b) an arbitrary bit string as input?

4. What does the Turing machine described by the five-tuples $(s_0, 0, s_0, 1, R)$, $(s_0, 1, s_0, 1, R)$, (s_0, B, s_1, B, L), $(s_1, 1, s_2, 1, R)$, do when given

a) 101 as input?

b) an arbitrary bit string as input?

5. What does the Turing machine described by the five-tuples $(s_0, 1, s_1, 0, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, 0, s_2, 0, R)$, $(s_2, 0, s_3, 1, L)$, $(s_2, 1, s_2, 1, R)$, $(s_3, 1, s_3, 1, L)$, $(s_3, 0, s_4, 0, L)$, $(s_4, 1, s_4, 1, L)$, and $(s_4, 0, s_0, 1, R)$ do when given

a) 11 as input?

b) a bit string consisting entirely of 1s as input?

6. Construct a Turing machine with tape symbols 0, 1, and B that, when given a bit string as input, adds a 1 to the end of the bit string and does not change any of the other symbols on the tape.

7. Construct a Turing machine with tape symbols 0, 1, and B that, when given a bit string as input, replaces the first 0 with a 1 and does not change any of the other symbols on the tape.

8. Construct a Turing machine with tape symbols 0, 1, and B that, given a bit string as input, replaces all 0s on the tape with 1s and does not change any of the 1s on the tape.

9. Construct a Turing machine with tape symbols 0, 1, and B that, given a bit string as input, replaces all but the leftmost 1 on the tape with 0s and does not change any of the other symbols on the tape.

10. Construct a Turing machine with tape symbols 0, 1, and B that, given a bit string as input, replaces the first two consecutive 1s on the tape with 0s and does not change any of the other symbols on the tape.

11. Construct a Turing machine that recognizes the set of all bit strings that end with a 0.

12. Construct a Turing machine that recognizes the set of all bit strings that contain at least two 1s.

13. Construct a Turing machine that recognizes the set of all bit strings that contain an even number of 1s.

14. Show at each step the contents of the tape of the Turing machine in Example 3 starting with each of these strings.

a) 0011 **b)** 00011 **c)** 101100 **d)** 000111

15. Explain why the Turing machine in Example 3 recognizes a bit string if and only if this string is of the form 0^n1^n for some positive integer n.

*16. Construct a Turing machine that recognizes the set $\{0^{2n}1^n \mid n \geq 0\}$.

*17. Construct a Turing machine that recognizes the set $\{0^n1^n2^n \mid n \geq 0\}$.

18. Construct a Turing machine that computes the function $f(n) = n + 2$ for all nonnegative integers n.

19. Construct a Turing machine that computes the function $f(n) = n - 3$ if $n \geq 3$ and $f(n) = 0$ for $n = 0, 1, 2$ for all nonnegative integers n.

20. Construct a Turing machine that computes the function $f(n) = n \bmod 3$.

21. Construct a Turing machine that computes the function $f(n) = 3$ if $n \geq 5$ and $f(n) = 0$ if $n = 0, 1, 2, 3$, or 4.

22. Construct a Turing machine that computes the function $f(n) = 2n$ for all nonnegative integers n.

23. Construct a Turing machine that computes the function $f(n) = 3n$ for all nonnegative integers n.

24. Construct a Turing machine that computes the function $f(n_1, n_2) = n_2 + 2$ for all pairs of nonnegative integers n_1 and n_2.

*25. Construct a Turing machine that computes the function $f(n_1, n_2) = \min(n_1, n_2)$ for all nonnegative integers n_1 and n_2.

26. Construct a Turing machine that computes the function $f(n_1, n_2) = n_1 + n_2 + 1$ for all nonnegative integers n_1 and n_2.

Suppose that T_1 and T_2 are Turing machines with disjoint sets of states S_1 and S_2 and with transition functions f_1 and f_2, respectively. We can define the Turing machine T_1T_2, the **composite** of T_1 and T_2, as follows. The set of states of T_1T_2 is $S_1 \cup S_2$. T_1T_2 begins in the start state of S_1. It first executes the transitions of T_1 using f_1 up to, but not including, the step at which T_1 would halt. Then, for all moves for which T_1 halts, it executes the same transitions of T_1 except that it moves to the start state of T_2. From this point on, the moves of T_1T_2 are the same as the moves of T_2.

27. By finding the composite of the Turing machines you constructed in Exercises 18 and 22, construct a Turing machine that computes the function $f(n) = 2n + 2$.

28. By finding the composite of the Turing machines you constructed in Exercises 18 and 23, construct a Turing machine that computes the function $f(n) = 3(n + 2) = 3n + 6$.

29. Which of the following problems is a decision problem?
   a) What is the smallest prime greater than n?
   b) Is a graph G bipartite?
   c) Given a set of strings, is there a finite-state automaton that recognizes this set of strings?
   d) Given a checkerboard and a particular type of polyomino (see Section 1.7), can this checkerboard be tiled using polyominoes of this type?

30. Which of the following problems is a decision problem?
   a) Is the sequence $a_1, a_2, \ldots, a_n$ of positive integers in increasing order?
   b) Can the vertices of a simple graph G be colored using three colors so that no two adjacent vertices are the same color?
   c) What is the vertex of highest degree in a graph G?
   d) Given two finite-state machines, do these machines recognize the same language?

Let $B(n)$ be the maximum number of 1s that a Turing machine with n states with the alphabet $\{1, B\}$ may print on a tape that is initially blank. The problem of determining $B(n)$ for particular values of n is known as the **busy beaver problem.** This problem was first studied by Tibor Rado in 1962. Currently it is known that $B(2) = 4$, $B(3) = 6$, and $B(4) = 13$, but $B(n)$ is not known for $n \geq 5$. $B(n)$ grows rapidly; it is known that $B(5) \geq 4098$ and $B(6) \geq 1.29 \times 10^{865}$.

*31. Show that $B(2)$ is at least 4 by finding a Turing machine with two states and alphabet $\{1, B\}$ that halts with four consecutive 1s on the tape.

**32. Show that the function $B(n)$ cannot be computed by any Turing machine. [*Hint:* Assume that there is a Turing machine that computes $B(n)$ in binary. Build a Turing machine T that, starting with a blank tape, writes n down in binary, computes $B(n)$ in binary, and converts $B(n)$ from binary to unary. Show that for sufficiently large n, the number of states of T is less than $B(n)$, leading to a contradiction.]

Key Terms and Results

TERMS

alphabet (or vocabulary): a set that contains elements used to form strings

language: a subset of the set of all strings over an alphabet

phrase-structure grammar (V, T, S, P): a description of a language containing an alphabet V, a set of terminal symbols T, a start symbol S, and a set of productions P

the production $w \to w_1$: w can be replaced by w_1 whenever it occurs in a string in the language

$w_1 \Rightarrow w_2$ (w_2 is directly derivable from w_1): w_2 can be obtained from w_1 using a production to replace a string in w_1 with another string

$w_1 \stackrel{*}{\Rightarrow} w_2$ (w_2 is derivable from w_1): w_2 can be obtained from w_1 using a sequence of productions to replace strings by other strings

type 0 grammar: any phrase-structure grammar

type 1 grammar: a phrase-structure grammar in which every production is of the form $w_1 \to w_2$, where $w_1 = lAr$ and $w_2 = lwr$, where $A \in N$, $l, r, w \in (N \cup T)$ and $w = \lambda$, or

e) Define a type 3 grammar.

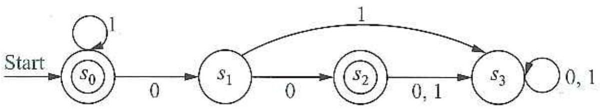
f) Give an example of a grammar that is not a type 3 grammar but is a type 2 grammar.

4. a) Define a regular grammar.
   b) Define a regular language.
   c) Show that the set $\{0^m 1^n \mid m, n = 0, 1, 2, \ldots\}$ is a regular language.

5. a) What is Backus–Naur form?
   b) Give an example of the Backus–Naur form of the grammar for a subset of English of your choice.

6. a) What is a finite-state machine?
   b) Show how a vending machine that accepts only quarters and dispenses a soft drink after 75 cents has been deposited can be modeled using a finite-state machine.

7. Find the set of strings recognized by the deterministic finite-state automaton shown here.



8. Construct a deterministic finite-state automaton that recognizes the set of bit strings that start with 1 and end with 1.

9. a) What is the Kleene closure of a set of strings?
   b) Find the Kleene closure of the set $\{11, 0\}$.

10. a) Define a finite-state automaton.
    b) What does it mean for a string to be recognized by a finite-state automaton?

11. a) Define a nondeterministic finite-state automaton.
    b) Show that given a nondeterministic finite-state automaton, there is a deterministic finite-state automaton that recognizes the same language.

12. a) Define the set of regular expressions over a set I.
    b) Explain how regular expressions are used to represent regular sets.

13. State Kleene's Theorem.

14. Show that a set is generated by a regular grammar if and only if it is a regular set.

15. Give an example of a set not recognized by a finite-state automaton. Show that no finite-state automaton recognizes it.

16. Define a Turing machine.

17. Describe how Turing machines are used to recognize sets.

18. Describe how Turing machines are used to compute number-theoretic functions.

19. What is an unsolvable decision problem? Give an example of such a problem.

Supplementary Exercises

*1. Find a phrase-structure grammar that generates each of these languages.
   a) the set of bit strings of the form $0^{2n}1^{3n}$, where n is a nonnegative integer
   b) the set of bit strings with twice as many 0s as 1s
   c) the set of bit strings of the form w^2, where w is a bit string

*2. Find a phrase-structure grammar that generates the set $\{0^{2^n} \mid n \geq 0\}$.

For Exercises 3 and 4, let $G = (V, T, S, P)$ be the context-free grammar with $V = \{(,), S, A, B\}$, $T = \{(,)\}$, starting symbol S, and productions $S \to A$, $A \to AB$, $A \to B$, $B \to (A)$, and $B \to ()$, $S \to \lambda$.

3. Construct the derivation trees of these strings.
   a) (())   b) ()(())   c) ((()()))

*4. Show that $L(G)$ is the set of all balanced strings of parentheses, defined in the preamble to Supplementary Exercise 55 in Chapter 4.

A context-free grammar is **ambiguous** if there is a word in $L(G)$ with two derivations that produce different derivation trees, considered as ordered, rooted trees.

5. Show that the grammar $G = (V, T, S, P)$ with $V = \{0, S\}$, $T = \{0\}$, starting state S, and productions $S \to 0S$, $S \to S0$, and $S \to 0$ is ambiguous by constructing two different derivation trees for 0^3.

6. Show that the grammar $G = (V, T, S, P)$ with $V = \{0, S\}$, $T = \{0\}$, starting state S, and productions $S \to 0S$ and $S \to 0$ is unambiguous.

7. Suppose that A and B are finite subsets of V^*, where V is an alphabet. Is it necessarily true that $|AB| = |BA|$?

8. Prove or disprove each of these statements for subsets A, B, and C of V^*, where V is an alphabet.
   a) $A(B \cup C) = AB \cup AC$
   b) $A(B \cap C) = AB \cap AC$
   c) $(AB)C = A(BC)$
   d) $(A \cup B)^* = A^* \cup B^*$

9. Suppose that A and B are subsets of V^*, where V is an alphabet. Does it follow that $A \subseteq B$ if $A^* \subseteq B^*$?

10. What set of strings with symbols in the set $\{0, 1, 2\}$ is represented by the regular expression $(\mathbf{2}^*)(\mathbf{0} \cup (\mathbf{12}^*))^*$?

The **star height** $h(\mathbf{E})$ of a regular expression over the set I is defined recursively by

$$h(\emptyset) = 0;$$
$$h(\mathbf{x}) = 0 \text{ if } \mathbf{x} \in I;$$
$$h((\mathbf{E}_1 \cup \mathbf{E}_2)) = h((\mathbf{E}_1\mathbf{E}_2)) = \max(h(\mathbf{E}_1), h(\mathbf{E}_2)) \text{ if } \mathbf{E}_1 \text{ and } \mathbf{E}_2 \text{ are regular expressions;}$$
$$h(\mathbf{E}^*) = h(\mathbf{E}) + 1 \text{ if } \mathbf{E} \text{ is a regular expression.}$$

11. Find the star height of each of these regular expressions.
    a) $\mathbf{0}^*\mathbf{1}$
    b) $\mathbf{0}^*\mathbf{1}^*$

c) $(0^*01)^*$
d) $((0^*1)^*)^*$
e) $(010^*)(1^*01^*)^*((01)^*(10)^*)^*$
f) $(((((0^*)1)^*0)^*)1)^*$

*12. For each of these regular expressions find a regular expression that represents the same language with minimum star height.

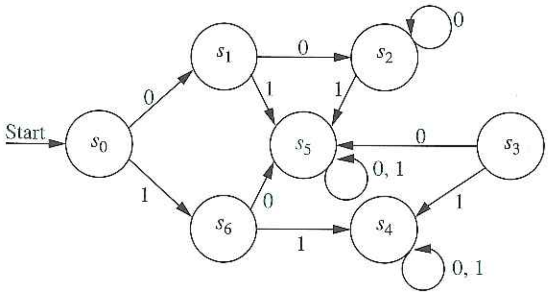
a) $(0^*1^*)^*$
b) $(0(01^*0)^*)^*$
c) $(0^* \cup (01)^* \cup 1^*)^*$

13. Construct a finite-state machine with output that produces an output of 1 if the bit string read so far as input contains four or more 1s. Then construct a deterministic finite-state automaton that recognizes this set.

14. Construct a finite-state machine with output that produces an output of 1 if the bit string read so far as input contains four or more consecutive 1s. Then construct a deterministic finite-state automaton that recognizes this set.

15. Construct a finite-state machine with output that produces an output of 1 if the bit string read so far as input ends with four or more consecutive 1s. Then construct a deterministic finite-state automaton that recognizes this set.

16. A state s' in a finite-state machine is said to be **reachable** from state s if there is an input string x such that $f(s, x) = s'$. A state s is called **transient** if there is no nonempty input string x with $f(s, x) = s$. A state s is called a **sink** if $f(s, x) = s$ for all input strings x. Answer these questions about the finite-state machine with the state diagram illustrated here.



a) Which states are reachable from s_0?
b) Which states are reachable from s_2?
c) Which states are transient?
d) Which states are sinks?

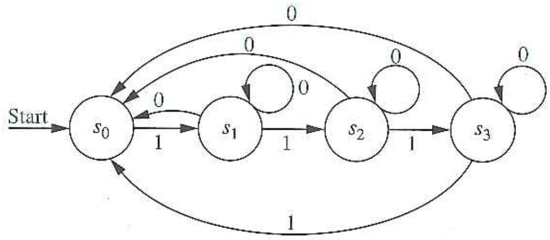
*17. Suppose that S, I, and O are finite sets such that $|S| = n$, $|I| = k$, and $|O| = m$.

a) How many different finite-state machines (Mealy machines) $M = (S, I, O, f, g, s_0)$ can be constructed, where the starting state s_0 can be arbitrarily chosen?
b) How many different Moore machines $M = (S, I, O, f, g, s_0)$ can be constructed, where the starting state s_0 can be arbitrarily chosen?

*18. Suppose that S and I are finite sets such that $|S| = n$ and $|I| = k$. How many different finite-state automata $M = (S, I, f, s_0, F)$ are there where the starting state s_0 and the subset F of S consisting of final states can be chosen arbitrarily

a) if the automata are deterministic?
b) if the automata may be nondeterministic? (*Note:* This includes deterministic automata.)

19. Construct a deterministic finite-state automaton that is equivalent to the nondeterministic automaton with the state diagram shown here.



20. What is the language recognized by the automaton in Exercise 19?

21. Construct finite-state automata that recognize these sets.

a) $0^*(10)^*$
b) $(01 \cup 111)^*10^*(0 \cup 1)$
c) $(001 \cup (11)^*)^*$

*22. Find regular expressions that represent the set of all strings of 0s and 1s

a) made up of blocks of even numbers of 1s interspersed with odd numbers of 0s.
b) with at least two consecutive 0s or three consecutive 1s.
c) with no three consecutive 0s or two consecutive 1s.

*23. Show that if A is a regular set, then so is $\overline{A}$.

*24. Show that if A and B are regular sets, then so is $A \cap B$.

*25. Find finite-state automata that recognize these sets of strings of 0s and 1s.

a) the set of all strings that start with no more than three consecutive 0s and contain at least two consecutive 1s.
b) the set of all strings with an even number of symbols that do not contain the pattern 101.
c) the set of all strings with at least three blocks of two or more 1s and at least two 0s.

*26. Show that $\{0^{2^n} \mid n \in \mathbf{N}\}$ is not regular. You may use the pumping lemma given in Exercise 22 of Section 12.4.

*27. Show that $\{1^p \mid p \text{ is prime}\}$ is not regular. You may use the pumping lemma given in Exercise 22 of Section 12.4.

*28. There is a result for context-free languages analogous to the pumping lemma for regular sets. Suppose that $L(G)$ is the language recognized by a context-free language G. This result states that there is a constant N such that if z

is a word in $L(G)$ with $l(w) \geq N$, then z can be written as $uvwxy$, where $l(vwx) \leq N$, $l(vx) \geq 1$, and $uv^i wx^i y$ belongs to $L(G)$ for $i = 0, 1, 2, 3, \ldots$. Use this result to show that there is no context-free grammar G with $L(G) = \{0^n 1^n 2^n \mid n = 0, 1, 2, \ldots\}$.

*29. Construct a Turing machine that computes the function $f(n_1, n_2) = \max(n_1, n_2)$.

*30. Construct a Turing machine that computes the function $f(n_1, n_2) = n_2 - n_1$ if $n_2 \geq n_1$ and $f(n_1, n_2) = 0$ if $n_2 < n_1$.

Computer Projects

Write programs with these input and output.

1. Given the productions in a phrase-structure grammar, determine which type of grammar this is in the Chomsky classification scheme.
2. Given the productions of a phrase-structure grammar, find all strings that are generated using twenty or fewer applications of its production rules.
3. Given the Backus–Naur form of type 2 grammar, find all strings that are generated using twenty of fewer applications of the rules defining it.

*4. Given the productions of a context-free grammar and a string, produce a derivation tree for this string if it is in the language generated by this grammar.

5. Given the state table of a Moore machine and an input string, produce the output string generated by the machine.
6. Given the state table of a Mealy machine and an input string, produce the output string generated by the machine.
7. Given the state table of a deterministic finite-state automaton and a string, decide whether this string is recognized by the automaton.
8. Given the state table of a nondeterministic finite-state automaton and a string, decide whether this string is recognized by the automaton.

*9. Given the state table of a nondeterministic finite-state automaton, construct the state table of a deterministic finite-state automaton that recognizes the same language.

**10. Given a regular expression, construct a nondeterministic finite-state automaton that recognizes the set that this expression represents.

11. Given a regular grammar, construct a finite-state automaton that recognizes the language generated by this grammar.
12. Given a finite-state automaton, construct a regular grammar that generates the language recognized by this automaton.

*13. Given a Turing machine, find the output string produced by a given input string.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Solve the busy beaver problem for two states by testing all possible Turing machines with two states and alphabet $\{1, B\}$.

*2. Solve the busy beaver problem for three states by testing all possible Turing machines with three states and alphabet $\{1, B\}$.

**3. Find a busy beaver machine with four states by testing all possible Turing machines with four states and alphabet $\{1, B\}$.

**4. Make as much progress as you can toward finding a busy beaver machine with five states.

**5. Make as much progress as you can toward finding a busy beaver machine with six states.

Writing Projects

Respond to these questions with essays using outside sources.

1. Describe how the growth of certain types of plants can be modeled using a Lidenmeyer system. Such a system uses a grammar with productions modeling the different ways plants can grow.
2. Describe the Backus–Naur form (and extended Backus–Naur form) rules used to specify the syntax of a programming language, such as Java, LISP, or ADA, or the database language SQL.