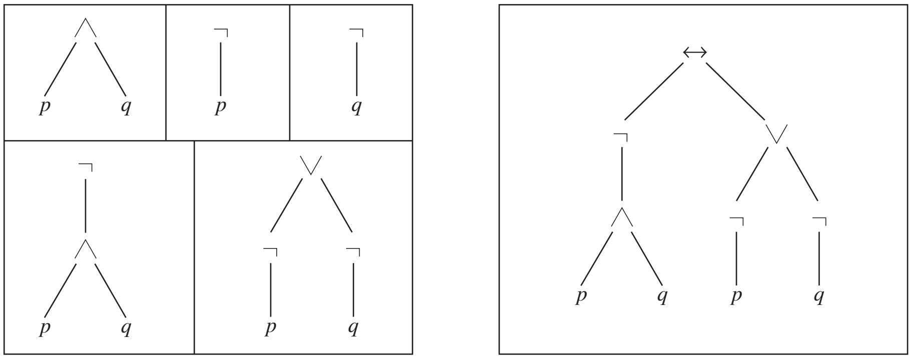


FIGURE 14 **Constructing the rooted tree for a compound proposition.**

Extra Examples

Solution: The rooted tree for this compound proposition is constructed from the bottom up. First, subtrees for $\neg p$ and $\neg q$ are formed (where $\neg$ is considered a unary operator). Also, a subtree for $p \wedge q$ is formed. Then subtrees for $\neg(p \wedge q)$ and $(\neg p) \vee (\neg q)$ are constructed. Finally, these two subtrees are used to form the final rooted tree. The steps of this procedure are shown in Figure 14.

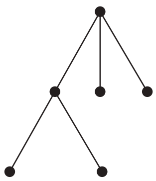
The prefix, postfix, and infix forms of this expression are found by traversing this rooted tree in preorder, postorder, and inorder (including parentheses), respectively. These traversals give $\leftrightarrow \neg \wedge pq \vee \neg p \neg q$, $pq \wedge \neg p \neg q \neg \vee \leftrightarrow$, and $(\neg(p \wedge q)) \leftrightarrow ((\neg p) \vee (\neg q))$, respectively. ◀

Because prefix and postfix expressions are unambiguous and because they can be evaluated easily without scanning back and forth, they are used extensively in computer science. Such expressions are especially useful in the construction of compilers.

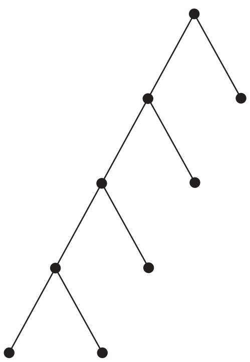
Exercises

In Exercises 1–3 construct the universal address system for the given ordered rooted tree. Then use this to order its vertices using the lexicographic order of their labels.

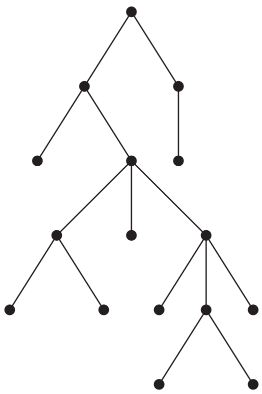
1.



2.



3.



4. Suppose that the address of the vertex v in the ordered rooted tree T is 3.4.5.2.4.

a) At what level is v?
b) What is the address of the parent of v?
c) What is the least number of siblings v can have?
d) What is the smallest possible number of vertices in T if v has this address?
e) Find the other addresses that must occur.

5. Suppose that the vertex with the largest address in an ordered rooted tree T has address 2.3.4.3.1. Is it possible to determine the number of vertices in T?

6. Can the leaves of an ordered rooted tree have the following list of universal addresses? If so, construct such an ordered rooted tree.

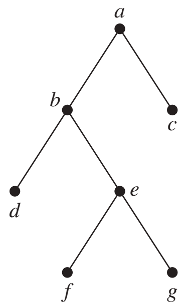
a) 1.1.1, 1.1.2, 1.2, 2.1.1.1, 2.1.2, 2.1.3, 2.2, 3.1.1, 3.1.2.1, 3.1.2.2, 3.2

b) 1.1, 1.2.1, 1.2.2, 1.2.3, 2.1, 2.2.1, 2.3.1, 2.3.2, 2.4.2.1, 2.4.2.2, 3.1, 3.2.1, 3.2.2

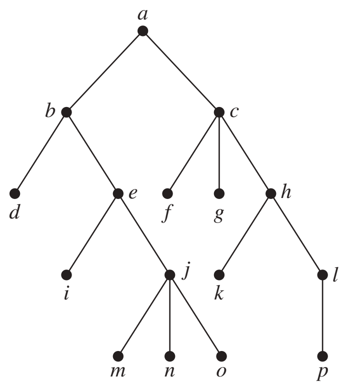
c) 1.1, 1.2.1, 1.2.2, 1.2.2.1, 1.3, 1.4, 2, 3.1, 3.2, 4.1.1.1

In Exercises 7–9 determine the order in which a preorder traversal visits the vertices of the given ordered rooted tree.

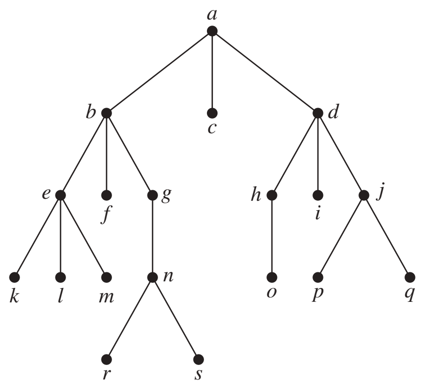
7.



8.



9.



10. In which order are the vertices of the ordered rooted tree in Exercise 7 visited using an inorder traversal?

11. In which order are the vertices of the ordered rooted tree in Exercise 8 visited using an inorder traversal?

12. In which order are the vertices of the ordered rooted tree in Exercise 9 visited using an inorder traversal?

13. In which order are the vertices of the ordered rooted tree in Exercise 7 visited using a postorder traversal?

14. In which order are the vertices of the ordered rooted tree in Exercise 8 visited using a postorder traversal?

15. In which order are the vertices of the ordered rooted tree in Exercise 9 visited using a postorder traversal?

16. a) Represent the expression $((x+2)\uparrow 3) * (y-(3+x)) - 5$ using a binary tree.

Write this expression in

b) prefix notation.

c) postfix notation.

d) infix notation.

17. a) Represent the expressions $(x+xy)+(x/y)$ and $x+((xy+x)/y)$ using binary trees.

Write these expressions in

b) prefix notation.

c) postfix notation.

d) infix notation.

18. a) Represent the compound propositions $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ and $(\neg p \wedge (q \leftrightarrow \neg p)) \vee \neg q$ using ordered rooted trees.

Write these expressions in

b) prefix notation.

c) postfix notation.

d) infix notation.

19. a) Represent $(A \cap B) - (A \cup (B - A))$ using an ordered rooted tree.

Write this expression in

b) prefix notation.

c) postfix notation.

d) infix notation.

***20.** In how many ways can the string $\neg p \wedge q \leftrightarrow \neg p \vee \neg q$ be fully parenthesized to yield an infix expression?

***21.** In how many ways can the string $A \cap B - A \cap B - A$ be fully parenthesized to yield an infix expression?

22. Draw the ordered rooted tree corresponding to each of these arithmetic expressions written in prefix notation. Then write each expression using infix notation.

a) $+ * + - 5\ 3\ 2\ 1\ 4$

b) $\uparrow + 2\ 3 - 5\ 1$

c) $* / 9\ 3 + * 2\ 4 - 7\ 6$

23. What is the value of each of these prefix expressions?

a) $- * 2 / 8\ 4\ 3$

b) $\uparrow - * 3\ 3 * 4\ 2\ 5$

c) $+ - \uparrow 3\ 2 \uparrow 2\ 3 / 6 - 4\ 2$

d) $* + 3 + 3 \uparrow 3 + 3\ 3\ 3$

24. What is the value of each of these postfix expressions?

a) $5\ 2\ 1 - - 3\ 1\ 4 + + *$

b) $9\ 3 / 5 + 7\ 2 - *$

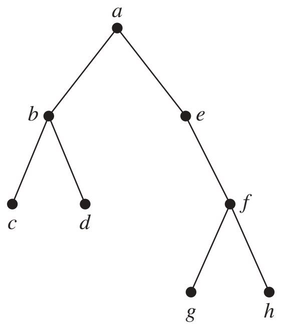
c) $3\ 2 * 2 \uparrow 5\ 3 - 8\ 4 / * -$

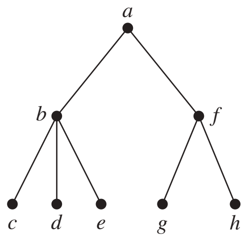
25. Construct the ordered rooted tree whose preorder traversal is *a, b, f, c, g, h, i, d, e, j, k, l*, where *a* has four children, *c* has three children, *j* has two children, *b* and *e* have one child each, and all other vertices are leaves.

***26.** Show that an ordered rooted tree is uniquely determined when a list of vertices generated by a preorder traversal of the tree and the number of children of each vertex are specified.

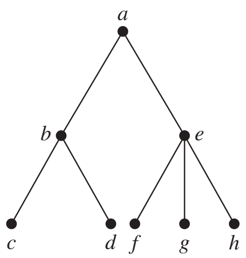
***27.** Show that an ordered rooted tree is uniquely determined when a list of vertices generated by a postorder traversal of the tree and the number of children of each vertex are specified.

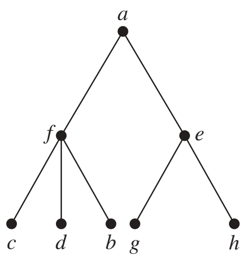
28. Show that preorder traversals of the two ordered rooted trees displayed below produce the same list of vertices. Note that this does not contradict the statement in Exercise 26, because the numbers of children of internal vertices in the two ordered rooted trees differ.





29. Show that postorder traversals of these two ordered rooted trees produce the same list of vertices. Note that this does not contradict the statement in Exercise 27, because the numbers of children of internal vertices in the two ordered rooted trees differ.





Well-formed formulae in prefix notation over a set of symbols and a set of binary operators are defined recursively by these rules:

(i) if x is a symbol, then x is a well-formed formula in prefix notation;

(ii) if X and Y are well-formed formulae and $*$ is an operator, then $* XY$ is a well-formed formula.

30. Which of these are well-formed formulae over the symbols $\{x, y, z\}$ and the set of binary operators $\{\times, +, \circ\}$?

a) $\times + + x\, y\, x$

b) $\circ\, x\, y \times x\, z$

c) $\times \circ\, x\, z \times \times\, x\, y$

d) $\times + \circ\, x\, x \circ x\, x\, x$

***31.** Show that any well-formed formula in prefix notation over a set of symbols and a set of binary operators contains exactly one more symbol than the number of operators.

32. Give a definition of well-formed formulae in postfix notation over a set of symbols and a set of binary operators.

33. Give six examples of well-formed formulae with three or more operators in postfix notation over the set of symbols $\{x, y, z\}$ and the set of operators $\{+, \times, \circ\}$.

34. Extend the definition of well-formed formulae in prefix notation to sets of symbols and operators where the operators may not be binary.

11.4 Spanning Trees

11.4.1 Introduction

Consider the system of roads in Maine represented by the simple graph shown in Figure 1(a). The only way the roads can be kept open in the winter is by frequently plowing them. The highway department wants to plow the fewest roads so that there will always be cleared roads connecting any two towns. How can this be done?

At least five roads must be plowed to ensure that there is a path between any two towns. Figure 1(b) shows one such set of roads. Note that the subgraph representing these roads is a tree, because it is connected and contains six vertices and five edges.

This problem was solved with a connected subgraph with the minimum number of edges containing all vertices of the original simple graph. Such a graph must be a tree.

Definition 1

Let G be a simple graph. A *spanning tree* of G is a subgraph of G that is a tree containing every vertex of G.