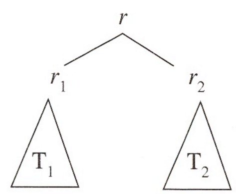
Basis: The basis consists of strictly binary trees of the form $(\{r\}, \emptyset, r)$. The equality clearly holds in this case since a tree of this form has one leaf and no arcs.

Inductive Hypothesis: Assume that every strictly binary tree T generated by n or fewer applications of the recursive step satisfies $2\,lv(\mathrm{T}) - 2 = arc(\mathrm{T})$.

Inductive Step: Let T be a strictly binary tree generated by $n+1$ applications of the recursive step in the definition of the family of strictly binary trees. T is built from a node r and two previously constructed strictly binary trees T_1 and T_2 with roots r_1 and r_2, respectively.



The node r is not a leaf since it has arcs to the roots of T_1 and T_2. Consequently, $lv(\mathrm{T}) = lv(\mathrm{T}_1) + lv(\mathrm{T}_2)$. The arcs of T consist of the arcs of the component trees plus the two arcs from r.

Since T_1 and T_2 are strictly binary trees generated by n or fewer applications of the recursive step, we may employ the inductive hypothesis to establish the desired equality. By the inductive hypothesis,

$$2\,lv(\mathrm{T}_1) - 2 = arc(\mathrm{T}_1)$$
$$2\,lv\,(\mathrm{T}_2) - 2 = arc(\mathrm{T}_2).$$

Now,

$$\begin{aligned} arc(\mathrm{T}) &= arc(\mathrm{T}_1) + arc(\mathrm{T}_2) + 2 \\ &= 2\,lv(\mathrm{T}_1) - 2 + 2\,lv(\mathrm{T}_2) - 2 + 2 \\ &= 2(lv(\mathrm{T}_1) + lv(\mathrm{T}_2)) - 2 \\ &= 2(lv(\mathrm{T})) - 2, \end{aligned}$$

as desired. □

Exercises

1. Let $\mathrm{X} = \{1, 2, 3, 4\}$ and $\mathrm{Y} = \{0, 2, 4, 6\}$. Explicitly define the sets described in parts (a) to (e).
   a) $\mathrm{X} \cup \mathrm{Y}$
   b) $\mathrm{X} \cap \mathrm{Y}$
   c) $\mathrm{X} - \mathrm{Y}$
   d) $\mathrm{Y} - \mathrm{X}$
   e) $\mathcal{P}(\mathrm{X})$

2. Let $\mathrm{X} = \{a, b, c\}$ and $\mathrm{Y} = \{1, 2\}$.
   a) List all the subsets of X.
   b) List the members of $\mathrm{X} \times \mathrm{Y}$.
   c) List all total functions from Y to X.
3. Let $\mathrm{X} = \{3^n \mid n > 0\}$ and $\mathrm{Y} = \{3n \mid n \geq 0\}$. Prove that $\mathrm{X} \subseteq \mathrm{Y}$.
4. Let $\mathrm{X} = \{n^3 + 3n^2 + 3n \mid n \geq 0\}$ and $\mathrm{Y} = \{n^3 - 1 \mid n > 0\}$. Prove that $\mathrm{X} = \mathrm{Y}$.

*5. Prove DeMorgan's Laws. Use the definition of set equality to establish the ident

6. Give functions $f : \mathbf{N} \to \mathbf{N}$ that satisfy the following.
   a) f is total and one-to-one but not onto.
   b) f is total and onto but not one-to-one.
   c) f is total, one-to-one, and onto but not the identity.
   d) f is not total but is onto.
7. Prove that the function $f : \mathbf{N} \to \mathbf{N}$ defined by $f(n) = n^2 + 1$ is one-to-one but no
8. Let $f : \mathbf{R}^+ \to \mathbf{R}^+$ be the function defined by $f(x) = 1/x$, where $\mathbf{R}^+$ denotes the positive real numbers. Prove that f is one-to-one and onto.
9. Give an example of a binary relation on $\mathbf{N} \times \mathbf{N}$ that is
   a) reflexive and symmetric but not transitive.
   b) reflexive and transitive but not symmetric.
   c) symmetric and transitive but not reflexive.
10. Let $\equiv$ be the binary relation on N defined by $n \equiv m$ if, and only if, $n = m$. Pro $\equiv$ is an equivalence relation. Describe the equivalence classes of $\equiv$.
11. Let $\equiv$ be the binary relation on N defined by $n \equiv m$ for all $n, m \in \mathbf{N}$. Prove th an equivalence relation. Describe the equivalence classes of $\equiv$.
12. Show that the binary relation LT, less than, is not an equivalence relation.
13. Let $\equiv_p$ be the binary relation on N defined by $n \equiv_p m$ if $n \bmod p = m \bmod$ $p \geq 2$, prove that $\equiv_p$ is an equivalence relation. Describe the equivalence cla $\equiv_p$.
14. Let $\mathrm{X}_1, \ldots, \mathrm{X}_n$ be a partition of a set X. Define an equivalence relation $\equiv$ on X equivalence classes are precisely the sets $\mathrm{X}_1, \ldots, \mathrm{X}_n$.
15. A binary relation $\equiv$ is defined on ordered pairs of natural numbers as fc $[m, n] \equiv [j, k]$ if, and only if, $m + k = n + j$. Prove that $\equiv$ is an equivalence r in $\mathbf{N} \times \mathbf{N}$.
16. Prove that the set of even natural numbers is denumerable.
17. Prove that the set of even integers is denumerable.

* **18.** Prove that the set of nonnegative rational numbers is denumerable.

19. Prove that the union of two disjoint countable sets is countable.

20. Prove that there are an uncountable number of total functions from **N** to {0, 1}.

21. A total function f from **N** to **N** is said to be *repeating* if $f(n) = f(n+1)$ for some $n \in \mathbf{N}$. Otherwise, f is said to be *nonrepeating*. Prove that there are an uncountable number of repeating functions. Also prove that there are an uncountable number of nonrepeating functions.

22. A total function f from **N** to **N** is *monotone increasing* if $f(n) < f(n+1)$ for all $n \in \mathbf{N}$. Prove that there are an uncountable number of monotone increasing functions.

23. Prove that there are uncountably many total functions from **N** to **N** that have a fixed point. See Example 1.4.3 for the definition of a fixed point.

24. A total function f from **N** to **N** is *nearly identity* if $f(n) = n - 1$, n, or $n + 1$ for every n. Prove that there are uncountably many nearly identity functions.

* **25.** Prove that the set of real numbers in the interval [0, 1] is uncountable. *Hint:* Use the diagonalization argument on the decimal expansion of real numbers. Be sure that each number is represented by only one infinite decimal expansion.

26. Let F be the set of total functions of the form $f : \{0, 1\} \to \mathbf{N}$ (functions that map from {0, 1} to the natural numbers). Is the set of such functions countable or uncountable? Prove your answer.

27. Prove that the binary relation on sets defined by $X \equiv Y$ if, and only if, $card(X) = card(Y)$ is an equivalence relation.

* **28.** Prove the Schröder-Bernstein Theorem.

29. Give a recursive definition of the relation *is equal to* on **N** × **N** using the operator s.

30. Give a recursive definition of the relation *greater than* on **N** × **N** using the successor operator s.

31. Give a recursive definition of the set of points $[m, n]$ that lie on the line $n = 3m$ in **N** × **N**. Use s as the operator in the definition.

32. Give a recursive definition of the set of points $[m, n]$ that lie on or under the line $n = 3m$ in **N** × **N**. Use s as the operator in the definition.

33. Give a recursive definition of the operation of multiplication of natural numbers using the operations s and addition.

34. Give a recursive definition of the predecessor operation

$$pred(n) = \begin{cases} 0 & \text{if } n = 0 \\ n - 1 & \text{otherwise} \end{cases}$$

using the operator s.

35. Subtraction on the set of natural numbers is defined by

$$n \dot{-} m = \begin{cases} n - m & \text{if } n > m \\ 0 & \text{otherwise.} \end{cases}$$

This operation is often called *proper subtraction*. Give a recursive definitio subtraction using the operations s and $pred$.

36. Let X be a finite set. Give a recursive definition of the set of subsets of X. as the operator in the definition.

* **37.** Give a recursive definition of the set of finite subsets of **N**. Use union and th s as the operators in the definition.

38. Prove that $2 + 5 + 8 + \cdots + (3n - 1) = n(3n + 1)/2$ for all $n > 0$.

39. Prove that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$ for all $n \geq 0$.

40. Prove $1 + 2^n < 3^n$ for all $n > 2$.

41. Prove that 3 is a factor of $n^3 - n + 3$ for all $n \geq 0$.

42. Let $P = \{A, B\}$ be a set consisting of two proposition letters (Boolean vari set E of well-formed conjunctive and disjunctive Boolean expressions over P recursively as follows:

i) Basis: $A, B \in E$.

ii) Recursive step: If $u, v \in E$, then $(u \vee v) \in E$ and $(u \wedge v) \in E$.

iii) Closure: An expression is in E only if it is obtained from the basis number of iterations of the recursive step.

a) Explicitly give the Boolean expressions in the sets E_0, E_1, and E_2.

b) Prove by mathematical induction that for every Boolean expression in E, t of occurrences of proposition letters is one more than the number of ope an expression u, let $n_p(u)$ denote the number of proposition letters in u denote the number of operators in u.

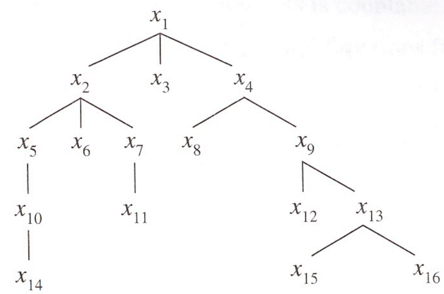
c) Prove by mathematical induction that, for every Boolean expression number of left parentheses is equal to the number of right parentheses.

43. Give a recursive definition of all the nodes in a directed graph that can be r paths from a given node x. Use the adjacency relation as the operation in the This definition also defines the set of descendants of a node in a tree.

44. Give a recursive definition of the set of ancestors of a node x in a tree.

45. List the members of the relation LEFTOF for the tree in Figure 1.6(a).

46. Using the tree below, give the values of each of the items in parts (a) to (e).



a) the depth of the tree

b) the ancestors of x_{11}

c) the minimal common ancestor of x_{14} and x_{11}, of x_{15} and x_{11}

d) the subtree generated by x_2

e) the frontier of the tree

47. Prove that a strictly binary tree with n leaves contains $2n - 1$ nodes.

48. A **complete binary tree** of depth n is a strictly binary tree in which every node on levels $1, 2, \ldots, n - 1$ is a parent and each node on level n is a leaf. Prove that a complete binary tree of depth n has $2^{n+1} - 1$ nodes.

Bibliographic Notes

The topics presented in this chapter are normally covered in a first course in discrete mathematics. A comprehensive presentation of the discrete mathematical structures important to the foundations of computer science can be found in Bobrow and Arbib [1974].

There are a number of classic books that provide detailed presentations of the topics introduced in this chapter. An introduction to set theory can be found in Halmos [1974], Stoll [1963], and Fraenkel, Bar-Hillel, and Levy [1984]. The latter begins with an excellent description of Russell's paradox and other antinomies arising in set theory. The diagonalization argument was originally presented by Cantor in 1874 and is reproduced in Cantor [1947]. The texts by Wilson [1985], Ore [1963], Bondy and Murty [1977], and Busacker and Saaty [1965] introduce the theory of graphs. Induction, recursion, and their relationship to theoretical computer science are covered in Wand [1980].

CHAPTER 2

Languages

The concept of language includes a variety of seemingly distinct categorie natural languages, computer languages, and mathematical languages. A genera of language must encompass all of these various types of languages. In this chapt set-theoretic definition of language is given: A language is a set of strings over a The alphabet is the set of symbols of the language and a string over the alphabe sequence of symbols from the alphabet.

Although strings are inherently simple structures, their importance in com and computation cannot be overemphasized. The sentence "The sun did not shine of English words. The alphabet of the English language is the set of words and p symbols that can occur in sentences. The mathematical equation

$$p = (n \times r \times t)/v$$

is a string consisting of variable names, operators, and parentheses. A digital ph stored as a bit string, a sequence of 0's and 1's. In fact, all data stored and man computers are represented as bit strings. As computer users, we frequently input i to the computer and receive output in the form of text strings. The source code of program is a text string made up of the keywords, identifiers, and special sy constitute the alphabet of the programming language. Because of the importanc we begin this chapter by formally defining the notion of string and studying the of operations on strings.

Languages of interest are not made up of arbitrary strings; not all strings words are sentences and not all strings of source code are legitimate compute Languages consist of strings that satisfy certain requirements and restrictions tha