Student Study Guide for

Discrete Structures, Logic, and Computability

Third Edition

James L. Hein

Portland State University

Preface

This study guide is written to accompany *Discrete Structures, Logic, and Computability*, Third Edition, by James L. Hein.

The study guide contains learning objectives, review questions, and a set of solved problems for each section of the book. Most of the learning objectives are statements of the form, "Be able to ... ." The review questions ask about the key ideas and notations from each section. The solutions to the problems, often with details included, are given at the end of each section for easy reference.

Some Notes On Learning The Material

1. Study each day. Read the text, and do problems, problems, and more problems. Problem solving skills are developed by practice. In other words, learn by doing.
2. Read ahead, so that your subconscious has plenty of time to process the symbols, definitions, ideas, and examples. Look at problems early too, so your subconscious can be working on them.
3. Include some time for review every day. Be sure you understand the meanings of the symbols and expressions in the symbol glossary and the definitions and results from the text. These items are the vocabulary for the textbook. Test your knowledge by using the vocabulary to restate ideas in your own words. You will also benefit by using the examples and problems given in the textbook as a guide to make up similar examples and problems of your own.
4. Don't cram. Problem solving skills can't be learned the night before an exam. On the other hand, if you have studied every day and solved a good number of the problems, then you should not need much preparation for exams.
5. Use the study guide as a supplement to the textbook. Be aware of the learning objectives for each section. After you have finished reading for the day, try to answer the appropriate review questions. Always try to solve a problem before looking at the solution.

I will be most grateful for suggestions or criticisms about the material in this study guide.

J. L. H.

Portland, Oregon

Contents

Chapter 1

Elementary Notions and Notations

1.1 A Proof Primer

Learning Objectives

Be able to describe the truth tables for simple logical statements.

Be able to use a variety of proof techniques to write short informal proofs about integers.

Review Questions

1. What is the converse of "A implies B"?

2. What does $d|n$ mean?

3. What is a prime number?

4. What does it mean for an integer to be even?

5. What does it mean for an integer to be odd?

6. What is proof by exhaustive checking?

7. What is conditional proof?

8. Why is proving the contrapositive important?

9. What is proof by contradiction?

10. What is an iff proof?

Solved Problems

1. Fill in the truth table for

   ((A implies B) and (B implies C) implies (A implies C)).

2. Prove that if x is even, then $x^2 + 3$ is odd.

3. Prove or give a counterexample for the following statement about the integers.

   If $d \mid m$ and d is odd, then m is odd.

4. Prove that if $x = 5m + 2$ and $y = 5n + 2$, where m and n are integers, then xy does not have the form $5k + 2$ for some integer k.

Solutions

1. All eight entries of the table are True.

2. If x is even, then it has the form $x = 2k$ for some integer k. Therefore,

$$x^2 + 3 = (2k)^2 + 3 = 4k^2 + 3 = 4k^2 + 2 + 1 = 2(2k^2 + 1) + 1,$$

which is in the form of an odd integer.

3. The statement is false. For a counterexample, let $d = 3$ and $m = 6$.

4. Assume that statement is false. Then there are integers m, n, and k such that

$$(5m + 2)(5n + 2) = 5k + 2.$$

Expand the left side to obtain the equation

$$25mn + 10m + 10n + 4 = 5k + 2.$$

Now collect terms, putting the variables on the left side to obtain the equation

$$25mn + 10m + 10n - 5k = -2.$$

Factor the left side of this equation to obtain

$$5(5mn + 2m + 2n - k) = -2.$$

This equation tells us that 5 divides –2, which is a contradiction. Therefore the given statement is true.

1.2 Sets

Learning Objectives

Be able to describe basic properties of sets and operations on sets.

Be able to describe characteristics of bags.

Be able to count finite sets using inclusion-exclusion.

Review Questions

1. What are the two characteristics of a set?

2. What is the meaning of each of the following symbols or expressions?

   a. $x \in S$?

   b. $x \notin S$?

c. $\varnothing$.

d. $\mathbb{N}$.

e. $\mathbb{Z}$.

f. $\mathbb{Q}$.

g. $\mathbb{R}$.

h. $\{x \mid P\}$.

i. $A \cup B$.

j. $A \cap B$.

k. $A - B$.

l. $A \oplus B$.

m. A'.

n. $|A|$.

3. How do you show $A \subseteq B$?

4. How do you show $A = B$?

5. What are the two characteristics of a bag or multiset?

6. Describe the union rule for counting sets.

7. Describe the difference rule for counting sets.

8. What is the inclusion-exclusion principle?

Solved Problems

1. Describe each set by listing each element.
   a. $\{x \mid x \in \mathbb{N} \text{ and } x \text{ divides } 12\}$.
   b. $\{x \mid x \in \mathbb{Z} \text{ and } x^3 < 60\}$.
2. Write down the power set for each set.
   a. $\{a\}$.
   b. $\{a, \{a\}\}$.
   c. $\{a, \varnothing, \{a, b\}\}$.
3. Find the smallest set S such that $\{\{a\}, \{b\}, \{\{a, b\}\}\} \subseteq \text{power}(S)$.
4. Write true or false for each of the following statements about sets.

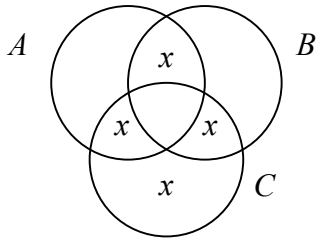
a. $A \cap (B \cup A) = A \cap B$.

b. $A \cup (B \cap C) = (A \cap B) \cup (A \cap C)$.

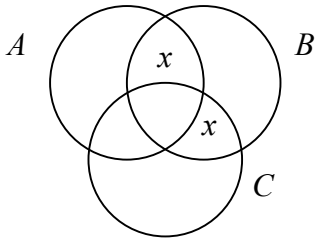
c. $A - (B \cap A) = A$.

5. Write down an expression to describe the set indicated by the x's in the following Venn diagram.

a.



b.



6. Given three sets A, B, and C. Suppose we know that the union of the three sets has cardinality 182. Further, $|A| = 92$, $|B| = 41$, $|C| = 118$. Also, $|A \cap B| = 15$, $|A \cap C| = 42$, and $|A \cap B \cap C| = 10$. Find $|B \cap C|$.

7. Evaluate each expression.

a. $\{a, b, c, d\} \oplus \{b, d, e, g\}$.

b. $[a, b, b, c, c, c] \cup [a, a, a, b, b, c]$.

c. $[a, b, b, c, c, c] \cap [a, a, a, b, b, c]$.

8. Prove the following statement about sets by letting x be an element of the left side and showing that x is an element of the right side.

$$((A \cap B) - C) \cup ((B \cap C) - A) \subseteq B \cap (A \cup C).$$

9. For each integer n let $A_n = \{x \mid x \in \mathbb{Z} \text{ and } 10n \leq x < 10(n + 1)\}$.

a. Find the intersection of the collection of sets $\{A_n \mid n \in \mathbb{N}\}$.

b. Find the union of the collection of sets $\{A_n \mid n \in \mathbb{N}\}$.

10. For any natural number n let $A_n = \{x \mid x \in \mathbb{N} \text{ and } n \leq x < 10n\}$. Describe each of the following sets.

a. $\bigcup_{n=0}^{\infty} A_n$.

b. $A_m - A_n$, where $m > n$.

c. $A_m - A_n$, where $m \leq n$.

11. Given the following two subsets of the rational numbers:

$$A = \{3n + 4 \mid n \in \mathbb{N}\} \quad \text{and} \quad B = \{3n + 1 \mid n \in \mathbb{N}\}.$$

Prove that $A \subseteq B$.

Solutions

1. a. $\{1, 2, 3, 4, 6, 12\}$.

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

2. a. $\{\varnothing, \{a\}\}$.

   b. $\{\varnothing, \{a\}, \{\{a\}\}, \{a, \{a\}\}\}$.

   c. $\{\varnothing, \{a\}, \{\varnothing\}, \{\{a, b\}\}, \{a, \varnothing\}, \{a, \{a, b\}\}, \{\varnothing, \{a, b\}\}, \{a, \varnothing, \{a, b\}\}\}$.

3. $S = \{a, b, \{a, b\}\}$

4. a. False. For example, let $A = \{a\}$ and $B = \{b\}$. With these choices, the equation becomes $\{a\} = \varnothing$, which is false.

   b. False. For example, let $A = \{a\}$ and $B = C = \varnothing$. With these choices, the equation becomes $\{a\} = \varnothing$, which is false.

   c. False. For example, let $A = B = \{a\}$. With these choices, the equation becomes $\varnothing = \{a\}$, which is false.

5. a. One of several answers is $(C \cup (A \cap B)) - (A \cap B \cap C)$.

   b. One of several answers is $(B \cap (A \cup C) - (A \cap B \cap C)$.

6. Use the inclusion-exclusion principle for three sets (1.12) and solve for the unknown to obtain $|B \cap C| = 22$.

7. a. $\{a, c, e, g\}$.

   b. $[a, a, a, b, b, c, c, c]$.

   c. $[a, b, b, c]$.

8. If x is an element of the left side, it follows that either $x \in (A \cap B) - C$ or $x \in (B \cap C) - A$. If $x \in (A \cap B) - C$, then $x \in A \cap B$ and $x \notin C$. So $x \in A$ and $x \in B$. Since $x \in A$, it follows that $x \in A \cup C$. Therefore x is an element of the right side. If $x \in (B \cap C) - A$, then a similar argument also puts x in the right side. Therefore the left side is a subset of the right side.

9. a. $\varnothing$.

   b. $\mathbb{Z}$. The answers follow because each set An consists of 10 consecutive digits starting at 10n. For example, $A_0 = \{0, 1, ..., 9\}$ and $A_1 = \{10, 11, ..., 10\}$.

10. a. $\mathbb{N}$.

    b. $\{x \mid x \in \mathbb{N} \text{ and } 10n \le x < 10m\}$.

    c. $\{x \mid x \in \mathbb{N} \text{ and } m \le x < n\}$.

11. Let $x \in A$. Then $x = 3n + 4$ for some natural number n. Now rewrite the expression as

$$x = 3n + 4 = 3n + 3 + 1 = 3(n + 1) + 1.$$

The latter expression represents a natural number since $n + 1$ is a natural number. So we have $A \subset B$. Notice that A is a proper subset of B because $1 \in B - A$.

1.3 Ordered Structures

Learning Objectives

Be able to describe basic properties of tuples, lists, strings, languages, and relations.

Be able to count tuples.

Review Questions

1. What are the two characteristics of a tuple?
2. What is the Cartesian product $A \times B$?
3. How do lists differ from tuples?
4. What is a string?
5. What is a language?
6. What does concatenation mean?
7. What is the product LM of two languages L and M?
8. If A is an alphabet, what is A^*?
9. If L is a language, what is L^n?
10. If L is a language, what is L^*?
11. If L is a language, what is L^+?
12. What is a relation?
13. What is a binary relation?

14. What is the product rule for counting tuples?

15. What is the meaning of each of the following symbols or expressions?

    a. ( ).

    b. $\langle\ \rangle$.

    c. Λ.

    d. cons(x, t).

    e. $a\ R\ b$.

    f. $R(a, b, c)$.

Solved Problems

1. Let $A = \{a, b\}$, $B = \{2, 3\}$, and $C = \{c\}$. Construct each of the following sets.
   a. $A \times B \times C$.
   b. $A \times (B \times C)$.
2. Evaluate each of the following expressions.
   a. $\{x \mid x \in \text{lists}(\{a, b\}) \text{ and length}(x) \leq 2\}$.
   b. $\{x \mid x \in \{a, b, c\}^* \text{ and } |x| = 2\}$.
   c. $|abba|$.
3. Write out the elements in the product $\{\Lambda, a, ba\}\{a, b\}$.
4. Solve the language equation for L.
   $$\{\Lambda, c, ab\}L = \{\Lambda, a, c, ab, bc, ca, aba, cbc, abbc\}.$$
5. Describe, in words, the strings in the following set.
   $$(\{a, b\}^* \cup \{b, c\}^*) - \{b\}^*.$$
6. Let L and M be two languages and let $x \in M^* \cup L$. Describe the general form of x by writing it as a concatenation of strings, where each string is in either L or M.
7. Let L and M be two languages. Let $x \in L \cup M$. Use only the definition of language product and the definition of set intersection to prove that
   $$x \in L^*M \cup LM^*.$$
8. Write out the tuples in the set $\{(x, y) \mid x, y \in \mathbb{N} \text{ and } 0 \leq x + y \leq 2\}$.
9. Write down the "greater than" relation over the set $\{1, 3, 5, 7, 9\}$.

10. The following two tables represent two relational databases for Farms and Harvests.

Farms

Name	*Crop*	*Acres*	*County*
Jones	corn	1500	Washington
Jones	barley	2500	Washington
Smith	wheat	600	Lincoln
Appleby	soybeans	2000	Washington
Nelson	corn	500	Jefferson
Nelson	soybeans	3500	Jefferson
Hein	hops	2000	Adams
Hein	grapes	200	Adams
Truman	corn	2500	Madison
Hill	wheat	3000	Lincoln

Harvests

Crop	*Month*
corn	July
barley	June
wheat	May
soybeans	June
grapes	September
hops	August

Answer each of the following questions by writing out the appropriate set of tuples.

a. What crops are harvested in June?

b. When is wheat harvested?

c. What is the list of acreages planted in corn?

d. What months does the Nelson farm harvest their crops?

e. What crops are planted in Washington county?

f. What counties plant corn?

g. What are the harvest months in Adams and Lincoln counties?

11. Find an expression for the number of strings of length 5 over the alphabet $\{a, b, c, d\}$ that contain at least one d.

12. Calculate the number of strings over the alphabet $\{a, b, c, d\}$ that have length 8 and such that the second letter in each string is either a or c, and in which each string contains at least one d.

Solutions

1. a. $A \times B \times C = \{(a, 2, c), (a, 3, c), (b, 2, c), (b, 3, c)\}$.

   b. $A \times (B \times C) = \{(a, (2, c)), (a, (3, c)), (b, (2, c)), (b, (3, c))\}$.

2. a. $\{\langle\,\rangle, \langle a\rangle, \langle b\rangle, \langle a, a\rangle, \langle a, b\rangle, \langle b, a\rangle, \langle b, b\rangle\}$.

   b. $\{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$.

   c. 4.

3. $\{a, b, aa, ab, baa, bab\}$.

4. $\{\Lambda, a, bc\}$.

5. The set consisting of any string over the alphabet $\{a, b\}$ that contains at least one a or any string over the alphabet $\{b, c\}$ that contains at least one c.

6. Either $x = \Lambda$, or $x = u_1 \ldots u_m$ where $u_i \in M$, or $x \in L$.

7. Let $x \in L \cap M$. Then $x \in L$ and $x \in M$. Since $\Lambda \in L^*$ and $\Lambda \in M^*$, it follows that $x = \Lambda x \in L^*M$ and $x = x\Lambda \in LM^*$. Therefore $x \in L^*M \cap LM^*$.

8. $\{(0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (2, 0)\}$.

9. $\{(9, 1), (9, 3), (9, 5), (9, 7), (7, 5), (7, 3), (7, 1), (5, 3), (5, 1), (3, 1)\}$.

10. a. {(barley), (soybeans)}.

    b. {(May)}.

    c. {(1500), (500), (2500)}.

    d. {(June), (July)}.

    e. {(corn), (barley), (soybeans)}.

    f. {(Washington), (Jefferson), (Madison)}.

    g. {(May), (August), (September)}.

11. $4^5 - 3^5$.

12. The answer can be found by taking the number of strings of length 8 whose second letter is a or c, which is $4 \cdot 2 \cdot 46 = 2 \cdot 47$, and subtracting the number of these strings that do not contain any d's, which is $2 \cdot 37$. This gives the answer $2 \cdot 47 - 2 \cdot 37$.

1.4 Graphs and Trees

Learning Objectives

Be able to describe basic properties of graphs and trees.

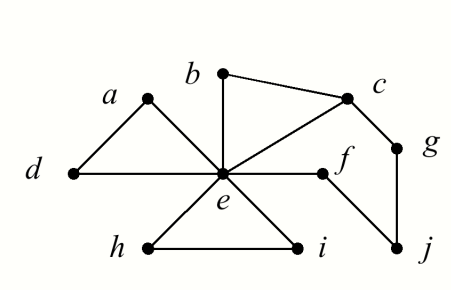
Be able to perform traversals of graphs and to construct spanning trees for graphs.

Review Questions

1. What is a graph?
2. What is a complete graph?
3. What is a planar graph?
4. What does it mean to color a graph?
5. What is a directed graph?
6. What is a path in a graph?
7. What is a depth-first traversal of a graph?
8. What is a breadth-first traversal of a graph?
9. What is a weighted graph?
10. What is a tree?
11. What is the level of a node?
12. What is a binary tree?
13. What is a binary search tree?
14. What is a spanning tree for a weighted graph?
15. Describe Prim's algorithm.

Solved Problems

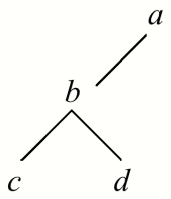
1. Draw a graph to represent the relation $R = \{(a, b), (a, c), (b, c), (c, a)\}$.
2. Given the following graph.



   a. Write down the vertices of the graph in the order that they are vis ited by some breadth-first search of the graph that starts at vertex *a*.

   b. Write down the vertices of the graph in the order that they are vis ited by some depth-first search of the graph that starts at vertex *a*.

3 Write down the tuple representation of the following binary tree.

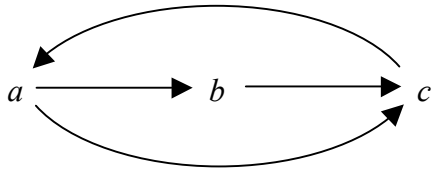


4. Draw the binary tree to represent the following algebraic expression.

$$2 + (3 – (4 – 5)).$$

Solutions

1. Each pair (x, y) is represented by an edge from x to y, so the graph has the following form.



2. a. For example, *a d e b c f h i g j*.

   b. For example, *a d e i h b c g j f*.

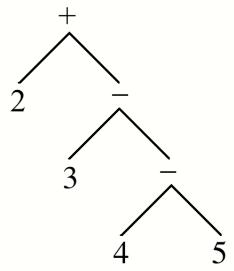
3. The tuple representation for the tree has the form $\langle L, \text{a}, \langle\,\rangle\rangle$, where L is the tuple representation of the left subtree of a. We can write L as the tuple

$$\langle\langle\langle\,\rangle, c, \langle\,\rangle\rangle, b, \langle\langle\,\rangle, d, \langle\,\rangle\rangle\rangle.$$

Therefore, the tuple representation of the given binary tree is

$$\langle L, a, \langle\,\rangle\rangle = \langle\langle\langle\langle\,\rangle, c, \langle\,\rangle\rangle, b, \langle\langle\,\rangle, d, \langle\,\rangle\rangle\rangle, a, \langle\,\rangle\rangle.$$

4. The binary tree to represent 2 + (3 – (4 – 5)) is



Chapter 2

Facts About Functions

2.1 Definitions and Examples

Learning Objectives

Be able to describe the parts of a function and to give examples of functions.

Be able to describe the floor, ceiling, gcd, and mod functions.

Be able to calculate values of functional expressions.

Review Questions

1. What are f, A, and B in the expression $f : A \to B$?

2. What is the arity of a function?

3. What is $f(C)$, the image of C under f?

4. What is the range of a function?

5. What is $f^{-1}(D)$, the pre-image of D under f?

6. What is a partial function?

7. What is the meaning of each of the following symbols or expressions?
   a. $\lfloor x \rfloor$.

   b. $\lceil x \rceil$.

   c. $\gcd(a, b)$.

   d. $a \bmod b$.

   e. $\log_b x$.

f. $\mathbb{N}_n$.

g. χ_B .

Solved Problems

1. Let $f : \mathbb{N} \to \mathbb{N}$ be defined by $f(x)$ = if x is odd then $x - 1$ else $x + 1$. Evaluate each of the following expressions:
   a. $f(\{0, 2, 4, 6, 8, 10\})$.
   b. Range(f).
   c. $f^{-1}(\{0\})$.
   d. $f^{-1}(\{1, 3, 5, 7, 9\})$.
2. Evaluate each of the following expressions:
   a. $\lfloor -5.1 \rfloor$.
   b. $\lfloor 7.9 \rfloor$.
   c. $\lceil -5.9 \rceil$.
   d. $\lceil 7.1 \rceil$.
3. Evaluate each of the following expressions:
   a. gcd(135, 210).
   b. gcd(117, 65).
4. Evaluate each of the following expressions:
   a. –23 mod 3.
   b. –23 mod 5.
   c. 28 mod 6.
   d. 77 mod 8.
5. Evaluate each of the following expressions:
   a. $\log_2(2048)$.
   b. $\log_2(8^{12})$.
   c. $\log_2(1/16)$.
6. Given the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(x) = 2x \bmod 5$, describe each of the following sets:
   a. The image of the set {1, 3, 5}. i.e., $f(\{1, 3, 5\})$.
   b. range(f). i.e., $f(\mathbb{N})$.
   c. The preimage of the set {0}. i.e., $f^{-1}(\{0\})$.

7. Given the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(x) = 3x \bmod 4$, describe each of the following sets:
   a. The image of $\{1, 3, 5\}$ under f. i.e., $f(\{1, 3, 5\})$.
   b. range(f). i.e., $f(\mathbb{N})$.
   c. The pre-image of $\{0\}$ under f. i.e., $f^{-1}(\{0\})$.

Solutions

1. a. $\{1, 3, 5, 7, 9, 11\}$. b. $\mathbb{N}$. c. $\{1\}$. d. $\{0, 2, 4, 6, 8, 10\}$.

2. a. –6. b. 7. c. –5. d. 8.

3. a. 15. b. 13.

4. a. 1. b. 2. c. 4. d. 5.

5. a. 11. b. 36. c. –4.

6. a. $\{0, 1, 2\}$. b. $\mathbb{N}_5 = \{0, 1, 2, 3, 4\}$. c. $\{5n \mid n \in \mathbb{N}\}$.

7. a. $\{1, 3\}$. b. $\mathbb{N}_4 = \{0, 1, 2, 3\}$. c. $\{4n \mid n \in \mathbb{N}\}$.

2.2 Constructing Functions

Learning Objectives

Be able to construct simple functions by composition of known functions.

Be able to use the map function.

Review Questions

1. What does $f \circ g$ mean?

2. What is the map function?

Solved Problems

1. Evaluate each of the following expressions.
   a. dist(3, seq(5)).
   b. map(f, $\langle a, b, c, d \rangle$).
   c. pairs(seq(2), seq(2)).

d. ceiling($\log_2(25)$).

e. floor($\log_2(19)$).

2. Describe the set of natural numbers x such that floor($\log_2(x)$) = 10.
3. Describe the set of natural numbers x such that ceiling($\log_2(x)$) = 10.
4. Let f be defined by $f(n, g) = \langle g(n), g(n-1), ..., g(0)\rangle$, where $n \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.
   a. Write down the type of f.
   b. Write f as a composition of known functions.
5. Describe the set of integers x that satisfy the following equation.

$$\gcd(x, x \bmod 5) = 3.$$

Solutions

1. a. $\langle(3, 0), (3, 1), (3, 2), (3,3), (3, 4), (3, 5)\rangle$.

   b. $\langle f(a), f(b), f(c), f(d)\rangle$.

   c. $\langle(0, 0), (1, 1), (2, 2)\rangle$.

   d. 5. e. 4.

2. $\{x \mid x \in \mathbb{N} \text{ and } 210 \le x < 211\}$.
3. $\{x \mid x \in \mathbb{N} \text{ and } 29 < x \le 210\}$.
4. a. $\mathbb{N} \times (\mathbb{N} \to \mathbb{N}) \to \text{lists}(\mathbb{N})$.

   b. Transform the definition as follows:

$$\begin{aligned} f(n, g) &= \langle g(n), g(n-1), ..., g(0)\rangle \\ &= \text{map}(g, \langle n, n-1, ..., 0\rangle) \\ &= \text{map}(g, \langle n-0, n-1, ..., n-n\rangle) \\ &= \text{map}(g, \text{map}(-, \langle(n, 0), (n, 1), ..., (n, n)\rangle)) \\ &= \text{map}(g, \text{map}(-, \text{dist}(n, \text{seq}(n)))). \end{aligned}$$

   Therefore, $f(n, g) = \text{map}(g, \text{map}(-, \text{dist}(n, \text{seq}(n))))$.

5. Since x mod 5 can take on only the values 0, 1, 2, 3, and 4, it follows that gcd(x, x mod 5) = 3 implies x mod 5 = 3. So x must have the form $3 + 5k$ for some integer k. So the equation becomes gcd($3 + 5k$, 3) = 3. This implies that $3 + 5k$ is divisible by 3. Thus $5k$ must be divisible by 3. Since 5 and 3 are relatively prime, it must be the case that k is divisible by 3. Therefore x has the form $x = 3 + 15n$ for any integer n.

2.3 Properties of Functions

Learning Objectives

Be able to determine whether simple functions are injective, surjective, or bijective.

Review Questions

1. What is an injective function?

2. What is a surjective function?

3. What is a bijective function?

4. What does f^{-1} mean?

5. What is the pigeonhole principle?

6. What is a hash function?

7. What properties of a function are useful in ciphers?

Solved Problems

1. In each case, find an example of a function $f : \mathbb{Z} \to \mathbb{Z}$ satisfying the given condition.
   a. Injective but not surjective.
   b. Surjective but not injective.
   c. Bijective but not the identity function.
   d. Neither injective nor bijective.
2. Let $\mathbb{Q}^+$ denote the positive rational numbers. Let $f : \mathbb{Q}^+ \to \mathbb{Q}^+$ be the function defined by
   $$f(x) = x/(2x + 1).$$
   a. Show that f is injective.
   b. Show that f is not surjective.
3. The function $f : \mathbb{N}_{15} \to \mathbb{N}_{15}$ defined by $f(x) = 4x \bmod 15$ is bijective. Find the inverse of f.
4. The function $f : \mathbb{N}_{15} \to \mathbb{N}_{15}$ defined by $f(x) = 2x \bmod 15$ is bijective. Find the inverse of f.

5. Let S = {Oregon, Washington, California, Idaho, Nevada, Hawaii} and let $h : S \to \{0, 1, 2, 3, 4, 5\}$ be a hash function defined by $h(x) = |x| \bmod 6$, where $|x|$ is the length of string x. In each case, fill in the hash table in the order that the states are listed in S.

   a. Resolve collisions with linear probing and a gap of 1.

   b. Resolve collisions with linear probing and a gap of 2.

6. Let $P = \{10, 12, 17, 19, 23, 25, 28\}$ and let $h: P \to \mathbb{N}_7$ be the hash function defined by

$$h(x) = x \bmod 7.$$

   Use h to place each element of P into a hash table indexed by {0, 1, .., 6} starting with 10, then 12, and so on up to 28.

   a. Resolve collisions by linear probing with a gap of 1.

   b. Resolve collisions by linear probing with a gap of 2.

7. Let $P = \{2, 3, 5, 7, 11, 13\}$ and let $h: P \to \mathbb{N}_6$ be the hash function defined by

$$h(x) = (2x + 1) \bmod 6.$$

   Use h to place each element of P into a hash table indexed by {0, 1, .., 5} starting with 2, then 3, and so on up to 13.

   a. Resolve collisions by linear probing with a gap of 1.

   b. Resolve collisions by linear probing with a gap of 3.

Solutions

1. a. e.g., $f(x) = 2x$. b. e.g., $f(x) =$ if $x < 0$ then x else $x - 1$.

   c. e.g., $f(x) = x + 1$. d. e.g., $f(x) = 0$ for all x.

2. a. Let $f(x) = f(y)$. Then $x/(2x + 1) = y/(2y + 1)$. Cross multiply to obtain the equation $2xy + x = 2xy + y$, which after subtracting $2xy$ from each side yields $x = y$. Therefore f is injective.

   b. Notice that $x/(2x + 1) < 1$ for all $x \in \mathbb{Q}^+$. Therefore f is not surjective. For example, $f(x) \neq 1$ for all $x \in \mathbb{Q}^+$.

3. f is bijective because gcd(4, 15) = 1. So by (2.6) $f^{-1}(x) = (kx + c) \bmod 15$, where $f(c) = 0$ and k is an integer such that $1 = 4k + 15m$ for some integer m. We can choose $c = 0$ since $f(0) = 0 \bmod 15 = 0$. Also, we can write $1 = 4 \cdot 4 + 15(-1)$. So we can let $k = 4$. Thus $f^{-1}(x) = 4x \bmod 15$. In other words, $f = f^{-1}$.

4. f is bijective because gcd(2, 15) = 1. So by (2.6) $f^{-1}(x) = (kx + c) \bmod 15$, where $f(c) = 0$ and k is an integer such that $1 = 2k + 15m$ for some integer m. We can choose $c = 0$ since $f(0) = 0 \bmod 15 = 0$. Also, we can write $1 = 2 \cdot 8 + 15(-1)$. So we can let $k = 8$. Thus $f^{-1}(x) = 8x \bmod 15$.

5. a. The table is represented by the following 6-tuple indexed from 0 to 5.

   $\langle$Oregon, Idaho, Nevada, Hawaii, Washington, California$\rangle$.

   b. The table is represented as the following 6-tuple indexed from 0 to 5, where $\varnothing$ means no entry.

   $\langle$Oregon, $\varnothing$, California, $\varnothing$, Washington, Idaho$\rangle$.

   In other words, Nevada and Hawaii cannot be placed in the table.

6. a. The table is represented by the following 7-tuple indexed from 0 to 6.

   $\langle 25, 28, 23, 10, 17, 12, 19\rangle$.

   b. The table is represented by the following 7-tuple indexed from 0 to 6.

   $\langle 17, 28, 19, 10, 23, 12, 25\rangle$.

7. a. The table is represented by the following 6-tuple indexed from 0 to 5.

   $\langle 5, 3, 11, 7, 13, 2\rangle$.

   b. The table is represented by the following 6-tuple indexed from 0 to 5, where $\varnothing$ means no entry.

   $\langle 13, 3, 5, 7, \varnothing, 2\rangle$.

   In other words, 11 cannot be placed in the table.

2.4 Countability

Learning Objectives

Be able to describe the concepts of countable and uncountable sets.

Be able to apply the diagonalization method to construct elements that are not in certain countable sets.

Review Questions

1. What does $|A| = |B|$ mean?

2. What does $|A| \leq |B|$ mean?

3. What does $|A| < |B|$ mean?

4. What is a countable set?

5. What does diagonalization technique do?

6. How many problems are solvable by computers?

Solved Problems

1. Find the cardinality of the set $\{x \mid x \in \mathbb{N} \text{ and } 0 \leq x^2 < 500\}$.
2. Find the cardinality of the set $\{5, 9, 13, ..., 145, 149\}$.
3. Show that the open intervals of real numbers (0, 1) and (0, 2) have the same cardinality.
4. Prove that power($\{a\}^*$) is uncountable.
5. Suppose that we have the countably infinite set $S = \{x_0, x_1, ...\}$ in which each x_i is an infinite list of the form $x_i = \langle x_{i0}, x_{i1}, ...\rangle$, where each $x_{ij} \in \{1, 3, 5, 7, 9\}$. Describe an element of this form that is not in S.

Solutions

1. Notice that $223 = 484$ and $233 = 529$. So the given set can be described as $\{0, 1, 2, ..., 22\}$, which has cardinality 23.
2. The given set can be described as $\{5 + 4x \mid x \in \mathbb{N} \text{ and } 0 \leq x \leq 36\}$, which can be described as $\{0, 1, 2, ..., 36\}$, which has cardinality 37.
3. Define $f : (0, 1) \rightarrow (0, 2)$ by $f(x) = 2x$. To see that f is an injection, let $f(x) = f(y)$. This means that $2x = 2y$, which implies that $x = y$, so f is an injection. To see that f is a surjection, let $y \in (0, 2)$. Then we have $f(y/2) = y$, so f is a surjection. Therefore f is a bijection.
4. The function $f : \{a\}^* \rightarrow \mathbb{N}$ defined by $f(a^n) = n$. f is a bijection. Therefore $|\{a\}^*| = |\mathbb{N}|$. This tells us that $\{a\}^*$ is countable and infinite. By (2.14) we have $|\{a\}^*| < |\text{power}(\{a\}^*)|$, so power($\{a\}^*$) must be uncountable.
5. Pick two elements, say 5 and 9, from the set $\{1, 3, 5, 7, 9\}$. By diagonalization (2.12) we can construct an infinite list $\langle y_0, y_1, ..., y_n, ... \rangle$ that is not in S where y_n = if $x_{nn} = 5$ then 9 else 5.

Chapter 3

Construction Techniques

3.1 Inductively Defined Sets

Learning Objectives

Be able construct inductive definitions for a variety of sets.

Review Questions

1. What steps are needed to define a set inductively?

2. Why is the closure case important?

3. What is the meaning of each of the following symbols or expressions?

   a. $x :: t$.

   b. tree(L, x, R).

Solved Problems

1. Describe the set S defined by: $0 \in S$, and if $x \in S$, then $x + 3 \in S$.
2. Write an inductive definition for $S = \{2, 6, 10, \ldots\}$.
3. Write an inductive definition for the set $\mathbb{Z}$ of integers.
4. Write an inductive definition for the set S of odd integers.
5. Write an inductive definition for $S = \{x \mid x \in \mathbb{Z} \text{ and } x \bmod 4 = 0\}$.
6. Write down an inductive definition for the set

$$B = \{1, 3, 5, 7, 9, \ldots\} \cup \{1, 4, 9, 16, 25, \ldots\}.$$

7. Describe the set S defined by: $a, b \in S$, and if $x \in S$, then $axb \in S$.
8. Give an inductive definition for the set $S = \{a^m b^{n+1} \mid m, n \in \mathbb{N}\}$.
9. Give an inductive definition for the set S of all palindromes over $\{a, b, c\}$ that have even length.
10. Describe the set S defined by: $\langle b \rangle \in S$, and if $x \in S$, then cons(a, x) $\in S$.
11. Write an inductive definition for the set S of all lists over $\{a, b\}$ of even length.

12. Write an inductive definition for the set

$$S = \{\langle b, a\rangle, \langle b, a, b\rangle, \langle b, a, b, b\rangle, \langle b, a, b, b, b\rangle, \ldots\}.$$

13. Write an inductive definition for the set B of nonempty binary trees over $\{a\}$ where the left and right subtrees of each node are identical.

14. Give an inductive definition of the set $\mathbb{N} \times \mathbb{Z}$.

15. Write down an inductive definition for the set $P = \text{power}(B)$ where B is a finite set.

Solutions

1. $S = \{0, 3, 6, 9, \ldots\} = \{3k \mid k \in \mathbb{N}\}$.

2. $2 \in S$; if $x \in S$, then $x + 4 \in S$.

3. $0 \in \mathbb{Z}$; if $x \in \mathbb{Z}$, then $x + 1, x - 1 \in \mathbb{Z}$.

4. $1 \in S$; if $x \in S$, then $x + 2, x - 2 \in S$.

5. $0 \in S$; if $x \in S$, then $x + 4, x - 4 \in S$.

6. To make things easier, write B as the union of two sets as follows:

$$B = \{1, 3, 5, 7, 9, \ldots\} \cup \{4, 16, 36, \ldots\}.$$

Now the following inductive definition is easier to discover.

$1, 4 \in B$;

if $x \in B$, then if x is odd, then $x + 2 \in B$ else $\left(\sqrt{x} + 2\right)^2 \in B$.

7. $S = \{a, b, aab, abb, aaabb, aabbb, \ldots\}$, which can be written formally as

$$S = \{a^n b^{n+1} \mid n \in \mathbb{N}\} \cup \{a^{n+1} b^n \mid n \in \mathbb{N}\}.$$

8. $b \in S$; if $x \in S$, then $ax, xb \in S$.

9. $\Lambda \in S$; if $x \in S$, then $axa, bxb, cxc \in S$.

10. $S = \{\langle b\rangle, \langle a, b\rangle, \langle a, a, b\rangle, \langle a, a, a, b\rangle, \ldots\}$.

11. $\langle\,\rangle \in S$; if $L \in S$, then for each pair $x, y \in \{a, b\}$, $x :: y :: L \in S$.

12. $\langle b, a\rangle \in S$; if $L \in S$, then $b :: a :: b :: \text{tail}(\text{tail}(L)) \in S$.

13. $\text{tree}(\langle\,\rangle, a, \langle\,\rangle) \in B$; if $T \in B$, then $\text{tree}(T, a, T) \in B$.

14. $(0, 0) \in \mathbb{N} \times \mathbb{Z}$; if $(x, y) \in \mathbb{N} \times \mathbb{Z}$, then $(x, y + 1), (x, y - 1), (x + 1, y) \in B$.

15. $\varnothing \in P$; if $S \in P$ and $b \in B$, then $S \cup \{b\} \in P$.

3.2 Recursive Functions and Procedures

Learning Objectives

Be able to construct recursive definitions for functions and procedures.

Review Questions

1. What steps are needed to define a function recursively?

2. What steps are needed to define a procedure recursively?

3. What do recursive definitions have to do with inductively defined sets?

Solved Problems

1. Write down each step in the evaluation of $f(13)$ where f has the following recursive definition.

   $f(0) = f(1) = 0$

   $f(n) = f(\text{ceiling}(n/2)) + n$.

2. Write a recursive definition for the function $f : \mathbb{N} \to \mathbb{N}$ defined by

   $$f(n) = 1 + 5 + 9 + ... + (4n + 1).$$

3. Write a recursive definition for the function f defined as follows, where n is a natural number and a is any number.

   $$f(a, n) = 1 + 2a + 3a^2 + ... + (n + 1)a^n.$$

4. Write a recursive definition for the function f to replace each occurrence of the letter a by b in a string over the alphabet $\{a, b, c\}$.

5. Write down each step in the evaluation of $f(3, \text{hello})$ where f has the following recursive definition.

   $f(0, x) = \langle\ \rangle$

   $f(n, x) = x :: f(n - 1, x)$.

6. Write a recursive definition for the function g on nonempty lists defined by

   $$g(\langle x_1, ..., x_n \rangle) = \langle x_1, ..., x_{n-1} \rangle.$$

7. Write a recursive definition for the function "get" that takes an element and a list of pairs and returns the list of pairs that begin with the given element. For example, $\text{get}(a, \langle (b, c), (a, c), (b, d), (a, b) \rangle) = \langle (a, c), (a, b) \rangle$.

8. Write down a recursive procedure to print out the elements of a list in reverse order.

9. Write down an recursive definition for the function "min" that returns the smallest number in a nonempty list of numbers.

10. For any binary tree T, the procedure $p(T)$ prints out only those nodes of T that have two children. Write a recursive procedure for $p(T)$ that traverses the tree in preorder looking for the nodes to print out.

Solutions

1. $f(13) = f(\text{ceiling}(13/2)) + 13$
   $= f(7) + 13$
   $= f(\text{ceiling}(7/2)) + 7 + 13$
   $= f(4) + 7 + 13$
   $= f(\text{ceiling}(4/2)) + 4 + 7 + 13$
   $= f(2) + 4 + 7 + 13$
   $= f(\text{ceiling}(2/2)) + 2 + 4 + 7 + 13$
   $= f(1) + 2 + 4 + 7 + 13$
   $= f(\text{ceiling}(1/2)) + 1 + 2 + 4 + 7 + 13$
   $= f(1) + 1 + 2 + 4 + 7 + 13$
   $= 0 + 1 + 2 + 4 + 7 + 13$
   $= 27.$
2. An equational definition can be given as follows:
   $f(0) = 1$
   $f(n) = f(n - 1) + 4n + 1.$
3. An equational definition can be given as follows:
   $f(a, 0) = 1$
   $f(a, n) = (n + 1)a^n + f(a, n - 1).$
4. An equational form for f can be described as follows:
   $f(\Lambda) = \Lambda$
   $f(ax) = bf(x)$
   $f(bx) = bf(x)$
   $f(cx) = cf(x)$
5. $f(3, \text{hello}) = \text{hello} :: f(2, \text{hello})$
   $= \text{hello} :: \text{hello} :: f(1, \text{hello})$
   $= \text{hello} :: \text{hello} :: \text{hello} :: f(0, \text{hello})$
   $= \text{hello} :: \text{hello} :: \text{hello} :: \langle\ \rangle$
   $= \text{hello} :: \text{hello} :: \langle\text{hello}\rangle$
   $= \text{hello} :: \langle\text{hello, hello}\rangle = \langle\text{hello, hello, hello}\rangle.$

6. An equational form for g can be described as follows:

   $g(\langle x \rangle) = \langle\ \rangle$

   $g(h :: t) = h :: g(t)$.

   An if-then-else form for g can be described as follows:

   $g(L)$ = if tail(L) = $\langle\ \rangle$ then $\langle\ \rangle$ else head(L) :: g(tail(L)).

7. An equational form for get can be described as follows:

   get(x, $\langle\ \rangle$) = $\langle\ \rangle$.

   get(x, $\langle x, z \rangle :: t$) = $\langle x, z \rangle$ :: get(x, t)

   get(x, $\langle y, z \rangle :: t$) = get($x$, t).

   An if-then-else form for get can be described as follows:

   get(x, L) = if L = $\langle\ \rangle$ then $\langle\ \rangle$
   else if x = head(head(L)) then
   head(L) :: get(x, tail(L))
   else
   get(x, tail(L)).

8. Letting rev be the name of the reverse procedure, we have the following definition:

   rev(L): if $L \neq \langle\ \rangle$ then rev(tail(L)); print(head(L)) fi.

9. An equational form for min can be described as follows:

   min($\langle x \rangle$) = x and min($h :: t$) = if h < min(t) then h else min(t).

10. P(T): if $T \neq \langle\ \rangle$ then
    if left(T) $\neq \langle\ \rangle$ and right(T) $\neq \langle\ \rangle$ then
    print(root(T)
    fi;
    P(left(T));
    P(right(T)
    fi.

3.3 Grammars

Learning Objectives

Be able to construct grammars for languages (sets of strings).

Be able to use grammars to derive strings.

Understand the idea of an ambiguous grammar.

Review Questions

1. What is a grammar production?

2. What is a grammar?

3. What is a derivation?

4. What is a leftmost derivation?

5. What is a rightmost derivation?

6. What is a parse tree?

7. What does it mean to say a grammar is ambiguous?

8. What is the language of a grammar?

9. What is the meaning of each of the following symbols or expressions?

   a. $A \rightarrow \alpha$.

   b. $A \rightarrow \alpha \mid \beta$.

   c. $A \Rightarrow \alpha$.

   d. $A \Rightarrow^{+} \alpha$.

   e. $A \Rightarrow^{*} \alpha$.

   f. $L(G)$.

10. What is a recursive production?

11. What is an indirectly recursive production?

Solved Problems

1. Given the following grammar.

$$S \to a \mid b \mid f(S, S) \mid g(S, S).$$

   a. Find a leftmost derivation for $f(g(a, b), f(b, a))$.

   b. Find a rightmost derivation for $f(g(a, b), f(b, a))$.

2. Describe the language generated by each of the following grammars.

   a. $S \to \Lambda \mid aaS$.

   b. $S \to \Lambda \mid aS \mid bS$.

   c. $S \to aS \mid bT$

      $T \to cdT \mid \Lambda$.

3. Define a grammar for each of the following languages.

   a. $\{a^{n+1}bc^n \mid n \in \mathbb{N}\}$.

   b. $\{a^{n+1}(bc)^n \mid n \in \mathbb{N}\}$.

   c. $\{a^n \mid n \in \mathbb{N}\}\{bc^n \mid n \in \mathbb{N}\}$.

   d. $\{a^n \mid n \in \mathbb{N}\} \cup \{bc^n \mid n \in \mathbb{N}\}$.

   e. $\{ab^n \mid n \geq 0\}^*$.

4. Show that each of the following grammars is ambiguous:

   a. $S \to SabS \mid \Lambda$.

   b. $S \to SbS \mid A$

      $A \to a \mid aA$.

5. Each of the following grammars is ambiguous. Try to find an equivalent grammar that is not ambiguous.

   a. $S \to SabS \mid \Lambda$.

   b. $S \to SbS \mid A$

      $A \to a \mid aA$.

Solutions

1. a. $S \Rightarrow f(S, S) \Rightarrow f(g(S, S), S) \Rightarrow f(g(a, S), S) \Rightarrow f(g(a, b), S)$

   $\Rightarrow f(g(a, b), f(S, S)) \Rightarrow f(g(a, b), f(b, S)) \Rightarrow f(g(a, b), f(b, a))$.

   b. $S \Rightarrow f(S, S) \Rightarrow f(S, f(S, S)) \Rightarrow f(S, f(S, a)) \Rightarrow f(S, f(b, a))$

   $\Rightarrow f(g(S, S), f(b, a)) \Rightarrow f(g(S, b), f(b, a)) \Rightarrow f(g(a, b), f(b, a))$.

2. a. The set of all strings of a's that have even length. In other words, the set $\{(aa)^m \mid m \in \mathbb{N}\}$.

   b. The set of all strings over the alphabet $\{a, b\}$. i.e., the set $\{a, b\}^*$.

   c. The set of all strings that start with any number of a's followed by b, which is followed by any number of cd pairs. In other words, the set $\{a^m b(cd)^n \mid m, n \in \mathbb{N}\}$.

3. a. Each string can be thought of as the string ab surrounded by an equal number of a's on the left and c's on the right. In other words, $a^n(ab)c^n$. This gives the following grammar:

   $$S \rightarrow aSc \mid ab.$$

   b. Just as in Part (a), each string has the form $a^n a(bc)^n$. This gives the following grammar:

   $$S \rightarrow aSbc \mid a.$$

   c. Use the product rule to obtain the following grammar:

   $S \rightarrow AB$

   $A \rightarrow aA \mid \Lambda$

   $B \rightarrow Bc \mid b.$

   d. Use the union rule to obtain the following grammar:

   $S \rightarrow A \mid B$

   $A \rightarrow aA \mid \Lambda$

   $B \rightarrow Bc \mid b.$

   e. Use the closure rule to obtain the following grammar:

   $S \rightarrow AS \mid \Lambda$

   $A \rightarrow Ab \mid a.$

4. a. The string abab has two distinct leftmost derivations as follows:

   $S \Rightarrow SabS \Rightarrow SabSabS \Rightarrow abSabS \Rightarrow ababS \Rightarrow abab.$

   $S \Rightarrow abS \Rightarrow abSabS \Rightarrow ababS \Rightarrow abab.$

   Therefore the grammar is ambiguous.

b. The string ababa has two distinct leftmost derivations as follows:

$S \Rightarrow SbS \Rightarrow SbSbS \Rightarrow AbSbS \Rightarrow abSbS \Rightarrow abAbS \Rightarrow ababS \Rightarrow ababA \Rightarrow ababa$.

$S \Rightarrow SbS \Rightarrow AbS \Rightarrow abS \Rightarrow abSbS \Rightarrow abAbS \Rightarrow ababS \Rightarrow ababA \Rightarrow ababa$.

Therefore the grammar is ambiguous.

5. a. The language of the grammar is $\{(ab)^n \mid n \in \mathbb{N}\}$. A grammar for the language can be written as follows:

$$S \rightarrow abS \mid \Lambda.$$

Since there is at most one nonterminal on the right side of each production, there can be only one leftmost derivation for each string. Therefore the grammar is not ambiguous.

b. The language of the grammar consists of strings that contain substrings of one or more *a*'s that are separated by the letter *b*. A grammar for the language can be written as follows:

$S \rightarrow AbS \mid A$

$A \rightarrow a \mid aA$.

Any leftmost derivation must start with the step $S \Rightarrow AbS$. Then A must be used to derive one or more *a*'s. After this, depending on whether there is another *b* in the derived string, one of the two productions $S \rightarrow AbS$ or $S \rightarrow A$ is used without ambiguity. The process continues in this way. Therefore there is only one leftmost derivation for any string. So the grammar is not ambiguous.

Chapter 4

Equivalence, Order, and Inductive Proof

4.1 Properties of Binary Relations

Learning Objectives

Be able to determine whether a binary relation is reflexive, symmetric, or transitive.

Be able to construct closures with respect to these properties.

Review Questions

1. What is the meaning of reflexive?

2. What is the meaning of symmetric?

3. What is the meaning of transitive?

4. What is the meaning of antisymmetric?

5. What is the composition of two binary relations R and S?

6. What is the meaning of each of the following symbols or expressions?

   a. $R \circ S$.

   b. $r(R)$.

   c. R^c.

   d. $s(R)$.

   e. $t(R)$.

f. R^+.

g. R^*.

7. What does Warshall's algorithm do?

8. What does Floyd's algorithm do?

9. What does Floyd's modified algorithm do?

Solved Problems

1. Given a nonempty set S and the binary relation R on power(S) defined by

   $A\ R\ B$ iff $A \cup B = S$.

   Which of the following properties does R possess: reflexive, symmetric, transitive, irreflexive, and antisymmetric?

2. Given the binary relation R on $\mathbb{Z}$ defined by $x\ R\ y$ iff $|x - y|$ is even. Which of the following properties does R possess: reflexive, symmetric, transitive, irreflexive, and antisymmetric?

3. Describe each of the following relations over the natural numbers.

   a. $= \circ <$.

   b. $> \circ <$.

4. Describe each of the following compositions:

   a. isMotherOf ∘ isFatherOf.

   b. isFatherOf ∘ isMotherOf.

   c. isChildOf ∘ isUncleOf.

   d. isAuntOf ∘ isParentOf.

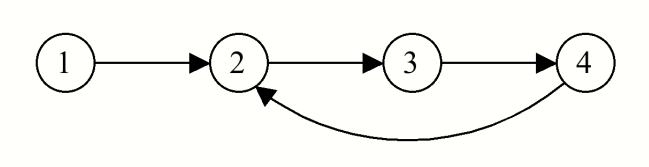
5. Give examples of binary relations over the set $\{a, b\}$ with the stated properties.

   a. Not reflexive.

   b. Not symmetric.

   c. Not transitive.

6. Given the binary relation $R = \{(1, 2), (2, 3), (3, 4), (4, 5)\}$, construct each of the following relations:
   a. R^2.
   b. R^3.
   c. R^4.
   d. R^5.
7. Given the binary relation $R = \{(1, 2), (3, 1), (3, 2), (2, 4)\}$ over $\{1, 2, 3, 4\}$, construct each of the following relations:
   a. $r(R)$.
   b. $s(R)$.
   c. $t(R)$.
8. Given the binary relation R represented as a directed graph.



   Construct each of the following relations:
   a. $r(R)$.
   b. $s(R)$.
   c. $t(R)$.

Solutions

1. R is not reflexive because, for example, $\varnothing \cup \varnothing \neq S$. R is symmetric be cause if $A \cup B = S$, then $B \cup A = S$. R is not transitive because, for example, we have $\varnothing \cup S = S$ and $S \cup \varnothing = S$, but $\varnothing \cup \varnothing \neq S$. R is not irreflexive because $S \cup S = S$. R is not antisymmetric because, for example, $\varnothing \cup S = S$ and $S \cup \varnothing = S$, but $S \neq \varnothing$.
2. R is reflexive because $|x - x| = 0$ is even for all integers x. R is symmetric because if $|x - y|$ is even, then $|y - x| = |x - y|$ is even. R is transitive because if $|x - y|$ is even and $|y - z|$ is even, then either x and y are both odd or both even and y and z are either both odd or both even. Thus x and z must be either both odd or both even, which tells us that $|x - z|$ is even. R is not irreflexive because it is reflexive. R is not antisymmetric because, for example, $3 \; R \; 5$ and $5 \; R \; 3$, but $3 \neq 5$.
3. a. The composition $= \circ <$ is equal to the $<$ relation. To see this, notice that

$$x \; (= \circ <) \; y \text{ iff } x = x \text{ and } x < y \text{ iff } x < y \; .$$

b. The composition $> \circ <$ is equal to the relation $(\mathbb{N} - \{0\}) \times (\mathbb{N} - \{0\})$. To see this, notice that $x (> \circ <) y$ iff $x > z$ and $z < y$ for some $z \in \mathbb{N}$. In other words, $x (> \circ <) y$ iff there is a number less than both x and y. Therefore, the pair (x, y) is not in the composition $> \circ <$ exactly when either $x = 0$ or $y = 0$.

4. a. isPaternalGrandMotherOf. b. isMaternalGrandFatherOf.
   c. isFirstCousinOf. d. isGreatAuntOf.
5. a. Each of the three relations $\varnothing$, $\{(a, a)\}$, and $\{(b, b)\}$ is not reflexive.
   b. Each of the relations $\{(a, b)\}$ and $\{(b, a)\}$ is not symmetric.
   c. The relation $\{(a, b), (b, a)\}$ is not transitive.
6. a. $R^2 = \{(1, 3), (2, 4), (3, 5)\}$. b. $R^3 = \{(1, 4), (2, 5)\}$. c. $R^4 = \{(1, 5)\}$. d. $R^5 = \varnothing$.
7. a. $R \cup \{(1, 1), (2, 2), (3, 3), (4, 4)\}$. b. $R \cup \{(2, 1), (1, 3), (2, 3), (4, 2)\}$.
   c. $R \cup \{(1, 4), (3, 4)\}$.
8. a. $R \cup \{(1, 1), (2, 2), (3, 3), (4, 4)\}$. b. $R \cup \{(2, 1), (3, 2), (4, 3), (2, 4)\}$.
   c. $R \cup \{(1, 3), (1, 4), (2, 2), (2, 4), (3, 2), (3, 3), (4, 3), (4, 4)\}$.

4.2 Equivalence Relations

Learning Objectives

Be able to identify equivalence relations.

Be able to construct equivalence classes for equivalence relations.

Review Questions

1. What are the defining characteristics of an equivalence relation?
2. What is an equivalence class?
3. What is a partition?
4. What does $[x]$ mean?
5. What does $tsr(R)$ mean?
6. What is a kernel relation?
7. What does it mean to say one partition is finer than another?
8. Describe Kruskal's algorithm.

Solved Problems

1. Find out whether each relation over $\mathbb{N}$ is an equivalence relation.
   a. $a\ R\ b$ iff $(a - b) \bmod 4 = 0$.
   b. $a\ R\ b$ iff either $a \leq b$ or $a > b$.
   c. $a\ R\ b$ iff $|a - b| \leq 2$.
2. Write down the equivalence classes induced by the following equivalence relation over the set $\mathbb{N}$ of natural numbers.

$$a\ R\ b \text{ iff } (a - b) \bmod 4 = 0.$$

3. Given the relation R on nonempty lists over $\mathbb{N}$ defined by

$$x\ R\ y \text{ iff } \text{head}(x) \bmod 3 = \text{head}(y) \bmod 3.$$

   a. Why is R is an equivalence relation?
   b. Describe the equivalence classes in partition of nonempty lists over $\mathbb{N}$ induced by R.
4. Let $f : \mathbb{N} \to \mathbb{N}$ be defined by $f(n) = \text{floor}(\sqrt{n})$. Describe the equivalence classes in the partition of $\mathbb{N}$ induced by the kernel relation of f.
5. Find $tsr(R)$, where R is the relation defined over $\mathbb{Z}$ by

$$R = \{(a, b) \mid \text{either } a \text{ or } b \text{ is an odd integer}\}.$$

6. A graph with vertex set $\{a, b, c, d, e, f, g\}$ has its edges sorted by weight as follows:

$$\langle \{f, g\}, \{a, d\}, \{c, e\}, \{b, f\}, \{d, c\}, \{c, g\}, \{e, f\}, \{a, b\}, \{d, e\} \rangle.$$

   Use Kruskal's algorithm to find a minimal spanning tree for the graph by showing the value of the spanning tree T and the corresponding equivalence classes produced at each stage of the algorithm.

Solutions

1. a. R is an equivalence relation. It is reflexive because $(a - a) \bmod 4 = 0$ for all $a \in \mathbb{N}$. It is symmetric because if $(a - b) \bmod 4 = 0$, then certainly $(b - a) \bmod 4 = 0$. To see that R is transitive, let $(a - b) \bmod 4 = 0$ and $(b - c) \bmod 4 = 0$. Then there are integers m and n such that $a - b = 4m$ and $b - c = 4n$. Add the two equations to obtain $a - c = 4(m + n)$. Therefore $(a - c) \bmod 4 = 0$.
   b. To see that R is an equivalence relation, notice that $a\ R\ b$ means that either $a \leq b$ or $a > b$. This can be written as either $a = b$ or $a < b$ or $a > b$. We know this property is true for any pair of numbers. Therefore $a\ R\ b$ for all $a, b \in \mathbb{N}$, so R is the Cartesian product $\mathbb{N} \times \mathbb{N}$.
   c. R is not an equivalence relation. It is reflexive and symmetric, but it is not transitive. For example, we have $|1 - 3| \leq 2$ and $|3 - 5| \leq 2$, but $|1 - 5| > 2$.

2. $[0] = \{4n \mid n \in \mathbb{N}\}$

   $[1] = \{4n + 1 \mid n \in \mathbb{N}\}$

   $[2] = \{4n + 2 \mid n \in \mathbb{N}\}$

   $[3] = \{4n + 3 \mid n \in \mathbb{N}\}$.

3. a. R is an equivalence relation because it is the kernel relation of the function f : lists($\mathbb{N}$) – $\{\langle\,\rangle\} \rightarrow \mathbb{N}$ defined by $f(x :: t) = x \bmod 3$.

   b. The three equivalence classes can be described as follows:

   $[\langle 0\rangle] = \{h :: t \mid h \bmod 3 = 0\}$,

   $[\langle 1\rangle] = \{h :: t \mid h \bmod 3 = 1\}$,

   $[\langle 2\rangle] = \{h :: t \mid h \bmod 3 = 2\}$.

4. Let $\sim$ be the kernel relation of f. Then $x \sim y$ iff $f(x) = f(y)$ iff floor($\sqrt{x}$) = floor($\sqrt{y}$). For example, $2 \sim 3$ and $5 \sim 8$. The equivalence classes can be written in the following form for each natural number n:

$$[n^2] = \{x \mid n^2 \leq x < (n + 1)^2\}$$

   e.g., $[0] = \{0\}$, $[1] = \{1, 2, 3\}$, $[4] = \{4, 5, 6, 7, 8\}$, and $[9] = \{9, 10, ..., 15\}$.

5. Let O be the set of odd integers. Then $R = (\mathbb{Z} \times O) \cup (O \times \mathbb{Z})$. To construct $r(R)$ we must add the set $\{(2x, 2x) \mid x \in \mathbb{Z}\}$. It follows that $r(R) = R \cup \{(2x, 2x) \mid x \in \mathbb{Z}\}$. This relation is already symmetric, so we have $sr(R) = r(R)$. This relation is not transitive. For example, we have (2, 1) and (1, 4) $\in sr(R)$, but (2, 4) $\notin sr(R)$. Notice that for any $x, y \in \mathbb{Z}$ we have $(x, 1), (1, y) \in sr(R)$. Thus $(x, y) \in tsr(R)$. Therefore $tsr(R) = \mathbb{Z} \times \mathbb{Z}$.

6.

Spanning Tree T	Equivalence Classes
$\{\ \}$	$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}$
$\{\{f, g\}\}$	$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f, g\}$
$\{\{f, g\}, \{a, d\}\}$	$\{a, d\}, \{b\}, \{c\}, \{e\}, \{f, g\}$
$\{\{f, g\}, \{a, d\}, \{c, e\}\}$	$\{a, d\}, \{b\}, \{c, e\}, \{f, g\}$
$\{\{f, g\}, \{a, d\}, \{c, e\}, \{b, f\}\}$	$\{a, d\}, \{c, e\}, \{b, f, g\}$
$\{\{f, g\}, \{a, d\}, \{c, e\}, \{b, f\}, \{d, c\}\}$	$\{a, c, d, e\}, \{b, f, g\}$
$\{\{f, g\}, \{a, d\}, \{c, e\}, \{b, f\}, \{d, c\}, \{c, g\}\}$	$\{a, b, c, d, e, f, g\}$

4.3 Order Relations

Learning Objectives

Be able to identify a partially ordered set.

Be able to construct a topological sort of a partially ordered set.

Be able to determine whether a partially ordered set is well-founded.

Review Questions

1. What are the two characteristics of a partial order relation?

2. Why do we use the word partial when referring to an order?

3. What do successor and predecessor mean for a poset?

4. What does it mean to sort a poset "topologically"?

5. What is a lower bound of a subset of a poset? Upper bound?

6. What is a minimal element of a subset of a poset? Maximal element?

7. What is a greatest lower bound of a subset of a poset? Least upper bound?

8. What are two equivalent ways to say a poset is well-founded?

9. What is the meaning of each of the following symbols or expressions?

   a. $\langle A, \preceq \rangle$.

   b. $\langle A, \prec \rangle$.

   c. $x \prec y$.

   d. $x \preceq y$.

Solved Problems

1. Let $D = \{1, 2, 4, 5, 10, 20, 25, 50, 100\}$ and for any $x, y \in D$, let

   $$x \prec y \text{ mean that } x|y \text{ (i.e., } x \text{ divides } y).$$

   Let $S = \{10, 20, 50\}$. Find each of the following elements or sets.

a. The minimal elements of S.

b. The maximal elements of S.

c. The lower bounds of S.

d. The upper bounds of S.

e. The glb of S.

f. The lub of S.

2. Given the following set W of western states:

   W = {Oregon, Idaho, Nevada, California, Washington, Utah, Arizona}.

   We can define a partial order $\prec$ on W by

   $$x \prec y \text{ iff } \text{length}(x) < \text{length}(y).$$

   Let S = {Nevada, Oregon, Arizona}. Find each of the following elements or sets.

   a. The minimal elements of S.

   b. The maximal elements of S.

   c. The lower bounds of S.

   d. The upper bounds of S.

   e. The glb of S.

   f. The lub of S.

3. Given the poset power($\{a, b, c\}$) with the subset relation, write down a topological sort of the subsets.

4. Describe a well-founded set that has two or more minimal elements.

5. Given the relation $\prec$ defined on $\mathbb{N} \times \mathbb{N}$ by

   $$(a, b) \prec (c, d) \text{ iff } a + b < c + d.$$

   Write down a descending chain of maximum length starting at (2, 5).

Solutions

1. a. 10. b. 20 and 50. c. 1, 2, 5, and 10. d. 100. e. 10. f. 100.

2. a. Nevada and Oregon. b. Arizona. c. Utah and Idaho.

   d. Arizona, California, and Washington. e. Idaho. f. Arizona.

3. One such sort is $\varnothing$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a, b, c\}$.

4. For example, the collection of nonempty subsets of $\{a, b\}$ with the subset relation is a well-founded set with minimal elements $\{a\}$ and $\{b\}$.

5. (2, 5), (2, 4), (2, 3), (2, 2), (2, 1), (2, 0), (1, 0), (0, 0).

4.4 Inductive Proof

Learning Objectives

Be able to use the technique of inductive proof to write short informal proofs about simple properties of numbers, sets, and ordered structures.

Review Questions

1. What is proof by the principle of mathematical induction?

2. What is proof by well-founded induction?

3. What is proof by the second principle of mathematical induction?

Solved Problems

1. Write an induction proof to show that the following equation holds for all $n \in \mathbb{N}$.

$$1 + 2 + 2^2 + \ldots + 2^n = 2^{n+1} - 1.$$

2. Write an inductive proof to show that the following statement is true for all natural numbers n.

$$2 + 4 + \ldots + (2n) = n^2 + n.$$

3. Write an inductive proof to show that the following statement is true for all natural numbers $n \geq 1$.

$$2 + 2 \cdot 3^2 + 3 \cdot 3^3 + 4 \cdot 3^4 + \cdots + n \cdot 3^n = \frac{3^{n+1}(2n-1)+3}{4}.$$

4. Given the following program:

$$f(n) = \text{if } n = 0 \text{ then } 1 \text{ else } (2n + 1) + f(n - 1).$$

Use inductive proof to show that $f(n) = (n + 1)^2$ for all $n \in \mathbb{N}$.

5. Given the following recursive definition for the function f:

$$f(A, B) = \text{if } A = \langle\,\rangle \text{ then } B \text{ else } f(\text{tail}(A), \text{head}(A) :: B).$$

The fact that $f(A, \langle\,\rangle)$ is the reverse of A follows from the more general statement that "$f(A, B) = \text{cat}(\text{reverse}(A), B)$". Give an inductive proof to show that this statement is true for all lists A and B.

Solutions

1. If $n = 0$, the equation becomes $1 = 2 - 1$, which is true. Assume that the equation is true for n and prove that it is true for $n + 1$. Starting with the left side of the equation for $n + 1$, we have

$$\begin{aligned} 1 + 2 + \cdots + 2^n + 2^{n+1} &= (1 + 2 + \cdots + 2^n) + 2^{n+1} \\ &= 2^{n+1} - 1 + 2^{n+1} \\ &= 2^{n+1+1} - 1. \end{aligned}$$

The last term is the right side of the desired equation for $n + 1$. Therefore the original equation is true for all $n \in \mathbb{N}$.

2. If $n = 0$, the equation becomes $0 = 0 + 0$, which is true. Assume that the equation is true for n and prove that it is true for $n + 1$. Starting with the left side of the equation for $n + 1$, we have

$$\begin{aligned} 2 + 4 + \cdots + 2n + 2(n+1) &= (2 + 4 + \cdots + 2n) + 2(n+1) \\ &= (n^2 + n) + 2(n+1) \\ &= n^2 + 3n + 2 \\ &= (n^2 + 2n + 1) + (n+1) \\ &= (n+1)^2 + (n+1). \end{aligned}$$

The last term is the right side of the desired equation for $n + 1$. Therefore the original equation is true for all $n \in \mathbb{N}$.

3. If $n = 1$, the equation becomes $3 = (3^2 \cdot 1 + 3)/4$, which is true. Assume that the equation is true for n and prove that it is true for $n + 1$. Starting with the left side of the equation for $n + 1$, we have

$$\begin{aligned} (3 + 2 \cdot 3^2 + 3 \cdot 3^3 + 4 \cdot 3^4 + \cdots + n3^n) + (n+1)3^{n+1} &= \frac{3^{n+1}(2n-1) + 3}{4} + (n+1)3^{n+1} \\ &= \frac{3^{n+1}(2n-1) + 3 + 4(n+1)3^{n+1}}{4} \\ &= \frac{3^{n+1}(6n+3) + 3}{4} \\ &= \frac{3^{n+1+1}(2n+1) + 3}{4} \\ &= \frac{3^{n+1+1}(2(n+1)-1) + 3}{4}. \end{aligned}$$

The last term is the right side of the desired equation for $n + 1$. Therefore the original equation is true for all natural numbers $n \geq 1$.

4. If $n = 0$, $f(0) = 1 = (0 + 1)^2$, which is true. Assume that $f(n) = (n + 1)^2$ and show that $f(n + 1) = (n + 1 + 1)^2$. Starting with the definition of f, we obtain

$$\begin{aligned} f(n+1) &= (2(n+1)+1) + f((n+1)-1) \\ &= 2n + 3 + f(n) \\ &= 2n + 3 + (n+1)^2 \\ &= n^2 + 4n + 4 \\ &= (n+2)^2 = (n+1+1)^2. \end{aligned}$$

   Therefore, $f(n) = (n + 1)^2$ for all $n \in \mathbb{N}$.

5. If we compare lists by their length, then they form a well-founded set with $\langle\ \rangle$ as the minimal element. Notice that the definition of f uses its first argument as the active member, so we'll prove the statement by induction on the first argument of f. If $A = \langle\ \rangle$, then the definition of f gives $f(\langle\ \rangle, B) = B$ for any list B. We know that reverse($\langle\ \rangle$)= $\langle\ \rangle$ and cat($\langle\ \rangle$, B) = B, so we have $f(\langle\ \rangle, B)$ = cat(reverse($\langle\ \rangle$), B). Now let A be a nonempty list and assume that $f(X, B)$ = cat(reverse(X), B) for all lists X with fewer elements than A and all lists B. Then the definition of f gives $f(A, B) = f$(tail(A), head(A) :: B). Since tail(A) has fewer elements than A, the induction assumption implies that

$$f(\text{tail}(A), \text{head}(A) :: B) = \text{cat}(\text{reverse}(\text{tail}(A)), \text{head}(A) :: B).$$

   From our knowledge of reverse and cat, it follows that a list can be reversed by reversing the tail and concatenating the result with the list consisting of the head. In other words, we have

$$\text{reverse}(A) = \text{cat}(\text{reverse}(\text{tail}(A)), \langle \text{head}(A) \rangle).$$

   Therefore we have

$$\begin{aligned} f(\text{tail}(A), \text{head}(A) :: B) &= \text{cat}(\text{reverse}(\text{tail}(A)), \text{head}(A) :: B) \\ &= \text{cat}(\text{cat}(\text{reverse}(\text{tail}(A)), \langle \text{head}(A) \rangle), B) \\ &= \text{cat}(\text{reverse}(A), B). \end{aligned}$$

   Therefore $f(A, B)$ = cat(reverse(A), B) for all lists A and B.

Chapter 5

Analysis Techniques

5.1 Analyzing Algorithms

Learning Objectives

Be able to describe the meaning of worst case analysis of an algorithm.

Be able to find lower bounds on decision tree algorithms.

Review Questions

1. How do you find the worst case running time for an algorithm?

2. What does lower bound mean with respect to solving a problem?

3. What does it mean to say an algorithm is optimal in the worst case?

4. What is the meaning of the symbol W_A?

5. What is a decision tree for an algorithm?

Solved Problems

1. A solution to a problem has x possible outcomes. An algorithm to solve the problem has a binary decision tree of depth 6. What can the value of x be?
2. A solution to a problem has 85 possible outcomes. An algorithm to solve the problem has an n-way decision tree of depth 4. What is the smallest value that n could be?
3. A solution to a problem has 87 possible outcomes. An algorithm A to solve the problem has a ternary decision tree of depth d. What is the smallest value that d could be?
4. Find a good lower bound for the maximum number of weighings that are required by any ternary pan balance algorithm to discover the bad coin among 15 coins, where the one bad coin is either heavier or lighter than the other coins and where the algorithm must state whether the coin is heavy or light.

5. Suppose that for a problem P we have discovered a worst-case lower bound of $\lceil n/3 \rceil$, which counts the number of times a procedure S is executed with respect to an input $n \in \mathbb{N}$. Show that the following algorithm to solve P is optimal.

```
i := 0;
while i < n do
    S;
    i := i + 3
od
```

6. Suppose that for a problem P we have discovered a worst-case lower bound of $\lceil \log_2 n \rceil$, which counts the number of times a procedure S is executed with respect to an input $n \in \mathbb{N}$. Show that the following algorithm to solve P is optimal.

```
i := 1;
while i < n do
    S;
    i := 2i
od
```

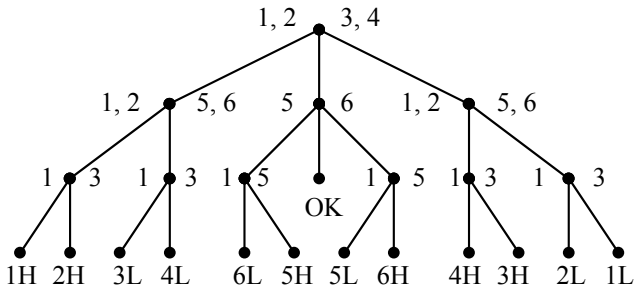
7. Find an optimal pan balance algorithm to find a bad coin, if it exists, among six coins, where at most one coin is bad, which means it is heavier or lighter than the others.

Solutions

1. Since there are x possible outcomes, there must be at least x leaves on the binary decision tree of depth 6. Such a binary tree has at most 2^6 leaves, so we have $x \leq 2^6$
2. Since there are 85 possible outcomes, there must be at least 85 leaves in the n-way decision tree of depth 4. Such a tree has at most n^4 leaves. Therefore, we have $85 \leq n^4$. Solving for n, we have ceiling($\sqrt[4]{85}$) $\leq n$, which gives $n \geq 4$.
3. Since there are 87 possible outcomes, there must be at least 87 leaves in the ternary decision tree of depth d. Such a tree has at most 3^d leaves, so we have $87 \leq 3^d$. Solving for d, we have ceiling($\log_3 87$) $\leq d$, which gives $d \geq 5$.
4. Since there is one bad coin that could be either heavy or light, there are 30 possible outcomes (each of 15 coins could be heavy or light). It follows that there must be at least 30 leaves in the ternary decision tree that represents the pan balance algorithm. Let d be the depth of the decision tree. Then the tree has at most 3^d leaves, so we have $30 \leq 3^d$. Solving for d, we have ceiling($\log_3 30$) $\leq d$, which gives $d \geq 4$. Therefore, four is a good lower bound for the maximum number of weighings required by the algorithm.
5. Notice that the body of the while loop is entered $k + 1$ times as i takes on the values 0, 3, 6, ..., $3k$, where $3k < n \leq 3(k + 1)$. Dividing by 3 we obtain $k < n/3 \leq k + 1$. In other words, we have $k + 1 = \lceil n/3 \rceil$.

6. Notice that the body of the while loop is entered $k + 1$ times as i takes on the values 1, 2, 4, ..., 2^k, where $2^k < n \leq 2^{k+1}$. Taking $\log_2$ we obtain $k < \log_2 n \leq k + 1$. Therefore, $k + 1 = \lceil \log_2 n \rceil$.

7. First we'll find a lower bound for the maximum number of weighings by a pan balance algorithm that is represented by a ternary decision tree. This can be found by noticing that there are 13 possible outcomes (each of 6 coins heavy or light, and all good coins). It follows that there must be at least 13 leaves in a ternary decision tree of depth d that has a maximum of 3^d leaves. In other words, we have $13 \leq 3^d$, which implies that $d \geq 3$. So 3 is a lower bound for the depth of ternary decision trees to solve the problem. An example optimal algorithm can be represented as follows, where the coins are numbered 1, 2, 3, 4, 5, 6, and the possible outcomes are labeled 1H, 1L, ..., 6H, 6L, and OK (for no bad coins).



5.2 Summations and Closed Forms

Learning Objectives

Be able to find closed forms for simple summations.

Be able to find approximate values for simple summations.

Review Questions

What does it mean to say an expression is in closed form?

What is a collapsing sum?

What is a simple way to find upper and lower bounds for a finite sum?

What is a harmonic number?

Solved Problems

1. Expand each expression into a sum of terms. Don't evaluate.

   a. $\sum_{k=0}^{3}(3k+5)$. b. $\sum_{k=2}^{4}k2^k$. c. $\sum_{k=1}^{3}(4-3k)5^k$.

2. For each of the following summations find an equivalent summation expression that starts with the lower limit of $k = -1$.

   a. $\sum_{k=0}^{3}(3k+5)$. b. $\sum_{k=2}^{4}k2^k$. c. $\sum_{k=1}^{3}(4-3k)5^k$.

3. For each of the following summations find an equivalent summation expression that starts with that lower limit of $k = 3$.

   a. $\sum_{k=0}^{3}(3k+5)$. b. $\sum_{k=2}^{4}k2^k$. c. $\sum_{k=1}^{3}(4-3k)5^k$.

4. Find a closed form for each of the following sums.

   a. $5+10+15+\cdots+5n$.

   b. $1+6+11+\cdots+(5n+1)$.

   c. $5+10+20+\cdots+5\cdot 2^n$.

5. Find a closed form for the following expression:

$$\sum_{k=0}^{n}[3^k+(n-i)2^k].$$

6. Find a closed form, in terms of n, for the number of times S is executed in the following algorithm:

```
for i := 1 to n do
    for j := i to n do
        S
    od
od
```

7. Given the following algorithm.

```
i := 0;
while i ≤ n do
    j := i;
    while j ≤ n do S; j := j + 1 od;
    i := i + 1
od
```

   a. Find a closed form expression for the number of times, in terms of the natural number n, that S is executed.

   b. Find a closed form expression for the number of times, in terms of the natural number n, that the operation + is executed.

c. Find a closed form expression for the number of times, in terms of the natural number n, that := is executed.

8. Given the following algorithm.

```
i := 0;
while i < n do
    j := i + 2;
    while j < n do S; j := j + 1 od;
    i := i + 1
od
```

a. Find a closed form expression for the number of times, in terms of the natural number n, that S is executed.

b. Find a closed form expression for the number of times, in terms of the natural number n, that the operation + is executed.

c. Find a closed form expression for the number of times, in terms of the natural number n, that := is executed.

9. Find bounds on the value of $\sum_{k=1}^{n}\sqrt{k}$ as indicated.

a. Use the least value and the greatest value to obtain lower and upper bounds.

b. Split the sum into two sums: a sum as k goes from 1 to $\lfloor n/2 \rfloor$ and a sum as k goes from $\lfloor n/2 \rfloor + 1$ to n. Then apply Part (a) to each sum.

10. Find bounds on the value of $\sum_{k=1}^{n}\log 2k$ as indicated in Parts (a) and (b) of Exercise 9.

11. Find a closed form for $\sum_{k=1}^{n} k/(k+2)$ in terms of harmonic numbers.

Solutions

1. a. $(3\cdot 0 + 5) + (3\cdot 1 + 5) + (3\cdot 2 + 5) + (3\cdot 3 + 5)$.

   b. $2\cdot 2^2 + 3\cdot 2^3 + 4\cdot 2^4$.

   c. $(4 - 3\cdot 1)5^1 + (4 - 3\cdot 2)5^2 + (4 - 3\cdot 3)5^3$.

2. a. $\sum_{k=-1}^{2}[3(k+1)+5]$. b. $\sum_{k=-1}^{1}(k+3)2^{k+3}$. c. $\sum_{k=-1}^{1}(4-3(k+2))5^{k+2}$.

3. a. $\sum_{k=3}^{6}[3(k-3)+5]$. b. $\sum_{k=3}^{5}(k-1)2^{k-1}$. c. $\sum_{k=3}^{5}(4-3(k-2))5^{k-2}$.

4. a. $\sum_{k=1}^{n}5k = 5\sum_{k=1}^{n}k = \frac{5n(n+1)}{2}$.

   b. $\sum_{k=0}^{n}(5k+1) = 5\sum_{k=0}^{n}k + \sum_{k=0}^{n}1 = \frac{5n(n+1)}{2} + (n+1) = \frac{(n+1)(5n+2)}{2}$.

c. $\sum_{k=0}^{n} 5 \cdot 2^k = 5 \sum_{k=0}^{n} 2^k = 5(2^{n+1} - 1).$

5.

$$\begin{aligned}\sum_{k=0}^{n} [3^k + (n-k)2^k] &= \sum_{k=0}^{n} 3^k + n \sum_{k=0}^{n} 2^k - \sum_{k=0}^{n} k2^k \\ &= \frac{3^{n+1} - 1}{2} + n(2^{n+1} - 1) - [2 - (n+1)2^{n+1} + n2^{n+2}] \\ &= \frac{3^{n+1} - 1}{2} + n2^{n+1} - n - 2 + n2^{n+1} + 2^{n+1} - n2^{n+2} \\ &= \frac{3^{n+1} - 1}{2} - n - 2 + 2^{n+1} \\ &= \frac{3^{n+1} - 2^{n+2} - 2n - 5}{2}.\end{aligned}$$

6. The body of the outer loop is executed n times as i goes through the values 1, 2, ..., n. For each of these values of i, the body of the inner loop, which contains S, is executed $n - i + 1$ times, as j goes through the values i, $i + 1$, ..., n. Therefore the number of times S is executed is given by the following summation and closed form.

$$\begin{aligned}\sum_{i=1}^{n} (n - i + 1) &= \sum_{n=1}^{n} (n+1) - \sum_{i=1}^{n} i \\ &= n(n+1) - \frac{n(n+1)}{2} \\ &= \frac{n(n+1)}{2}.\end{aligned}$$

7. a. The body of the outer loop is executed $n + 1$ times as i goes through the values 0, 1, ..., n. For each of these values of i, the body of the inner loop, which contains S, is executed $n - i + 1$ times, as j goes through the values i, $i + 1$, ..., n. Therefore the number of times S is executed is given by the following summation and closed form.

$$\begin{aligned}\sum_{i=0}^{n} (n - i + 1) &= \sum_{n=0}^{n} (n+1) - \sum_{i=0}^{n} i \\ &= (n+1)^2 - \frac{n(n+1)}{2} \\ &= \frac{(n+1)(n+2)}{2}.\end{aligned}$$

b. Add $n + 1$ to the result of Part (a).

c. Add $1 + 2(n + 1)$ to the result of Part (a).

8. a. The body of the outer loop is executed n times as i goes through the values 0, 1, ..., $n - 1$. For each of these values of i, the body of the inner loop, which contains S, is executed $n - i - 2$ times, as j goes through the values $i + 2$, $i + 3$, ..., $n - 1$. Therefore the number of times S is executed is given by the following summation and closed form:

$$\begin{aligned}\sum_{i=0}^{n-1}(n-i-2) &= \sum_{n=0}^{n-1}(n-2) - \sum_{i=0}^{n-1} i \\ &= n(n-2) - \frac{n(n-1)}{2} \\ &= \frac{n(n-3)}{2}.\end{aligned}$$

   b. Add $2n$ to the result of Part (a).

   c. Add $1 + 2n$ to the result of Part (a).

9. a. The lower bound is n and the upper bound is $n^{3/2}$.

   b. Since $n = \lfloor n/2 \rfloor + \lceil n/2 \rceil$, there are $\lceil n/2 \rceil$ terms in the second sum. So the lower and upper bounds are $\lfloor n/2 \rfloor + \lceil n/2 \rceil \sqrt{\lfloor n/2 \rfloor + 1}$ and $\lfloor n/2 \rfloor \sqrt{\lfloor n/2 \rfloor} + \lceil n/2 \rceil \sqrt{n}$.

10. a. The lower bound is $n \log 2$ and the upper bound is $n \log 2n$.

   b. Since $n = \lfloor n/2 \rfloor + \lceil n/2 \rceil$, there are $\lceil n/2 \rceil$ terms in the second sum. So the lower and upper bounds are $\lfloor n/2 \rfloor \log 2 + \lceil n/2 \rceil \log\left(2\lfloor n/2 \rfloor + 1\right)$ and $\lfloor n/2 \rfloor \log\left(2\lfloor n/2 \rfloor\right) + \lceil n/2 \rceil \log 2n$.

11. $$\sum_{k=1}^{n} \frac{k}{k+2} = \sum_{k=1}^{n}\left(1 - \frac{2}{k+2}\right) = n - 2\sum_{k=1}^{n}\frac{1}{k+2} = n - 2\sum_{k=3}^{n+2}\frac{1}{k} = n - 2\left(H_{n+2} - H_2\right).$$

5.3 Permutations and Combinations

Learning Objectives

Be able to use elementary counting techniques to count simple finite structures that are either ordered or unordered.

Review Questions

1. What is a permutation?

2. What is a permutation of a bag?

3. What is a combination?

4. What is a bag combination?

5. What is the meaning of each of the following symbols or expressions?

   a. $P(n, r)$.

   b. $C(n, r)$.

   c. $\binom{n}{r}$.

6. What is Pascal's triangle?

Solved Problems

1. For each of the following expressions, calculate the value expression and then write down a typical problem whose answer is given by the expression.

   a. 7!. b. 23!/18!. c. $C(4 + 6 – 1, 6)$. d. 10!/(2!3!4!).

2. How many sets of 3 courses can be chosen from a set of 10 courses?
3. Given a movie theater that shows 7 different movies each day, how many ways can 3 different movies be seen, one on each of three successive days?
4. How many committees consisting of 4 people can be chosen from 15 people?
5. How many different strings can be made by rearranging the letters of the word "LETTERS"?
6. How many different lists of length 8 can be made from the eight numbers in the set {1, 2, 3, 4, 5, 6, 7, 8}, where each list contains all eight numbers?
7. How many bags of 7 pieces of fruit can be chosen from a collection of bananas, apples, oranges, dates, and pears?
8. Find the number of permutations of the letters in the word "radon" (i.e., in the set $\{r, a, d, o, n\}$).
9. Find the number of sets of five cans of soda that can be chosen from a machine that dispenses four different kinds of soda.
10. For each of the following restrictions, find the smallest size n for strings over $\{a, b, c\}$ that can be used as codes for 27 people.

    a. There are k a's, l b's, and m c's and $k + l + m = n$.

    b. Same as Part (a) with the additional restriction that $k = l = m$.

Solutions

1. a. 7! = 5,040. How many ways can seven people be arranged in a row?
   b. 23!/18! = 4,037,880. How many different arrangements of 5 people can be made from a set of 23 people?
   c. $C(4 + 6 - 1, 6) = 84$. How many ways can six coins be selected from a collection of pennies, nickels, dimes, and quarters?
   d. 10!/(2!3!4!) = 12,600. How many ways can the letters in the string *aabbbccccd* be arranged?
2. $C(10, 3) = 120$.
3. $P(7, 3) = 210$.
4. $C(15, 4)$.
5. 7!/(2!2!).
6. 8!.
7. $C(5 + 7 - 1, 7)$.
8. 5!
9. $C(4 + 5 - 1, 5)$
10. a. The number of strings of length n over $\{a, b, c\}$ that contain k a's, l b's, and m c's is given by $n!/(k!l!m!)$. Solve the inequality $n!/(k!l!m!) \geq 27$ for the smallest n such that $k + l + m = n$. By trial and error, we obtain $n = 5$, where two of the three numbers k, l, and m are 2 and the other one is 1.
    b. The number of strings of length n over $\{a, b, c\}$ that contain k a's, k b's, and k c's is given by $n!/3(k!)$. Solve the inequality $n!/3(k!) \geq 27$ for the smallest n such that $3k = n$. Trial and error gives $n = 6$ and $k = 2$.

5.4 Discrete Probability

Learning Objectives

Be able to use discrete probability to solve simple problems.

Be able to describe the average-case analysis of an algorithm.

Review Questions

1. What is a sample space?

2. What is a probability distribution?

3. What is an event?

4. What is the conditional probability of A given B?

5. What are independent events?

6. What is expectation?

7. How do you find the average case running time for an algorithm?

8. What is a Markov chain?

9. What is the Monte Carlo method?

Solved Problems

1. Write down a probability distribution for tossing an unfair die, where it lands with the number 6 on top 25% of the time, and the other numbers are equally likely to be on top.
2. Suppose four fair coins are flipped. Find the probability of each event.
   a. Exactly three coins are heads.
   b. At most one coin is a head.
   c. At most three coins are heads.
3. Suppose a pair of fair dice are tossed. Find the probability of each event.
   a. The sum of the dots is 9.
   b. The sum of dots is at most 4.
   c. The sum of dots is at least 5.
4. When three fair dice are tossed, what is the probability that the sum of the dots is 10?
5. Suppose that an unfair die is tossed. Let $P(n)$ denote the probability that the number of dots on top is n. Assume that $P(2) = P(4) = P(6) = 1/4$ and $P(1) = P(3) = P(5) = 1/12$. What is the expected number of dots on the top of the die?
6. Let the set $S = \{I_1, I_2, ..., I_8\}$ represent 8 types of input to an algorithm. Suppose that the inputs occur with the following probabilities:

   $$P(I_1) = P(I_2) = P(I_3) = P(I_4) = 1/16,\ P(I_5) = P(I_6) = 1/8,\ P(I_7) = P(I_8) = 1/4.$$

   Suppose further that each input I_n causes the algorithm to execute $2n$ instructions. What is the expected number of instructions executed by the algorithm?
7. A faculty senate consists of 30 members of whom 20 are tenured. Of the 20 who are tenured, 12 are women and 8 are men; of the 10 who are not tenured, 4 are women and 6 are men. A senator is randomly chosen.
   a. What is the probability that the senator is a woman.

b. Given the senator is tenured, what is the probability that the senator is a woman?

8. A new website with two pages, A and B, has been tested with volunteers. The results of the test show that when a person is on page A, the probability of clicking on a link to stay on page A is 0.8 and the probability of clicking on a link to go to page B is 0.2. When on page B, the probabilities of going to page A and page B are 0.6 and 0.4, respectively.

   a. Find the transition matrix P for this Markov chain.

   b. Find the unique fixed probability vector X for P.

   c. Assuming that a person is equally likely to start on page A or page B, what is the probability that the first click on a link will put the person on page A? What is the probability that after the second click on a link the person will be on page A? What about after the third click on a link? After a large number of clicks, what is the probability that the person will be on page A?

Solutions

1. $P(6) = 1/4$ and $P(1) = P(2) = P(3) = P(4) = P(5) = 3/20$.

2. Let H and T denote head and tail. Then we can denote the result of flipping the four coins as a string of length 4 over the alphabet $\{H, T\}$. So there are 16 possible outcomes, each with a probability of 1/16.

   a. There are 4 outcomes that contain exactly three heads, namely *THHH*, *HTHH*, *HHTH*, and *HHHT*. So the probability of this event is $4(1/16) = 1/4$.

   b. The event that at most one coin is a head has 5 outcomes, *TTTT*, *HTTT*, *THTT*, *TTHT*, *TTTH*. So the probability of this event is $5(1/16) = 5/16$.

   c. The event that at most three coins are heads is the complement of the event $\{HHHH\}$, which has probability 1/16. So the event has probability $1 – P(HHHH) = 1 – 1/16 = 15/16$.

3. The sample space has 36 possible outcomes where each outcome can be represented by (i, j), where i and j are the number of dots on the top of the two dice. Since the dice are fair, we have $P(i, j) = 1/36$.

   a. The event that the sum of the dots is 9 is the set $\{(3, 6), (4, 5), (5, 4), (6, 3)\}$. So the probability of the event is $4(1/36) = 1/9$.

   b. The event that the sum of the dots is at most 4 is the set

   $$\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 1)\}.$$

   So the probability of the event is $6(1/36) = 1/6$.

   c. The event that the sum of the dots is at least 5 is the complement of the event that the sum is at most 4, which is the set

   $$\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 1)\}.$$

   So the probability that the sum is at least 5 is $1 –1/6 = 5/6$.

4. A sample space for the experiment contains all 3-tuples (i, j, k) where i, j, and k represent the number of dots on each die. So there are $6^3 = 216$ points in the sample space. Since the dice are fair, the probability of any 3-tuple is 1/216. Next we need to find the number of points (i, j, k) such that $i + j + k = 10$. If we listed them all, then we would be listing all permutations of each 3-tuple. For example, since 2 + 2 + 6 = 10, we need to list the three permutations of the bag [2, 2, 6]. Instead, we'll list the distinct bags of three numbers that add up to 10, and then count of the number of permutations for each bag. Here are the bags, some of which are sets, followed by the number of permutations.

Bag	Permutations
[1, 3, 6]	3! = 6
[1, 4, 5]	3! = 6
[2, 2, 6]	3!/2! = 3
[2, 3, 5]	3! = 6
[2, 4, 4]	3!/2! = 3
[3, 3, 4]	3!/2! = 3

So the total number of 3-tuples is 27. Therefore the desired probability is 27/216 = 1/8.

5. The expectation is the following sum of the products.

$$\sum_{i=1}^{6} iP(i) = (2 + 4 + 6)\left(\tfrac{1}{4}\right) + (1 + 3 + 5)\left(\tfrac{1}{12}\right) = 3.75.$$

6. The expectation is the following sum of the products.

$$\begin{aligned}\sum_{i=1}^{8} 2iP(I_i) &= 2\sum_{i=1}^{8} iP(I_i) \\ &= 2[(1 + 2 + 3 + 4)\left(\tfrac{1}{16}\right) + (5 + 6)\left(\tfrac{1}{8}\right) + (7 + 8)\left(\tfrac{1}{4}\right)] \\ &= 11.5.\end{aligned}$$

7. Let A be the event "is a woman" and let B be the event "is tenured". Then the assumptions give us the following cardinalities: $|A| = 16$, $|B| = 20$, and $|A \cap B| = 12$. Then we have the following solutions.
   a. $P(A) = |A|/30 = 16/30 = 8/15$.
   b. $P(A|B) = P(A \cap B)/P(B) = (12/30)/(20/30) = 3/5$.

8. a. $P = \begin{pmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{pmatrix}$.
   b. Solve the equation $XP = X$ for probability vector X to obtain $X = (0.75, 0.25)$.
   c. The initial probability vector is $X_0 = (0.5, 0.5)$. So the first three answers are the first components of the vectors X_0P, X_0P^2, and X_0P^3. The last answer is the first component of X. So we have the answers 0.7, 0.74, 0.748, 0.75.

5.5 Solving Recurrences

Learning Objectives

Be able to find closed form solutions for simple recurrences using the techniques of substitution, cancellation, and generating functions.

Review Questions

1. What is a recurrence?

2. What does it mean to solve a recurrence?

3. What form of recurrence can be solved by substitution or cancellation?

4. What is a divide-and-conquer recurrence?

5. What is a generating function?

6. How does one solve a recurrence with generating functions?

Solved Problems

1. Solve the following recurrence for a_n by cancellation or substitution and write the answer in closed form.

   $a_1 = 1$
   $a_n = 2a_{n-1} + 9 \qquad (n \geq 2).$

2. Solve the following recurrence for a_n by cancellation or substitution and write the answer in closed form.

   $a_0 = 1$
   $a_n = 3a_{n-1} + 5 \qquad (n \geq 1)$

3. Solve the following recurrence for a_n by cancellation or substitution and write the answer in closed form.

   $a_0 = 1$
   $a_n = 3a_{n-1} + 7n \qquad (n \geq 1)$

4. Given the following procedure P defined for all natural numbers n.

   ```
   P(n):  if n = 0 then
              C(0)
          else
              P(n – 1);
              C(n);
              P(n – 1)
          fi
   ```

Let a_n denote the number of times that a certain operation is executed during the execution of $P(n)$. Suppose that $C(n)$ executes the operation $3n$ times. Write down a recurrence to describe a_n. Then solve it.

5. Suppose we have a recurrence whose nth term is a_n. Let

$$A(x) = \sum_{n=0}^{\infty} a_n x^n.$$

For each of the following closed forms for $A(x)$, find the corresponding closed form for a_n.

a. $\dfrac{1}{1-dx}$. b. $\dfrac{1}{c-dx}$. c. $\dfrac{b}{c-dx}$.

6. Suppose we have a recurrence whose nth term is a_n. Let

$$A(x) = \sum_{n=0}^{\infty} a_n x^n.$$

For each of the following closed forms for $A(x)$, find the corresponding closed form for a_n.

a. $\dfrac{1}{1+x}$. b. $\dfrac{1}{1+dx}$. c. $\dfrac{1}{c+dx}$. d. $\dfrac{b}{c+dx}$.

7. Suppose we have a recurrence whose nth term is a_n. Let

$$A(x) = \sum_{n=0}^{\infty} a_n x^n.$$

For each of the following closed forms for $A(x)$, find the corresponding closed form for a_n.

a. $\dfrac{1}{(1+x)^2}$. b. $\dfrac{b}{(c+dx)^2}$. c. $\left(\dfrac{b}{c-dx}\right)^3$.

8. Suppose we have a recurrence whose nth term is a_n. Let

$$A(x) = \sum_{n=0}^{\infty} a_n x^n.$$

Suppose we have the following closed form for A(x).

$$A(x) = \frac{2}{3+4x} - \frac{5}{(6-7x)^2}.$$

Find the closed form for a_n.

9. Given the following algorithm:

$f(n) =$ if $n = 0$ then 1
else if $n = 1$ then 2
else $f(n-1) + f(n-2)$.

Find a recurrence formula for each of the following conditions.

a. The number of times that f is called.

b. The number of times that + is executed.

10. Given the following recurrence:

$a_0 = 1$
$a_1 = 2$
$a_n = 5a_{n-1} - 4a_{n-2} \quad (n \geq 2)$

Use the generating function technique to find a closed form for a_n.

11. Find a closed form for the following divide-and-conquer recurrence, where $n = 2^k$:

$T(1) = 1$
$T(n) = 2T(n/2) + \log_2 n$

Solutions

1. $5 \cdot 2^n - 9$.

2. $(7 \cdot 3^n - 5)/2$.

3. By using either cancellation or substitution we obtain the following expression for a_n.

$$\begin{aligned} a_n &= 3^n + 7\left(n + 3(n-1) + 3^2(n-2) + \cdots + 3^{n-1}\right) \\ &= 3^n + 7\sum_{i=0}^{n-1} 3^i(n-i) \\ &= 3^n + 7n\sum_{i=0}^{n-1} 3^i - 7\sum_{i=0}^{n-1} i3^i \\ &= 3^n + 7n\left(\frac{3^n - 1}{2}\right) - 7\left(\frac{3 - n3^n + (n-1)3^{n+1}}{4}\right) \\ &= \frac{25 \cdot 3^n - 14n - 21}{4}. \end{aligned}$$

4. Since $P(0) = C(0)$, we have $a_0 = 0$. When $n \geq 0$ the definition tells us that $P(n)$ calls $P(n-1)$ twice and $C(n)$ once. Therefore $a_n = 2a_{n-1} + 3n$. So the definition of a_n is given by the recurrence

$a_0 = 0$
$a_n = 2a_{n-1} + 3n \quad$ for $n \geq 1$.

Using cancellation or substitution and summation formulas we obtain the following closed form solution.

$$\begin{aligned} a_n &= 3\left(n + 2(n-1) + 2^2(n-2) + \cdots + 2^{n-1}\right) \\ &= 3\sum_{i=0}^{n-1} 2^i (n-i) \\ &= 3n\sum_{i=0}^{n-1} 2^i - 3\sum_{i=0}^{n-1} i2^i \\ &= 3n\left(2^n - 1\right) - 3\left(2 - n2^n + (n-1)2^{n+1}\right) \\ &= 6 \cdot 2^n - 3n - 6. \end{aligned}$$

5. a. $a_n = d^n$. b. $a_n = \left(\frac{1}{c}\right)\left(\frac{d}{c}\right)^n$. c. $a_n = \left(\frac{b}{c}\right)\left(\frac{d}{c}\right)^n$.

6. a. $a_n = (-1)^n$. b. $a_n = (-d)^n$. c. $a_n = \left(\frac{1}{c}\right)\left(-\frac{d}{c}\right)^n$. d. $a_n = \left(\frac{b}{c}\right)\left(-\frac{d}{c}\right)^n$.

7. a. $a_n = (n+1)(-1)^n$. b. $a_n = \left(\frac{b}{c^2}\right)(n+1)\left(\frac{-d}{c}\right)^n$.

   c. $a_n = \left(\frac{b}{c}\right)^3\left(\frac{(n+1)(n+2)}{2}\right)\left(\frac{d}{c}\right)^n$.

8. $a_n = \left(\frac{2}{3}\right)\left(-\frac{4}{3}\right)^n - \left(\frac{5}{36}\right)(n+1)\left(\frac{7}{6}\right)^n$.

9. a. $a_0 = 1$
   $a_1 = 1$
   $a_n = a_{n-1} + a_{n-2}$.

   b. $a_0 = 0$
   $a_1 = 0$
   $a_n = 1 + a_{n-1} + a_{n-2}$

10. Let $A(x) = \sum_{n=0}^{\infty} a_n x^n$. Then the general recurrence for a_n gives the following equation:

$$A(x) - 2x - 1 = 5x(A(x) - 1) - 4x^2 A(x).$$

Solving the equation for $A(x)$ gives the following sequence of equations:

$$\begin{aligned}
A(x) &= \frac{1-3x}{1-5x+4x^2} \\
&= \frac{1-3x}{(1-x)(1-4x)} \\
&= \frac{2/3}{1-x} + \frac{1/3}{1-4x} \\
&= \left(2/3\right)\sum_{n=0}^{\infty} x^n + \left(1/3\right)\sum_{n=0}^{\infty}(4x)^n \\
&= \sum_{n=0}^{\infty}\left(2/3 + \left(1/3\right)4^n\right)x^n.
\end{aligned}$$

Therefore, $a_n = 2/3 + \left(1/3\right)4^n = \dfrac{2+4^n}{3}$.

11. The recurrence fits the form given in (5.34), where $a = b = 2$ and $f(n) = \log_2 n$. Since $n = 2^k$, we can use (5.35) to write $T(n)$ as follows:

$$T(n) = 2^k T(1) + \sum_{i=0}^{k-1} 2^i \log_2(n/2^i).$$

We can find a closed form for $T(n)$ by using properties of sums and logs together with the given information that $n = 2^k$ and thus also $k = \log_2 n$.

$$\begin{aligned}
T(n) &= 2^k T(1) + \sum_{i=0}^{k-1} 2^i \log_2(n/2^i) \\
&= 2^k + \sum_{i=0}^{k-1} 2^i(\log_2 n - \log_2 2^i) \\
&= 2^k + \sum_{i=0}^{k-1} 2^i(\log_2 n - i\log_2 2) \\
&= 2^k + \sum_{i=0}^{k-1} 2^i(k-i) \\
&= 2^k + k\sum_{i=0}^{k-1} 2^i(k-i) \\
&= 2^k + k\sum_{i=0}^{k-1} 2^i - \sum_{i=0}^{k-1} i2^i \\
&= 2^k + k\left(2^k - 1\right) - \left(2 - k2^k + (k-1)2^{k+1}\right) \\
&= 3\cdot 2^k - k - 2 \\
&= 3n - \log_2 n - 2.
\end{aligned}$$

5.6 Comparing Rates of Growth

Learning Objectives

Be able to compare simple functions by rate of growth.

Review Questions

1. What does it mean to say that the growth rate of f is bounded above by the growth rate of of g?

2. What does it mean to say two functions have the same order?

3. What does it mean to say f has lower order than g?

4. What is the meaning of each of the following expressions?

   a. $O(f)$.

   b. $\Omega(f)$.

   c. $\Theta(f)$.

   d. $o(f)$.

5. What problems have solutions that can be approximated by the Akra-Bazzi method?

Solved Problems

1. Prove that $n^2 = \Theta(500n^2)$.
2. Prove that $\log(\log n) = o((\log n)^2)$.
3. Indicate the "order of growth" relationships between the following expressions by listing them from lowest order to highest order. Also be sure to indicate expressions having the same order by placing them in a set.

   $$n^2, \quad n \log(n^2), \quad 2(\log n)^2, \quad \log n, \quad \log(\log n).$$

4. Indicate the "order of growth" relationships between the following expressions by listing them from lowest order to highest order. Also be sure to indicate expressions having the same order by placing them in a set.

   n^2, $n \log n$, 2^n, $\log(n^2)$, $\text{floor}(n/2)$, $(\log n)^2$, 2^n, $\log(\log n)$.

5. Prove each of the following statements.

   a. If $0 < k \leq 1$, then $\log(1 + k + k^2 + \ldots + k^n) = O(\log n)$.

   b. If $k > 1$, then $\log(1 + k + k^2 + \ldots + k^n) = \Theta(n)$.

6. Find an approximation for each of the following divide and conquer recurrences, where n is a power of 2 and $T(1) = 1$.

   a. $T(n) = 2T(n/2) + 1$.

   b. $T(n) = 2T(n/2) + 3n$.

   c. $T(n) = 2T(n/2) + 5\log n$.

Solutions

1. We can use the definition of big theta by observing, for example, the inequalities $(1/500)\cdot 500n^2 \le n^2 \le (1)\cdot 500n^2$ for all $n \ge 0$. We could also look at the limit of the quotient $500n^2/n^2 = 500$, which is not 0 and not infinity. So (5.30) gives the result.

2. For convenience, we can assume that base e is used for the logs. Then we can take the limit of the quotients as follows:

$$\lim_{n\to\infty}\frac{\log(\log n)}{(\log n)^2} = \lim_{n\to\infty}\frac{\left(1/\log n\right)\left(1/n\right)}{\left(2\log n/n\right)} = \lim_{n\to\infty}\frac{1}{2(\log n)^2} = 0.$$

   Since the limit is zero, we obtain $\log(\log n) = o((\log n)^2)$.

3. $\log(\log n)$, $\log n$, $2(\log n)^2$, $n\log(n^2)$, n^2.

4. $\log(\log n)$, $\log(n^2)$, $(\log n)^2$, $\{2n, \text{floor}(n/2)\}$, $n\log n$, n^2, 2^n.

5. a. If $0 < k \le 1$, then $k^i \le 1$ for $0 \le i \le n$. So we can write the following inequality:

$$\begin{aligned}\log(1 + k + k^2 + \cdots + k^n) &\le \log(1 + 1 + 1 + \cdots + 1)\\ &= \log(n+1)\\ &= O(\log n).\end{aligned}$$

   b. If $k > 1$, then $\log(k) > 0$ and we have the following inequality.

$$n\log(k) = \log(k^n) \le \log(1 + k + k^2 + \cdots + k^n).$$

   For the other inequality, recall from the binomial theorem that

$$(1+k)^n = 1 + C(n,1)k + C(n,2)k^2 + \cdots + k^n.$$

   Since each $C(n, i) \ge 1$, we have the inequality

$$(1+k)^n \ge 1 + k + k^2 + \cdots + k^n.$$

   Now apply the log to this inequality to obtain

$$\log(1 + k + k^2 + \cdots + k^n) \le \log((1+k)^n) = n\log(1+k).$$

   So there are positive constants $c = \log k$ and $d = \log(1 + k)$ such that

$$cn \le \log(1 + k + k^2 + \cdots + k^n) \le dn.$$

   Therefore, $\log(1 + k + k^2 + \cdots + k^n) = \Theta(n)$.

An alternative way to proceed is to use the formula for a geometric progression to obtain

$$1 + k + k^2 + \cdots + k^n = \frac{k^{n+1} - 1}{k - 1}.$$

Then show that the following limit is nonzero.

$$\lim_{n \to \infty} \frac{\log\left((k^{n+1} - 1)/(k - 1)\right)}{n}.$$

6. a. We can use the divide-and-conquer test (5.66). In this case, we have $a = b = 2$ and $\alpha = \beta = 0$. So we have $\alpha < \log_b a$. Therefore, $T(n) = \Theta(n)$.

   b. Again, we can use the test (5.66). In this case, we have $a = b = 2$, $\alpha = 1$, and $\beta = 0$. So we have $\alpha = \log_b a$. Since $\beta > -1$, it follows that $T(n) = \Theta(n \log n)$.

   c. Again, we can use the test (5.66). In this case, we have $a = b = 2$, $\alpha = 0$, and $\beta = 1$. So we have $\alpha < \log_b a$. Therefore, $T(n) = \Theta(n)$.

Chapter 6

Elementary Logic

6.1 How Do We Reason?

Learning Objectives

Be able to describe the modus ponens rule, a non sequitur, and a calculus.

Review Questions

1. What is the modus ponens rule?

2. What is a non sequitur?

3. What is a calculus?

Solved Problems

1. How did you learn the modus ponens rule?
2. How would you teach a dog the modus ponens rule?

6.2 Propositional Calculus

Learning Objectives

Be able to determine whether a wff is a tautology, a contradiction, or a contingency by truth tables and by Quine's method.

Be able to construct equivalence proofs.

Be able to transform truth functions and wffs into conjunctive or disjunctive normal form.

Review Questions

1. What is the meaning of each of the following symbols or expressions?

   a. $\neg P$.

   b. $P \wedge Q$.

c. $P \vee Q$.

d. $P \rightarrow Q$.

e. $P \equiv Q$.

2. What is a wff in propositional calculus?
3. What is the meaning of a wff?
4. What is a tautology?
5. What is a contradiction?
6. What is a contingency?
7. When are two wffs equivalent?
8. What is Quine's method?
9. What is a truth function?
10. What does DNF mean?
11. What does CNF mean?
12. What is full DNF?
13. What is full CNF?
14. What is a literal?

Solved Problems

1. Find truth values for A, B, and C such that the wff $(A \rightarrow B) \rightarrow C$ is not equivalent to $A \rightarrow (B \rightarrow C)$.
2. Use Quine's method to analyze the truth value of the following wff.

$$(A \wedge B \rightarrow C) \wedge (A \rightarrow B) \rightarrow (A \rightarrow C).$$

3. Use Quine's method to analyze the truth value of the following wff.

$$(A \vee B) \wedge (A \rightarrow C) \wedge (C \rightarrow D) \rightarrow (C \vee D).$$

4. Use equivalences to prove that the following wff is equivalent to True.

$$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A).$$

5. Find a propositional wff to represent the truth function f defined by $f(A, B, C)$ is true iff exactly two of A, B, and C have the same truth value.

6. Find a conjunctive normal form (CNF) for the wff

$$(A \wedge B) \vee (C \wedge D \wedge E).$$

7. Find a disjunctive normal form (DNF) for the wff

$$(A \rightarrow B) \wedge (C \vee D).$$

8. Construct a full DNF for $P \vee Q$ using equivalences.

9. Construct a full CNF for the wff $P \wedge Q$ using equivalences.

10. Prove each of the following equivalences with equivalences.

    a. $(A \rightarrow B) \equiv \neg (A \wedge \neg B)$.

    b. $(A \rightarrow B) \rightarrow (A \rightarrow \neg (B \vee A)) \equiv A \rightarrow \neg B$.

    c. $(A \rightarrow B) \rightarrow (\neg (B \wedge C) \rightarrow \neg A) \equiv \neg A \vee \neg B \vee C$.

    d. $(A \wedge B \rightarrow C) \rightarrow (A \wedge C \rightarrow B) \equiv A \wedge C \rightarrow B$.

Solutions

1. One answer is A = False, C = False and B is either True or False.

2. If W is the wff, then calculate $W(C/\text{True})$ and $W(C/\text{False})$ as follows:

$$\begin{aligned} W(C/\text{True}) &= (A \wedge B \rightarrow \text{True}) \wedge (A \rightarrow B) \rightarrow (A \rightarrow \text{True}) \\ &\equiv \text{True} \wedge (A \rightarrow B) \rightarrow \text{True} \equiv \text{True}. \end{aligned}$$

$$\begin{aligned} W(C/\text{False}) &= (A \wedge B \rightarrow \text{False}) \wedge (A \rightarrow B) \rightarrow (A \rightarrow \text{False}) \\ &\equiv \neg (A \wedge B) \wedge (A \rightarrow B) \rightarrow \neg A. \end{aligned}$$

   If $U = W(C/\text{False})$, then calculate $U(A/\text{True})$ and $U(A/\text{False})$ as follows:

$$\begin{aligned} U(A/\text{True}) &\equiv \neg (\text{True} \wedge B) \wedge (\text{True} \rightarrow B) \rightarrow \neg \text{True} \\ &\equiv \neg B \wedge B \rightarrow \text{False} \equiv \text{False} \rightarrow \text{False} \equiv \text{True}. \end{aligned}$$

$$\begin{aligned} U(A/\text{False}) &\equiv \neg (\text{False} \wedge B) \wedge (\text{False} \rightarrow B) \rightarrow \neg \text{False} \\ &\equiv \neg \text{False} \wedge \text{True} \rightarrow \text{True} \equiv \text{True}. \end{aligned}$$

   Since all expressions evaluate to True, it follows that W is a tautology.

3. If W is the wff, then calculate $W(C/\text{True})$ and $W(C/\text{False})$ as follows:

$$W(C/\text{True}) = (A \vee B) \wedge (A \rightarrow \text{True}) \wedge (\text{True} \rightarrow D) \rightarrow (\text{True} \vee D) \equiv \text{True}.$$

$$\begin{aligned} W(C/\text{False}) &= (A \vee B) \wedge (A \rightarrow \text{False}) \wedge (\text{False} \rightarrow D) \rightarrow (\text{False} \vee D) \\ &\equiv (A \vee B) \wedge \neg A \wedge \text{True} \rightarrow D \\ &\equiv (A \vee B) \wedge \neg A \rightarrow D. \end{aligned}$$

If $U = W(C/\text{False})$, then calculate $U(D/\text{True})$ and $U(D/\text{False})$ as follows:

$$U(D/\text{True}) \equiv (A \vee B) \wedge \neg A \rightarrow \text{True} \equiv \text{True}.$$

$$\begin{aligned} U(D/\text{False}) &\equiv (A \vee B) \wedge \neg A \rightarrow \text{False} \\ &\equiv \neg ((A \vee B) \wedge \neg A). \end{aligned}$$

If $V = U(D/\text{False})$, then calculate $V(A/\text{True})$ and $V(A/\text{False})$ as follows:

$$\begin{aligned} V(A/\text{True}) &\equiv \neg ((\text{True} \vee B) \wedge \neg \text{True}) \\ &\equiv \neg (\text{True} \wedge \text{False}) \equiv \neg \text{False} \equiv \text{True}. \end{aligned}$$

$$\begin{aligned} V(A/\text{False}) &\equiv \neg ((\text{False} \vee B) \wedge \neg \text{False}) \\ &\equiv \neg (B \wedge \text{True}) \equiv \neg B. \end{aligned}$$

Since $\neg B$ can be true or false, it follows that W is a contingency.

4. $$\begin{aligned} (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A) &\equiv \neg (A \rightarrow B) \vee (\neg B \rightarrow \neg A) \\ &\equiv \neg (\neg A \vee B) \vee (\neg \neg B \vee \neg A) \\ &\equiv (\neg \neg A \wedge \neg B) \vee (B \vee \neg A) \\ &\equiv (A \wedge \neg B) \vee B \vee \neg A \\ &\equiv ((A \vee B) \wedge (\neg B \vee B)) \vee \neg A \\ &\equiv ((A \vee B) \wedge \text{True}) \vee \neg A \\ &\equiv (A \vee B) \vee \neg A \\ &\equiv (A \vee \neg A) \vee B \equiv \text{True} \vee B \equiv \text{True}. \end{aligned}$$

5. Notice that $f(A, B, C)$ is false iff $A = B = C = \text{True}$ or $A = B = C = \text{False}$. So we can construct a CNF of the following form:

$$f(A, B, C) = (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B \vee C) .$$

6. $$\begin{aligned} &(A \wedge B) \vee (C \wedge D \wedge E) \\ &\equiv [(A \wedge B) \vee C] \wedge [(A \wedge B) \vee D] \wedge [(A \wedge B) \vee E] \\ &\equiv (A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D) \wedge (A \vee E) \wedge (B \vee E). \end{aligned}$$

7. $$\begin{aligned} (A \rightarrow B) \wedge (C \vee D) &\equiv (\neg A \vee B) \wedge (C \vee D) \\ &\equiv [(\neg A \vee B) \wedge C] \vee [(\neg A \vee B) \wedge D] \\ &\equiv (\neg A \wedge C) \vee (B \wedge C) \vee (\neg A \wedge D) \vee (B \wedge D). \end{aligned}$$

8. $$\begin{aligned} P \vee Q &\equiv (P \wedge \text{True}) \vee (Q \wedge \text{True}) \\ &\equiv (P \wedge (Q \vee \neg Q)) \vee (Q \wedge (P \vee \neg P)) \\ &\equiv (P \wedge Q) \vee (P \wedge \neg Q) \vee (Q \wedge P) \vee (Q \wedge \neg P). \end{aligned}$$

9. $P \wedge Q \equiv (P \vee \text{False}) \wedge (Q \vee \text{False})$
$\equiv (P \vee (Q \wedge \neg Q)) \vee (Q \vee (P \wedge \neg P))$
$\equiv (P \vee Q) \wedge (P \vee \neg Q) \wedge (Q \vee P) \wedge (Q \vee \neg P).$

10. a. $\neg (A \wedge \neg B) \equiv \neg A \vee \neg \neg B \equiv \neg A \vee B \equiv A \rightarrow B.$

b. $(A \rightarrow B) \rightarrow (A \rightarrow \neg (B \vee A))$
$\equiv \neg (A \rightarrow B) \vee (A \rightarrow \neg (B \vee A))$
$\equiv (A \wedge \neg B) \vee (\neg A \vee \neg (B \vee A))$
$\equiv (A \wedge \neg B) \vee (\neg A \vee (\neg B \wedge \neg A))$
$\equiv (A \wedge \neg B) \vee \neg A$ (absorption)
$\equiv \neg B \vee \neg A$ (absorption)
$\equiv A \rightarrow \neg B.$

c. $(A \rightarrow B) \rightarrow (\neg (B \wedge C) \rightarrow \neg A)$
$\equiv \neg (A \rightarrow B) \vee (\neg (B \wedge C) \rightarrow \neg A)$
$\equiv (A \wedge \neg B) \vee (\neg (B \wedge C) \rightarrow \neg A)$
$\equiv (A \wedge \neg B) \vee (\neg \neg (B \wedge C) \vee \neg A)$
$\equiv (A \wedge \neg B) \vee (B \wedge C) \vee \neg A$
$\equiv ((A \wedge \neg B) \vee \neg A) \vee (B \wedge C)$
$\equiv ((A \vee \neg A) \wedge (\neg B \vee \neg A)) \vee (B \wedge C)$
$\equiv (\text{True} \wedge (\neg B \vee \neg A)) \vee (B \wedge C)$
$\equiv \neg B \vee \neg A \vee (B \wedge C)$
$\equiv \neg A \vee (\neg B \vee (B \wedge C))$
$\equiv \neg A \vee ((\neg B \vee B) \wedge (\neg B \vee C))$
$\equiv \neg A \vee (\text{True} \wedge (\neg B \vee C))$
$\equiv \neg A \vee \neg B \vee C.$

d. $(A \wedge B \rightarrow C) \rightarrow (A \wedge C \rightarrow B)$
$\equiv \neg (A \wedge B \rightarrow C) \vee (A \wedge C \rightarrow B)$
$\equiv (A \wedge B \wedge \neg C) \vee (A \wedge C \rightarrow B)$
$\equiv (A \wedge B \wedge \neg C) \vee \neg (A \wedge C) \vee B$
$\equiv (A \wedge B \wedge \neg C) \vee \neg A \vee \neg C \vee B$
$\equiv (A \vee \neg A \vee \neg C \vee B) \wedge (B \vee \neg A \vee \neg C \vee B) \wedge (\neg C \vee \neg A \vee \neg C \vee B)$
$\equiv \text{True} \wedge (B \vee \neg A \vee \neg C) \wedge (\neg C \vee \neg A \vee B)$
$\equiv B \vee \neg A \vee \neg C$
$\equiv B \vee \neg (A \wedge C)$
$\equiv A \wedge C \rightarrow B.$

6.3 Formal Reasoning

Learning Objectives

Be able to describe the proof rules used in propositional calculus.

Be able to use the proof rules to write formal proofs.

Review Questions

1. What is a proof (or derivation)?
2. What is the conjunction rule?
3. What is the simplification rule?
4. What is the addition rule?
5. What is the disjunctive syllogism rule?
6. What is the modus ponens rule?
7. What is the conditional proof rule?
8. What is the double negation rule?
9. What is the contradiction rule?
10. What is the indirect proof rule?
11. What is the modus tollens rule?
12. What is the cases rule?
13. What is the hypothetical syllogism rule?
14. What is the constructive dilemma rule?
15. What is the destructive dilemma rule?

Solved Problems

1. Give a formal proof for each of the following tautologies by using CP. Do not use IP and do not use *T*.

   a. $(A \vee B) \rightarrow (\neg B \rightarrow A)$.

   b. $A \wedge (B \rightarrow \neg A) \wedge (\neg B \rightarrow C) \rightarrow C$.

   c. $(\neg A \vee \neg B) \wedge (B \vee C) \wedge (\neg C \vee D) \rightarrow (A \rightarrow D)$.

2. Give a formal proof for each of the following tautologies by using CP and by using IP at least once in each proof. Do not use *T*.

   a. $(A \rightarrow B \wedge C) \wedge \neg B \rightarrow \neg A$.

   b. $(B \rightarrow A) \rightarrow (A \vee B \rightarrow A)$.

   c. $(A \rightarrow B) \rightarrow (\neg (B \wedge C) \rightarrow \neg (C \wedge A))$.

   d. $(\neg A \vee \neg B) \wedge (B \vee C) \wedge (\neg C \vee D) \rightarrow (A \rightarrow D)$.

Solutions

1. a.

1.	$A \vee B$	*P*
2.	$\neg B$	*P* [for $\neg B \rightarrow A$]
3.	A	1, 2, DS
4.	$\neg B \rightarrow A$	2, 3, CP
	QED	1– 4, CP.

   b.

1.	A	*P*
2.	$B \rightarrow \neg A$	*P*
3.	$\neg B \rightarrow C$	*P*
4.	$\neg \neg A$	1, DN
5.	$\neg B$	2, 4, MT
6.	C	3, 5, MP
	QED	1–6, CP

   c.

1.	$\neg A \vee \neg B$	*P*
2.	$B \vee C$	*P*
3.	$\neg C \vee D$	*P*
4.	A	*P* [for $A \rightarrow D$]
5.	$\neg B$	1, 4, DS
6.	C	2, 5, DS
7.	D	3, 6, DS
8.	$A \rightarrow D$	4–7, CP
	QED	1–3, 8, CP.

2. a.

1.	$A \rightarrow B \wedge C$	P
2.	$\neg B$	P
3.	$\neg \neg A$	P [for $\neg A$]
4.	A	3, DN
5.	$B \wedge C$	1, 4, MP
6.	B	5, Simp
7.	False	2, 6, Contr
8.	$\neg A$	3–7, IP
	QED	1, 2, 8, CP.

b.

1.	$B \rightarrow A$	P
2.	$A \vee B$	P [for $A \vee B \rightarrow A$]
3.	$\neg A$	P [for A]
4.	B	2, 3, DS
5.	A	1, 4, MP
6.	False	3, 6, Contr
7.	A	3–6, IP
8.	$A \vee B \rightarrow A$	2, 7, CP
	QED	1, 8, CP.

c.

1.	$A \rightarrow B$	P
2.	$\neg (B \wedge C)$	P [for $\neg (B \wedge C) \rightarrow \neg (C \wedge A)$]
3.	$\neg \neg (C \wedge A)$	P [for $\neg (C \wedge A)$]
4.	$C \wedge A$	3, DN
5.	A	4, Simp
6.	B	1, 5, MP
7.	C	4, Simp
8.	$B \wedge C$	6, 7, Conj
9.	False	9, Contr
10.	$\neg (C \wedge A)$	3–10, IP
11.	$\neg (B \wedge C) \rightarrow \neg (C \wedge A)$	2, 10, CP
	QED	1, 11, CP.

d.

1.	$\neg A \vee \neg B$	P
2.	$B \vee C$	P
3.	$\neg C \vee D$	P
4.	A	P [for $A \rightarrow D$]
5.	$\neg D$	P [for D]
6.	$\neg C$	3, 5, DS
7.	B	2, 6, DS
8.	$\neg \neg B$	7, DN
9.	$\neg A$	1, 8, DS
10.	False	4, 9, Contr
11.	D	5–10, IP
12.	$A \rightarrow D$	4, 11, CP
	QED	1–3, 12, CP.

6.4 Formal Axiom Systems

Learning Objectives

Be able to describe the Frege-Lukasiewicz axiom system.

Review Questions

1. What does it mean to say a formal reasoning system is sound?

2. What does it mean to say a formal reasoning system is complete?

3. How can you be sure that a system is sound?

4. What is an example of a small axiomatic system for the propositional calculus that is sound and complete?

Solved Problems

1. Suppose we are given the following three axioms, where $A \rightarrow B$ is an abbreviation for $\neg (A \wedge \neg B)$.

   Axiom 1: $A \rightarrow (A \wedge A)$.

   Axiom 2: $(A \wedge A) \rightarrow A$.

   Axiom 3: $(A \rightarrow B) \rightarrow (\neg (B \wedge C) \rightarrow \neg (C \wedge A))$.

   Use this axiom system (due to Rosser) to prove each of the following statements without using CP. You may use MP and any statements in the list to prove subsequent statements.

   a. If $A \rightarrow B$ and $B \rightarrow C$ are theorems, then $\neg (\neg C \wedge A)$ is a theorem.

   b. $\neg (\neg A \wedge A)$.

   c. $\neg \neg A \rightarrow A$.

   d. $\neg (A \wedge B) \rightarrow (B \rightarrow \neg A)$.

   e. $A \rightarrow \neg \neg A$.

   f. $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$.

Solutions

1. a.

1.	$A \rightarrow B$	P
2.	$B \rightarrow C$	P
3.	$(A \rightarrow B) \rightarrow (\neg (B \wedge \neg C) \rightarrow \neg (\neg C \wedge A))$	Axiom 3
4.	$\neg (B \wedge \neg C) \rightarrow \neg (\neg C \wedge A)$	1, 3, MP
5.	$\neg (B \wedge \neg C)$	2, Abbreviation
6.	$\neg (\neg C \wedge A)$	4, 5, MP

QED.

b.

1.	$A \rightarrow (A \wedge A)$	Axiom 1
2.	$(A \wedge A) \rightarrow A$	Axiom 2
3.	$\neg (\neg A \wedge A)$	1, 2, Part (a)

QED.

c.

1.	$\neg (\neg \neg A \wedge \neg A)$	Part (b)
2.	$\neg \neg A \rightarrow A$	1, Abbreviation

QED.

d.

1.	$\neg \neg A \rightarrow A$	Part (c)
2.	$(\neg \neg A \rightarrow A) \rightarrow (\neg (A \wedge B) \rightarrow \neg (B \wedge \neg \neg A))$	Axiom 3
3.	$\neg (A \wedge B) \rightarrow \neg (B \wedge \neg \neg A)$	1, 2, MP
4.	$\neg (A \wedge B) \rightarrow (B \rightarrow \neg A)$	3, Abbreviation

QED.

e.

1.	$\neg (\neg A \wedge A)$	Part (b)
2.	$\neg (\neg A \wedge A) \rightarrow (A \rightarrow \neg \neg A)$	Part (d)
3.	$A \rightarrow \neg \neg A$	1, 2, MP

QED.

f.

1.	$\neg (A \wedge \neg B) \rightarrow (\neg B \rightarrow \neg A)$	Part (d)
2.	$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$	1, Abbreviation

QED.

Chapter 7

Predicate Logic

7.1 First-Order Predicate Calculus

Learning Objectives

Be able to determine whether a wff is valid, invalid, satisfiable, or unsatisfiable.

Review Questions

1. What is a predicate?

2. What is an atom?

3. What is the meaning of each of the following symbols or expressions?

   a. $\exists x$.

   b. $\forall x$.

   c. $W(x/t)$.

   d. x/t .

   e. $W(x)$.

4. What is a wff?

5. What is the scope of a quantifier?

6. What is a bound variable?

7. What is a free variable?

8. What is an interpretation?

9. What is a model?

10. What is a countermodel?

11. What does valid mean?

12. What does invalid mean?

13. What does satisfiable mean?

14. What does unsatisfiable mean?

Solved Problems

1. Write down the propositional wff obtained by interpreting the following wff over the domain $\{a, b\}$.

$$\exists x\, \forall y\, (p(x) \rightarrow q(y)).$$

2. Each of the following expressions represents an interpretation of a wff over the domain $\mathbb{N}$. In each case, write down the original wff.

   a. $p(0, 0) \wedge p(1, 0) \wedge p(2, 0) \wedge \ldots$.

   b. $p(0, 0) \vee p(0, 1) \vee p(0, 2) \vee \ldots$.

3. Find examples of wffs with the given properties.

   a. The variable x has three bound occurrences and one free occurrence.

   b. The variable x has four bound occurrences and the variable y has two bound occurrences and one free occurrence.

4. Find a countermodel for the following wff.

$$[\forall x\, \forall y\, A(x, y) \rightarrow \forall x\, \forall y\, B(x, y)] \rightarrow \forall x\, \forall y\, (A(x, y) \rightarrow B(x, y)).$$

5. Give an informal argument to show that the following wff is unsatisfiable.

$$\exists x\, \forall y\, (\neg p(x, x) \wedge p(x, y)).$$

6. Given the wff

$$\forall x\, \exists y\, (p(f(x, y), y) \wedge \neg p(y, x)).$$

   a. Find a model for the wff.

   b. Find a countermodel for the wff.

7. Let W denote the following wff.

$$\forall x\, (p(x, x) \rightarrow \exists y\, p(x, y)).$$

   a. Give a direct proof that W is valid.

   b. Give an indirect proof that W is valid.

Solutions

1. $[(p(a) \rightarrow q(a)) \wedge (p(a) \rightarrow q(b)] \vee [(p(b) \rightarrow q(a) \wedge (p(b) \rightarrow q(b))]$.

2. a. $\forall x\, p(x, a)$. b. $\exists x\, p(a, x)$.

3. a. For example, $\exists x\, p(x, x) \rightarrow q(x)$.

   b. For example, $\exists x\, \forall y\, (p(x, x) \rightarrow q(x, y)) \wedge r(y)$.

4. Let the domain be $D = \{a, b\}$. Let $A(a, a)$ = True and $A(a, b) = A(b, a) = A(b, b)$ = False. Let $B(x, y)$ = False for all $x, y \in D$. Then $\forall x\, \forall y\, A(x, y)$ is false, $\forall x\, \forall y\, B(x, y)$ is false, and since $A(a, a) \rightarrow B(a, a)$ is false, it follows that $\forall x\, \forall y\, (A(x, y) \rightarrow B(x, y))$ is false. So the interpreted wff becomes

   $$(\text{False} \rightarrow \text{False}) \rightarrow \text{False} \equiv \text{True} \rightarrow \text{False} \equiv \text{False}.$$

5. Assume, by way of contradiction, that the wff is satisfiable. Then there is some model for the wff, say with domain D. This implies that there is an element $d \in D$ such that $\forall y\, (\neg\, p(d, d) \wedge p(d, y))$ is true. But we know that $\neg\, p(d, d) \wedge p(d, d)$ is false. So $\forall y\, (\neg\, p(d, d) \wedge p(d, y))$ must be false. This contradiction tells us that the original wff is unsatisfiable.

6. a. For example, let $D = \{a, b, c\}$, let $f(x, y) = x$ for all $x, y \in D$, let $p(a, b) = p(b, c) = p(c, a)$ = True, let $p(b, a) = p(c, b) = p(a, c)$ = False, and let $p(a, a)$, $p(b, b)$, and $p(c, c)$ have arbitrary truth values.

   b. For example, over any domain D, and any definition of f, let $p(x, y)$ = True for all $x, y \in D$.

7. a. Let I be an interpretation with domain D for W. Let $x \in D$ and consider the wff $p(x, x) \rightarrow \exists y\, p(x, y)$. If $p(x, x)$ is false, then this wff is vacuously true. If $p(x, x)$ is true, then $\exists y\, p(x, y)$ is also true because we can pick $y = x$. So the wff $p(x, x) \rightarrow \exists y\, p(x, y)$ is true. Since x was an arbitrary element of D, it follows $p(x, x) \rightarrow \exists y\, p(x, y)$ for all $x \in D$. Therefore W is true for I. Thus I is a model for W. Since I was arbitrary, it follows that every interpretation of W is a model. So W is valid.

   b. Assume, by way of contradiction, that W is invalid. Then there is an interpretation I with domain D such that I is a countermodel for W. So there is some $d \in D$ such that $p(d, d) \rightarrow \exists y\, p(d, y)$ is false. This means that $p(d, d)$ is true and $\exists y\, p(d, y)$ is false. But the fact that $p(d, d)$ is true implies that $\exists y\, p(d, y)$ is true by letting $y = d$. So we have a contradiction, namely that $\exists y\, p(d, y)$ is true and false. Therefore W is valid.

7.2 Equivalent Formulas

Learning Objectives

Be able to construct equivalence proofs and transform first-order wffs into prenex conjunctive or disjunctive normal form.

Be able to transform simple English sentences into formal logic.

Review Questions

1. What does it mean to say two wffs are equivalent?

2. What is the universal closure of a wff?

3. What is the existential closure of a wff?

4. What is a literal?

5. What is renaming?

6. What is a prefix normal form?

Solved Problems

1. Find a prenex conjunctive normal form for the following wff.

$$\forall x\, \exists y\, (p(x, y) \vee q(x, y)) \rightarrow \exists y\, \forall x\, (p(x, y) \rightarrow q(x, y)).$$

2. Find a prenex disjunctive normal form for the following wff.

$$\forall y\, [\exists x\, (p(x, y) \rightarrow \exists x\, q(x)) \rightarrow \exists x\, (p(x, y) \wedge q(x))].$$

3. Let $s(x)$ mean x is a senator, let $p(x)$ mean x is a politician, and let $e(x)$ mean x is ethical. Find a wff to describe each sentence over the domain of people.
   a. Some senators are ethical.
   b. Not all senators are ethical.
   c. All senators are politicians.
   d. No politician is a senator.

4. Let $c(x)$ mean x is a criminal, let $r(x)$ mean x is rich, and let $s(x)$ mean x is sane. Find a wff to describe each sentence over the domain of people.

   a. Every criminal is rich.

   b. Some criminals are sane.

   c. Some criminals are neither rich nor sane.

   d. Not all criminals are sane.

5. Let $g(x, y, z)$ mean that z is the greatest common divisor of x and y, and let $l(x, y, z)$ mean that z is the least common multiple of x and y. Let $p(x)$ mean x is positive. Formalize the following statement.

   Any pair of positive natural numbers has a greatest common divisor and a least common multiple.

Solutions

1. First, rename quantifier variables to have distinct names:

$$\forall x\, \exists y\, (p(x, y) \vee q(x, y)) \rightarrow \exists u\, \forall v\, (p(v, u) \rightarrow q(v, u)).$$

   Next, remove implications:

$$\neg\, \forall x\, \exists y\, (p(x, y) \vee q(x, y)) \vee \exists u\, \forall v\, (\neg\, p(v, u) \vee q(v, u)).$$

   Next, move negations to the right;

$$\exists x\, \forall y\, (\neg\, p(x, y) \wedge \neg\, q(x, y)) \vee \exists u\, \forall v\, (\neg\, p(v, u) \vee q(v, u)).$$

   Next, move quantifiers to the left:

$$\exists x\, \forall y\, \exists u\, \forall v\, [(\neg\, p(x, y) \wedge \neg\, q(x, y)) \vee (\neg\, p(v, u) \vee q(v, u))].$$

   Next, distribute $\vee$ over $\wedge$ to obtain a prenex conjunctive normal form:

$$\exists x\, \forall y\, \exists u\, \forall v\, [(\neg\, p(x, y) \vee \neg\, p(v, u) \vee q(v, u))$$
$$\wedge\, (\neg\, q(x, y) \vee \neg\, p(v, u) \vee q(v, u))].$$

2. First, rename quantifier variables to have distinct names:

$$\forall y\, [\exists x\, (p(x, y) \rightarrow \exists u\, q(u)) \rightarrow \exists v\, (p(v, y) \wedge q(v))].$$

   Next, remove implications:

$$\forall y\, [\neg\, \exists x\, (\neg\, p(x, y) \vee \exists u\, q(u)) \vee \exists v\, (p(v, y) \wedge q(v))].$$

   Next, move negations to the right:

$$\forall y\, [\forall x\, (p(x, y) \wedge \forall u\, \neg\, q(u)) \vee \exists v\, (p(v, y) \wedge q(v))].$$

   Next, move quantifiers to the left:

$$\forall y\, \forall x\, \forall u\, \exists v\, [(p(x, y) \wedge \neg\, q(u)) \vee (p(v, y) \wedge q(v))].$$

   The wff is in the desired prenex disjunctive normal form:

3. a. $\exists x\,(s(x) \wedge e(x))$.
   b. $\neg\,\forall x\,(s(x) \rightarrow e(x)) \equiv \exists x\,(s(x) \wedge \neg\, e(x))$.
   c. $\forall x\,(s(x) \rightarrow p(x))$.
   d. $\forall x\,(p(x) \rightarrow \neg\, s(x)) \equiv \neg\,\exists x\,(p(x) \wedge s(x))$.
4. a. $\forall x\,(c(x) \rightarrow r(x))$.
   b. $\exists x\,(c(x) \wedge s(x))$.
   c. $\exists x\,(c(x) \wedge \neg\, r(x) \wedge \neg\, s(x))$.
   d. $\neg\,\forall x\,(c(x) \rightarrow s(x)) \equiv \exists x\,(c(x) \wedge \neg\, s(x))$.
5. $\forall x\,\forall y\,(p(x) \wedge p(y) \rightarrow \exists u\,\exists v\,(g(x, y, u) \wedge l(x, y, v)))$.

7.3 Formal Proofs in Predicate Calculus

Learning Objectives

Be able to describe the proof rules for quantifiers.

Be able to use the quantifier rules along with the basic proof rules to write formal proofs in first-order predicate calculus.

Review Questions

1. What is the universal instantiation rule?

2. What is the existential generalization rule?

3. What is the existential instantiation rule?

4. What is the universal generalization rule?

Solved Problems

1. Quantifiers cannot be removed and replaced all the time without restrictions. Demonstrate this by constructing an incorrect proof for each of the following statements by applying a quantifier proof rule without regard to the restrictions on its use.
   a. $p(x) \rightarrow \forall x\, p(x)$.
   b. $\forall x\,\exists y\, p(x, y) \rightarrow \exists y\,\forall x\, p(x, y)$.
   c. $\exists x\, p(x) \rightarrow \forall x\, p(x)$.
   d. $\exists x\,\exists y\, p(x, y) \rightarrow \exists x\, p(x, x)$.
   e. $\forall x\,\exists y\, p(x, y) \rightarrow \exists y\, p(y, y)$.

2. Give a formal proof that the following wff is valid.

$$\forall x\, \forall y\, (A(x, y) \rightarrow B(x, y)) \rightarrow [\forall x\, \forall y\, A(x, y) \rightarrow \forall x\, \forall y\, B(x, y)].$$

3. Give a formal proof that the following wff is valid.

$$[\forall x\, (A(x) \wedge B(x) \rightarrow C(x))] \rightarrow [\forall x\, A(x) \wedge \forall x\, B(x) \rightarrow \forall x\, C(x)].$$

4. Give a formal direct proof that the following wff is valid.

$$\exists x\, (p(x) \vee q(x)) \rightarrow \exists x\, p(x) \vee \exists x\, q(x).$$

Hint: The constructive dilemma proof rule (CD) may come in handy.

5. Give a formal direct proof that the following wff is valid.

$$\exists x\, p(x) \vee \exists x\, q(x) \rightarrow \exists x\, (p(x) \vee q(x)).$$

Hint: The constructive dilemma proof rule (CD) may come in handy.

Solutions

1. a.

1.	$p(x)$	P
2.	$\forall x\, p(x)$	1, UG

This proof is incorrect because x is free in the premise on line 1. Therefore line 2 cannot be inferred from line 1 by UG.

b.

1.	$\forall x\, \exists y\, p(x, y)$	P
2.	$\exists y\, p(x, y)$	1, UI
3.	$p(x, c)$	2, EI
4.	$\forall x\, p(x, c)$	3, UG
5.	$\exists y\, \forall x\, p(x, y)$	4, EG

This proof is incorrect because x is free in $p(x, c)$ on line 3, which is inferred by EI. Therefore line 4 cannot be inferred from line 3 by UG.

c.

1.	$\exists x\, p(x)$	P
2.	$p(x)$	1, EI
3.	$\forall x\, p(x)$	2, UG

This proof is incorrect because $p(x)$ on line 2 is inferred from $\exists x\, p(x)$ by EI, which can only create constants.

d.

1.	$\exists x\, \exists y\, p(x, y)$	P
2.	$\exists y\, p(c, y)$	1, EI
3.	$p(c, c)$	2, EI
4.	$\exists x\, p(x, x)$	3, EG

This proof is incorrect because $p(c, c)$ on line 3 is inferred from the wff $\exists y\, p(c, y)$ by EI, which must create a new constant other than c.

e. 1. $\forall x\, \exists y\, p(x, y)$ *P*
2. $\exists y\, p(y, y)$ 1, UI

This proof is incorrect because y is not free to replace x on line 1. Therefore UI cannot be used to replace x by y on line 2.

2.

1.	$\forall x\, \forall y\, (A(x, y) \rightarrow B(x, y))$	*P*
2.	$A(x, y) \rightarrow B(x, y)$	1, UI, UI
3.	$\forall x\, \forall y\, A(x, y)$	*P* [for $\forall x\, \forall y\, A(x, y) \rightarrow \forall x\, \forall y\, B(x, y)$]
4.	$A(x, y)$	3, UI, UI
5.	$B(x, y)$	2, 4, MP
6.	$\forall x\, \forall y\, B(x, y)$	5, UG, UG
7.	$\forall x\, \forall y\, A(x, y) \rightarrow \forall x\, \forall y\, B(x, y)$	3–6, CP
	QED	1, 2, 7, CP.

3.

1.	$\forall x\, (A(x) \wedge B(x) \rightarrow C(x))$	*P*
2.	$A(x) \wedge B(x) \rightarrow C(x)$	1, UI
3.	$\forall x\, A(x) \wedge \forall x\, B(x)$	*P* [for $\forall x\, A(x) \wedge \forall x\, B(x) \rightarrow \forall x\, C(x)$]
4.	$\forall x\, A(x)$	2, Simp
5.	$A(x)$	4, UI
6.	$\forall x\, B(x)$	2, Simp
7.	$B(x)$	6, UI
8.	$A(x) \wedge B(x)$	5, 7, Conj
9.	$C(x)$	2, 8, MP
10.	$\forall x\, C(x)$	9, UG
11.	$\forall x\, A(x) \wedge \forall x\, B(x) \rightarrow \forall x\, C(x)$	3–10, CP
	QED	1, 2, 11, CP.

4.

1.	$\exists x\, (p(x) \vee q(x))$	*P*
2.	$p(c) \vee q(c)$	1, EI
3.	$p(c)$	*P*[for $p(c) \rightarrow \exists x\, p(x)$]
4.	$\exists x\, p(x)$	3, EG
5.	$p(c) \rightarrow \exists x\, p(x)$	3, 4, CP
6.	$q(c)$	*P* [for $q(c) \rightarrow \exists x\, q(x)$]
7.	$\exists x\, q(x)$	6, EG
8.	$q(c) \rightarrow \exists x\, q(x)$	6, 7, CP
9.	$\exists x\, p(x) \vee \exists x\, q(x)$	2, 5, 8, CD
	QED	1, 2, 5, 8, 9, CP.

5.	1.	$\exists x\, p(x) \vee \exists x\, q(x)$	*P*
	2.	$\exists x\, p(x)$	*P* [for $\exists x\, p(x) \rightarrow \exists x\, (p(x) \vee q(x))$]
	3.	$p(c)$	2, EI
	4.	$p(c) \vee q(c)$	3, Add
	5.	$\exists x\, (p(x) \vee q(x))$	4, EG
	6.	$\exists x\, p(x) \rightarrow \exists x\, (p(x) \vee q(x))$	2–5, CP
	7.	$\exists x\, q(x)$	*P* [for $\exists x\, q(x) \rightarrow \exists x\, (p(x) \vee q(x))$]
	8.	$q(c)$	7, EI
	9.	$p(c) \vee q(c)$	8, Add
	10.	$\exists x\, (p(x) \vee q(x))$	9, EG
	11.	$\exists x\, q(x) \rightarrow \exists x\, (p(x) \vee q(x))$	7–10, CP
	12.	$\exists x\, (p(x) \vee q(x))$	1, 6, 11, Cases
		QED	1, 6, 11, 12, CP.

Chapter 8

Applied Logic

8.1 Equality

Learning Objectives

Be able to write formal proofs in first-order predicate calculus with equality.

Review Questions

1. What is a first-order theory with equality?

2. What is the equality axiom?

3. What is EE rule for predicates?

4. What is EE rule for functions?

5. What is the general EE rule?

Solved Problems

1. Give a formal proof that the following wff is valid in a first-order theory with equality.

$$\exists x\,((x = a[k]) \wedge (a[x] = t)) \rightarrow (a[a[k]] = t).$$

2. Give a formal proof that the following wff is valid in a first-order theory with equality.

$$(a[a[k]] = a[b[j]]) \rightarrow \exists x\,\exists y\,((x = a[k]) \wedge (y = b[j]) \wedge (a[x] = a[y])).$$

3. Prove that $p(x, y) \equiv \forall u\,\forall v\,((x = u) \wedge (y = v) \rightarrow p(u, v))$ by giving formal proofs that the following two wffs are valid in a first-order theory with equality.

   a. $p(x, y) \rightarrow \forall u\,\forall v\,((x = u) \wedge (y = v) \rightarrow p(u, v))$

   b. $\forall u\,\forall v\,((x = u) \wedge (y = v) \rightarrow p(u, v)) \rightarrow p(x, y)$.

4. Formalize the definition of each of the following predicates in terms of properties of integers.

   a. lcm(x, y, m) means m is the least common multiple of x and y.

   b. prime(x) means x is a prime number.

Solutions

1. | | | |
   |---|---|---|
   | 1. | $\exists x\ ((x = a[k]) \wedge (a[x] = t))$ | P |
   | 2. | $(c = a[k]) \wedge (a[c] = t)$ | 1, EI |
   | 3. | $a[a[k]] = t$ | 2, EE |
   | | QED | 1–3, CP. |

2. | | | |
   |---|---|---|
   | 1. | $a[a[k]] = a[b[j]]$ | P |
   | 2. | $\quad \forall x\ \forall y((x \neq a[k]) \vee (y \neq b[j]) \vee (a[x] \neq a[y]))$ | P [for IP] |
   | 3. | $\quad (a[k] \neq a[k]) \vee (b[j] \neq b[j]) \vee (a[a[k]] \neq a[b[j]])$ | 2, UI, UI |
   | 4. | $\quad a[k] = a[k]$ | EA |
   | 5. | $\quad b[j] = b[j]$ | EA |
   | 6. | $\quad a[a[k]] \neq a[b[j]]$ | 3, 4, 5, DS, DS |
   | 7. | $\quad$ False | 1, 6, Contr |
   | 8. | $\exists x\ \exists y\ ((x = a[k]) \wedge (y = b[j]) \wedge (a[x] = a[y]))$ | 2–7, IP |
   | | QED | 1, 8, CP. |

3. a. | | | |
      |---|---|---|
      | 1. | $p(x, y)$ | P |
      | 2. | $\quad (x = u) \wedge (y = v)$ | P [for $(x = u) \wedge (y = v) \rightarrow p(u, v)$] |
      | 3. | $\quad p(u, v)$ | 1, 2, EE |
      | 4. | $(x = u) \wedge (y = v) \rightarrow p(u, v)$ | 2, 3, CP |
      | 5. | $\forall u\ \forall v\ ((x = u) \wedge (y = v) \rightarrow p(u, v))$ | 4, UG, UG |
      | | QED | 1, 4, 5, CP. |

   b. | | | |
      |---|---|---|
      | 1. | $\forall u\ \forall v\ ((x = u) \wedge (y = v) \rightarrow p(u, v))$ | P |
      | 2. | $(x = x) \wedge (y = y) \rightarrow p(x, y)$ | 1, UI, UI |
      | 3. | $(x = x) \wedge (y = y)$ | EA, EA, Conj |
      | 4. | $p(x, y)$ | 2, 3, MP |
      | | QED | 1–4, CP. |

4. a. $\text{lcm}(x, y, m) = \text{div}(x, m) \wedge \text{div}(y, m) \wedge \forall u\ (\text{div}(x, u) \wedge \text{div}(y, u) \rightarrow \text{div}(m, u))$,

   where $\text{div}(u, v) = (u \neq 0) \wedge \exists w\ (v = uw)$.

   b. $\text{prime}(x) = (x > 1) \wedge \forall u\ (\text{div}(u, x) \rightarrow (u = 1) \vee (u = x))$,

   where $\text{div}(u, v) = (u \neq 0) \wedge \exists w\ (v = uw)$.

8.2 Program Correctness

Learning Objectives

Be able to construct partial correctness proofs for elementary imperative programs.

Be able to construct termination proofs for simple loops.

Review Questions

1. What is the meaning of the expression $\{P\}\ S\ \{Q\}$?
2. What is a Hoare triple?
3. What does it mean to say that $\{P\}\ S\ \{Q\}$ is correct?
4. What is the AA axiom?
5. What is the composition rule?
6. What is the consequence rule?
7. What is the if-then rule?
8. What is the if-then-else rule?
9. What is the while rule?
10. What is a loop invariant?
11. What is a precondition?
12. What is a postcondition?
13. What is the AAA axiom?
14. What are the steps to show that a while-loop terminates?

Solved Problems

1. Prove the correctness of the following wff.

$$\{(a > 0) \wedge (b > 0)\}\ x := a - 2; y := b - 2\ \{(x - y) = (a - b)\}.$$

2. Prove the correctness of the following wff.

$$\{\exists x\ (y = 2x + 1)\}\ y := y + 1\ \{\exists x\ (y = 2x)\}.$$

3. Prove the correctness of the following wff, where $q(x)$ means x is an integer and odd and even are tests for the oddness and eveness of an integer.

$\{q(x) \wedge q(y)\}$
if even$(x - y)$ **then** $x := x + 1$ **else** $y := y - 2$ **fi**
$\{\text{odd}(x - y)\}$.

4. Given the following wff, where $p(x)$ means that x is a positive integer:

$\{p(x) \wedge p(y)\}$
while $x > y$ **do** $x := x - y$ **od**
$\{p(x) \wedge p(y) \wedge (x \leq y)\}$

Prove that the program is correct for the loop invariant $p(x) \wedge p(y)$.

5. Prove the correctness of the following program with respect to the loop invariant

$$P = (m = 2^n) \wedge (n \leq k),$$

where we assume that all variables take integer values.

while $n < k$ **do** $m := 2 * m$; $n := n + 1$ **od**

6. Complete the following partial wff by applying the array assignment axiom to find the precondition.

$$\{\quad\}\ a[i\,] := 39\ \{a[j] = 39\}.$$

7. Prove the correctness of the following wff.

$$\{(i = j) \wedge (a[i] = 2)\}\ a[j] := 5\ \{a[i] = 5\}.$$

8. Prove the correctness of the following wff.

$$\{a[i] < a[j]\} \quad x := a[i];\ a[i] := a[j];\ a[j] := x \quad \{a[j] < a[i]\}.$$

9. Given the following wff, where $q(x)$ means that x is a rational number:

$\{q(x) \wedge q(y)\}$
while $x + y > 0$ **do** $x := x + 1$; $y := y - 2$ **od**
$\{q(x) \wedge q(y) \wedge (x + y \leq 0)\}$

Prove that the loop terminates with respect to $q(x) \wedge q(y)$.

10. Given the following wff, where $p(x)$ means that x is a positive integer:

    $\{p(x) \wedge p(y)\}$
    while $x > y$ **do** $x := x - y$ **od**
    $\{p(x) \wedge p(y) \wedge (x \leq y)\}$

    For each of the following cases, decide whether a termination proof exists for the program that uses $f(x, y)$ and the well-founded set $\mathbb{N}$.

    a. $f(x, y) = x + y$. Yes No

    b. $f(x, y) = x - y$. Yes No

    c. $f(x, y) = x$. Yes No

    d. $f(x, y) = y$. Yes No

    e. $f(x, y) = \min(x, y)$. Yes No

    f. $f(x, y) = \max(x, y)$. Yes No

Solutions

1. 1. $\{(x - b + 2) = (a - b)\}\ y := b - 2\ \{(x - y) = (a - b)\}$ AA
   2. $\{(a - 2 - b + 2) = (a - b)\}\ x := a - 2\ \{(x - b + 2) = (a - b)\}$ AA
   3. $\{(a - 2 - b + 2) = (a - b)\}$
      $x := a - 2; y := b - 2$
      $\{(x - y) = (a - b)\}$ 1, 2, Comp
   4. $(a > 0) \wedge (b > 0) \rightarrow (a - 2 - b + 2) = (a - b)$ *T* (trivially)

   QED 3, 4, Consequence.

2. 1. $\{\exists x\ (y + 1 = 2x)\}\ y := y + 1\ \{\exists x\ (y = 2x)\}$ AA
   2. $\exists x\ (y = 2x + 1)$ *P* [for CP]
   3. $y = 2c + 1$ 2, EI
   4. $y + 1 = 2c + 1 + 1$ 3, EE
   5. $y + 1 = 2(c + 1)$ 4, *T*
   6. $\exists x\ (y + 1 = 2x)$ 3, EG
   7. $\exists x\ (y = 2x + 1) \rightarrow \exists x\ (y + 1 = 2x)$ 2–6, CP
   8. $\{\exists x\ (y = 2x + 1)\}\ y := y + 1\ \{\exists x\ (y = 2x)\}$ 1, 7, Consequence

   QED.

3. The if-then-else rule requires two proofs.

   (First proof)

   1. $\{\text{odd}(x + 1 - y)\}\ x := x + 1\ \{\text{odd}(x - y)\}$ AA
   2. $q(x) \wedge q(y) \wedge \text{even}(x - y)$ *P* [for CP]
   3. $\text{even}(x - y)$ 2, Simp
   4. $\text{odd}(x - y + 1)$ 3, *T*
   5. $q(x) \wedge q(y) \wedge \text{even}(x - y) \rightarrow \text{odd}(x + 1 - y)$ 2–4, CP
   6. $\{q(x) \wedge q(y) \wedge \text{even}(x - y)\}\ x := x + 1\ \{\text{odd}(x - y)\}$ 1, 5, Consequence

(Second proof)

7.	$\{\text{odd}(x - y + 2)\}\ y := y - 2\ \{\text{odd}(x - y)\}$	AA
8.	$q(x) \wedge q(y) \wedge \neg\ \text{even}(x - y)$	*P* [for CP]
9.	$\neg\ \text{even}(x - y)$	8, Simp
10.	$\text{odd}(x - y)$	9, *T*
11.	$\text{odd}(x - y + 2)$	10, *T*
12.	$q(x) \wedge q(y) \wedge \neg\ \text{even}(x - y) \rightarrow \text{odd}(x - y + 2)$	8–11, CP
13.	$\{q(x) \wedge q(y) \wedge \text{even}(x - y)\}\ y := y - 2\ \{\text{odd}(x - y)\}$	7, 5, Consequence
	QED	1, 13, If-then-else.

4. According to the while-rule, we need to prove the following statement:

$$\{p(x) \wedge p(y) \wedge (x > y)\}\ x := x - y\ \{p(x) \wedge p(y)\}.$$

1.	$\{p(x - y) \wedge p(y)\}\ x := x - y\ \{p(x) \wedge p(y)\}$	AA
2.	$p(x) \wedge p(y) \wedge (x > y)$	*P* [for CP]
3.	$x > y$	2, Simp
4.	$x - y > 0$	3, *T*
5.	$p(x)$	2, Simp
6.	$p(y)$	2, Simp
7.	$p(x - y)$	4, 5, 6, *T*
8.	$p(x - y) \wedge p(y)$	6, 7, Conj
9.	$p(x) \wedge p(y) \wedge (x > y) \rightarrow p(x - y) \wedge p(y)$	2–8, CP
10.	$\{p(x) \wedge p(y) \wedge (x > y)\}\ x := x - y\ \{p(x) \wedge p(y)\}$	1, 9, Consequence
	QED	10, While-rule.

5. By the while-rule, it suffices to prove the following statement:

$$\{P \wedge (n < k)\}\ m := 2 * m;\ n := n + 1\ \{P\}.$$

1.	$\{(m = 2^{n+1}) \wedge (n + 1 \leq k)\}\ n := n + 1\ \{(m = 2^n) \wedge (n \leq k)\}$	AA
2.	$\{(2m = 2^{n+1}) \wedge (n + 1 \leq k)\}\ m := 2 * m\ \{(m = 2^{n+1}) \wedge (n + 1 \leq k)\}$	AA
3.	$(m = 2^n) \wedge (n \leq k) \wedge (n < k)$	*P* [for CP]
4.	$m = 2^n$	3, Simp
5.	$2m = 2^{n+1}$	4, EE
6.	$n < k$	3, Simp
7.	$n + 1 \leq k$	6, *T*
8.	$(2m = 2^{n+1}) \wedge (n + 1 \leq k)$	5, 7, Conj
9.	$(m = 2^n) \wedge (n \leq k) \wedge (n < k) \rightarrow (2m = 2^{n+1}) \wedge (n + 1 \leq k)$	3–8, CP
10.	$\{P \wedge (n < k)\}\ m := 2 * m;\ n := n + 1\ \{P\}$	1, 2, Comp, 9, Consequence
	QED	10, While-rule.

6. $\{(\text{if } j = i \text{ then } 39 \text{ else } a[j]) = 39\}\ a[i\ ] := 39\ \{a[j] = 39\}$.

7. 1. $\{(\text{if } i = j \text{ then } 5 \text{ else } a[i]) = 5\}\ a[j] := 5\ \{a[i] = 5\}$ AAA
   2. $(i = j) \wedge (a[i] = 2)$ *P* [for CP]
   3. $i = j$ 2, Simp
   4. $(i \neq j) \rightarrow (a[i] = 5)$ 3, *T* (vacuously)
   5. $(i = j) \rightarrow (5 = 5)$ *T* (trivially)
   6. $(\text{if } i = j \text{ then } 5 \text{ else } a[i]) = 5$ 4, 5, Conj, *T*
   7. $(i = j) \wedge (a[i] = 2) \rightarrow (\text{if } i = j \text{ then } 5 \text{ else } a[i]) = 5$ 2–6, CP

   QED 1, 7, Consequence.

8. 1. $\{x < (\text{if } i = j \text{ then } x \text{ else } a[i])\}\ a[j] := x\ \{a[j] < a[i]\}$ AAA
   2. $\{x < (\text{if } i = j \text{ then } x \text{ else } a[j])\}$
      $a[i] := a[j]$
      $\{x < (\text{if } i = j \text{ then } x \text{ else } a[i])\}$ AAA
   3. $\{a[i] < (\text{if } i = j \text{ then } a[i] \text{ else } a[j])\}$
      $x := a[i]$
      $\{x < (\text{if } i = j \text{ then } x \text{ else } a[j])\}$ AAA
   4. $a[i] < a[j]$ *P* [for CP]
   5. $(i \neq j) \rightarrow a[i] < a[j]$ 4, *T* (trivially)
   6. $(i = j) \rightarrow a[i] < a[i]$ 4, EE
   7. $a[i] < (\text{if } i = j \text{ then } a[i] \text{ else } a[j])$ 5, 6, Conj, *T*
   8. $(a[i] < a[j]) \rightarrow (a[i] < (\text{if } i = j \text{ then } a[i] \text{ else } a[j]))$ 4–7, CP

   QED 1, 2, Comp, 3, Comp, 8, Consequence.

9. For a well-founded set we can choose $\mathbb{N}$ with the usual ordering. For the program state (x, y) let $f(x, y) = \text{ceiling}(x + y)$. If $s = (x, y)$ is the state before the execution of the loop's body and t is the state after the execution of the loop's body, then $t = (x + 1, y - 2)$. Therefore we have:

   $f(s) = f(x, y) = \text{ceiling}(x + y)$ and $f(t) = f(x + 1, y - 2) = \text{ceiling}(x + y - 1)$.

   To prove termination, we must show that if $q(x) \wedge q(y)$ is true (i.e., x and y are rational) and $x + y > 0$, then

   $f(s), f(t) \in \mathbb{N}$ and $f(s) > f(t)$.

   Since $x + y > 0$, it follows that $\text{ceiling}(x + y) \geq 0$. So $f(s) \in \mathbb{N}$. Since $x + y > 0$, it follows that $x + y - 1 > -1$. So $\text{ceiling}(x + y - 1) \geq 0$. So $f(t) \in \mathbb{N}$. Since $x + y$ and $x + y - 1$ differ by 1, it follows that $\text{ceiling}(x + y) > \text{ceiling}(x + y - 1)$. In other words, we have $f(s) > f(t)$. Therefore the program terminates with respect to $q(x) \wedge q(y)$.

10. a. Yes.

    b. No, because $f(t) = x - 2y$, which may be negative.

    c. Yes.

    d. No, because $f(s) = f(t)$.

    e. No, because it might be the case that $f(s) = f(t)$. For example, if $s = (5, 1)$, then $t = (4, 1)$ so that $f(s) = f(t) = 1$.

    f. Yes.

8.3 Higher-Order Logics

Learning Objectives

Be able to describe higher order logics.

Be able to transform simple English sentences into higher-order logic.

Review Questions

1. In what way is a predicate a set?

2. What is the order of a predicate?

3. What is the order of a quantifier?

4. What is the order of a wff?

5. How is a higher-order wff given a meaning?

6. What is second-order logic?

Solved Problems

1. Write down the minimal order of logic to which each of the following wffs belong.
   a. $\forall y\ (p(x, y) \rightarrow \exists L\ (L(x) \wedge q(y, L)))$.
   b. $\forall x\ \exists L\ \forall M\ (L(x) \rightarrow M(L))$.
   c. $\forall A\ \exists L\ \exists M\ (L(A) \wedge \forall x\ \exists B\ (M(B, x) \rightarrow B(L)))$.
2. Write a wff in higher-order logic to formalize each of the following equality ideas.
   a. There are two sets A and B that are not equal.
   b. There are two elements x and y are not equal.
3. Write down a wff in higher-order logic without using equality that formalizes the following statement about lines of latitude on a globe: "All distinct lines are parallel."
4. Write down a wff in higher-order logic that formalizes the following statement about great circles on a globe: "All distinct circles meet in exactly two points."
5. Write down a wff in higher-order logic that formalizes the following statement from geometry.

   "For every line L and every point x not on L there is a circle C that passes through x and C touches L in exactly one point y."

6. Do a truth analysis for the following wff. In other words, find out which of the properties {valid, invalid, satisfiable, unsatisfiable} the wff satisfies.

$$\forall p\ \exists x\ p(x).$$

7. Do a truth analysis for the following wff. In other words, find out which of the properties {valid, invalid, satisfiable, unsatisfiable} the wff satisfies.

$$\exists p\ \forall q\ \forall x\ (p(x) \rightarrow q(x)).$$

8. Give a formal prove that the following wff is valid.

$$\exists x\ \exists y\ \forall P\ P(x, y) \rightarrow \forall P\ \exists x\ \exists y\ P(x, y).$$

9. Here's an example of higher order reasoning based on Euclidean geometry. Consider the following claim:

   If there is a point on a line, then there is another point on the line.

   It's clear that the claim follows directly from Axiom 3—Every line has at least two distinct points. But suppose we formalize each statement and then try to prove that Axiom 3 implies the claim. Here are two formalizations:

   Axiom 3: $\forall L\ \exists x\ \exists y\ ((x \neq y) \wedge L(x) \wedge L(y))$.

   Claim: $\forall x\ \forall L\ (L(x) \rightarrow \exists y\ ((x \neq y) \wedge L(y)))$.

   a. Prove that Axiom 3 implies the claim.

   b. Prove that Axiom 3 does not follow from the claim. Hint: find an interpretation that is a model for Axioms 1, 2, 4, and the claim, but not for Axiom 3.

Solutions

1. a. Two. Since L is a predicate of order one, it follows that q and $\exists L$ both have order two.

   b. Three. Since L is a predicate of order one, it follows that M and $\exists L$ both have order two. Thus $\forall M$ has order three.

   c. Four. Since L is a predicate of order one, it follows that B and $\exists L$ both have order two. So M and $\exists B$ both have order three. Therefore $\exists M$ has order four.

2. a. $\exists A\ \exists B\ \exists x\ ((A(x) \wedge \neg B(x)) \vee (\neg A(x) \wedge B(x)))$.

   b. $\exists x\ \exists y\ \exists S\ (S(x) \wedge \neg S(y))$.

3. If we agree to use equality symbols, then we have the following second order wff:

$$\forall L\ \forall M\ ((L \neq M) \rightarrow \forall x\ (L(x) \rightarrow \neg M(x))).$$

4. If we agree to use equality symbols, then we have the following second order wff:

$$\forall C\ \forall D\ ((C \neq D) \rightarrow \exists x\ \exists y\ ((x \neq y) \wedge C(x) \wedge C(y) \wedge D(x) \wedge D(y) \wedge \forall z\ (C(z) \wedge D(z) \rightarrow (z = x) \vee (z = y)))).$$

5. If we agree to use equality symbols, then we have the following second order wff:

   $\forall L\ \forall x\ (\neg L(x) \rightarrow \exists C\ (C(x) \wedge \exists y\ (C(y) \wedge L(y) \wedge \forall z\ (C(z) \wedge L(z) \rightarrow (z = y)))))$.

6. Think of the predicate p as a set and $p(x)$ as the statement $x \in p$. Then for any interpretation with domain D, the wff can be restated as follows: For every subset p of D there is an element $x \in p$. But if we choose $p = \varnothing$, then there is no element in p. So any interpretation makes the wff false. Therefore the wff is unsatisfiable and invalid.

7. Think of the predicates p and q as sets and $p(x)$ and $q(x)$ as the statements $x \in p$ and $x \in q$, respectively. Then for any interpretation with domain D, the wff can be restated as follows: There is a subset p of D such that p is a subset of every subset q of D. If we choose $p = \varnothing$, then p is a subset of every subset of D. So any interpretation makes the wff true. Therefore the wff is valid and satisfiable.

8.

1.	$\exists x\ \exists y\ \forall P\ P(x, y)$	P
2.	$\forall P\ P(a, b)$	1, EI, EI
3.	$\quad \exists P\ \forall x\ \forall y\ \neg P(x, y)$	P [for $\forall P\ \exists x\ \exists y\ P(x, y)$]
4.	$\quad \neg p(a, b)$	3, EI, UI, UI
5.	$\quad p(a, b)$	2, UI
6.	$\quad$ False	4, 5, Contr
7.	$\forall P\ \exists x\ \exists y\ P(x, y)$	3–6, IP
	QED	1, 2, 7, CP.

9. a.

1.	$\exists x\ \exists L\ (L(x) \wedge \forall y\ ((x = y) \vee \neg L(y)))$	P [for IP]
2.	$l(a) \wedge \forall y\ ((a = y) \vee \neg l(y))$	1, EI, EI
3.	$\forall y\ ((a = y) \vee \neg l(y))$	2, Simp
4.	$\forall L\ \exists x\ \exists y\ ((x \neq y) \wedge L(x) \wedge L(y))$	Axiom 3
5.	$(b \neq c) \wedge l(b) \wedge l(c)$	4, UI, EI, EI
6.	$b \neq c$	5, Simp
7.	$(a = b) \vee \neg l(b)$	3, UI
8.	$l(b)$	5, Simp
9.	$a = b$	7, 8, DS
10.	$(a = c) \vee \neg l(c)$	3, UI
11.	$l(c)$	5, Simp
12.	$a = c$	10, 11, DS
13.	$b = c$	9, 12, Symmetry, Transitive
14.	False	6, 13, Contr
	QED	1–15, IP.

   b. We can define a little geometry with three "points" x, y, and z, and four "lines" $l_1 = \varnothing$, $l_2 = \{x, y\}$, $l_3 = \{x, z\}$, and $l_4 = \{y, z\}$. It is clear that this little geometry satisfies Axioms 1, 2, 4, and the claim. But Axiom 3 is not satisfied. Thus Axiom 3 can't follow from the claim.

Chapter 9

Computational Logic

9.1 Automatic Reasoning

Learning Objectives

Be able to transform first-order wffs into clausal form.

Be able to unify atoms from a set of clauses.

Be able to describe the resolution proof rule.

Be able to use resolution to write formal proofs in first-order logic.

Review Questions

1. What is a clause?
2. What is a clausal form?
3. What is Skolem's rule?
4. What is Skolem's algorithm?
5. What is a substitution?
6. What is the composition of two substitutions?
7. What is a unifier?
8. What is a most general unifier?
9. What is a unification algorithm?
10. What is the resolution rule for propositions?
11. What is the resolution rule for first-order wffs?

12. What are the steps to prove a wff is valid if resolution must be used?

13. What is the meaning of each of the following symbols or expressions?

    a. $\square$.

    b. $\{x/t, y/s\}$.

    c. ε.

    d. $E\theta$.

    e. $\theta\sigma$.

    f. $C\theta - N$.

    g. $R(S)$.

14. How do resolution proofs work?

Solved Problems

1. Use Skolem's rule to remove the existential quantifiers from the following wff.

$$\exists x\ \forall y\ \exists z\ \forall w\ A(x, y, z, w).$$

2. Transform the following wff into clausal form.

$$\exists x\ \forall y\ q(x, y) \rightarrow \forall z\ \exists x\ p(x, z).$$

3. Find a clausal form and corresponding set of clauses for each of the following propositions.

    a. $\neg\ [(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)]$.

    b. $\neg\ ((p \vee q \rightarrow r) \wedge p \rightarrow r)$.

4. Given the following propositional wff.

$$(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r).$$

    Prove that the wff is valid by using resolution to show the unsatisfiability of the set of clauses in the clausal form of the negation of the wff. Find two different resolution proofs.

5. Let $\theta = \{x/f(y), z/w\}$ and $\sigma = \{y/a, w/f(b), x/a\}$.

    a. Find the composition $\theta\sigma$.

    b. Find the composition $\sigma\theta$.

6. Given the following two wffs:

$$p(x, g(y), f(x)) \text{ and } p(a, w, f(y)).$$

   a. Use Robinson's algorithm to find a most general unifier.

   b. Use the Martelli-Montanari algorithm to find a most general unifier.

   c. Is there a unifier that is not most general?

7. Given the following two wffs:

$$p(x, h(x, g(z)), y) \text{ and } p(x, h(a, w), g(x)).$$

   a. Use Robinson's algorithm to find a most general unifier.

   b. Use the Martelli-Montanari algorithm to find a most general unifier.

   c. Is there a unifier that is not most general?

8. Given the following two clauses to resolve:

$$p(x) \vee p(a) \vee q(x, z) \text{ and } \neg p(y) \vee \neg p(g(x)) \vee r(y) \vee q(b, z).$$

   a. Resolve the clauses by choosing two atoms from the first clause and one literal from the second clause.

   b. Resolve the clauses by choosing one atom from the first clause and two literals from the second clause.

9. For each of the following wffs, prove that the wff is valid by using resolution to show that set of clauses in the clausal form of the negation the wff is unsatisfiable.

   a. $\exists x\, p(x) \vee \exists x\, q(x) \rightarrow \exists x\, (p(x) \vee q(x))$.

   b. $\forall x\, \forall y\, (A(x, y) \rightarrow B(x, y)) \rightarrow [\forall x\, \forall y\, A(x, y) \rightarrow \forall x\, \forall y\, B(x, y)]$.

   c. $[\forall x\, (A(x) \wedge B(x) \rightarrow C(x))] \rightarrow [\forall x\, A(x) \wedge \forall x\, B(x) \rightarrow \forall x\, C(x)]$.

Solutions

1. $\forall y\, \forall w\, A(c, y, f(y), w)$.

2. First we'll construct the prenex conjunctive normal form. After renaming variables we obtain the following wff.

$$\exists x\, \forall y\, q(x, y) \rightarrow \forall z\, \exists w\, p(w, z).$$

   Now remove the conditional and move negations to the right to obtain the following wff.

$$\forall x\, \exists y\, \neg q(x, y) \vee \forall z\, \exists w\, p(w, z).$$

   Next, apply Skolem's rule and move remaining universal quantifiers left to obtain the following wff in clausal form.

$$\forall x\, \forall z\, (\neg q(x, f(x)) \vee p(g(z), z)).$$

Note: We could have moved all quantifiers left and then applied Skolem's rule to obtain the following wff in clausal form.

$$\forall x\, \forall z\, (\neg q(x, f(x)) \vee p(g(x, z), z)).$$

3. a. We can use equivalences to obtain the clausal form, which for propositional wffs is just the conjunctive normal form obtained as follows:

$$\begin{aligned} \neg [(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)] &\equiv (p \rightarrow q) \wedge (q \rightarrow r) \wedge \neg (p \rightarrow r) \\ &\equiv (\neg p \vee q) \wedge (\neg q \vee r) \wedge p \wedge \neg r. \end{aligned}$$

The corresponding set of four clauses is $\{\neg p \vee q,\ \neg q \vee r, p, \neg r\}$.

b. We can use equivalences to obtain the clausal form, which for propositional wffs is just the conjunctive normal form obtained as follows:

$$\begin{aligned} \neg ((p \vee q \rightarrow r) \wedge p \rightarrow r) &\equiv (p \vee q \rightarrow r) \wedge p) \wedge \neg r \\ &\equiv (\neg (p \vee q) \vee r) \wedge p) \wedge \neg r \\ &\equiv ((\neg p \wedge \neg q) \vee r) \wedge p) \wedge \neg r \\ &\equiv (\neg p \vee r) \wedge (\neg q \vee r) \wedge p \wedge \neg r. \end{aligned}$$

The corresponding set of four clauses is $\{\neg p \vee r,\ \neg q \vee r,\ p, \neg r\}$.

4. In Problem (3a) we found the set of clauses in the clausal form of the negation of this wff to be $\{\neg p \vee q, \neg q \vee r, p, \neg r\}$. One of several resolution proofs proceeds as follows:

1.	$\neg p \vee q$	*P*
2.	$\neg q \vee r$	*P*
3.	p	*P*
4.	$\neg r$	*P*
5.	q	1, 3, *R*
6.	$\neg q$	2, 4, *R*
7.	$\square$	5, 6, *R*

QED.

A second resolution proof proceeds as follows:

1.	$\neg p \vee q$	*P*
2.	$\neg q \vee r$	*P*
3.	p	*P*
4.	$\neg r$	*P*
5.	$\neg p \vee r$	1, 2, *R*
6.	r	3, 5, *R*
7.	$\square$	4, 6, *R*

QED.

5. a. $\theta\sigma = \{x/f(a), z/f(b), y/a, w/f(b)\}$.

b. $\sigma\theta = \{y/a, w/f(b), x/a\} = \theta$.

6. a and b. $\{x/a, w/g(a), y/a\}$. c. No, because there are no variables in the denominators of a most general unifier.

7. a and b. $\{x/a, y/g(a), w/g(z)\}$. c. Yes, because z can be chosen to be any term.

8. First, rename variables so that the clauses have distinct variable names. For example, in the second clause we'll replace x by u and z by v to obtain the following two clauses.

$$p(x) \vee p(a) \vee q(x, z) \text{ and } \neg p(y) \vee \neg p(g(u)) \vee r(y) \vee q(b, v).$$

   a. Choose $p(x)$ and $p(a)$ from the first clause and $\neg p(y)$ from the second clause. An mgu for the set $\{p(x), p(a), p(y)\}$ is $\theta = \{x/a, y/a\}$. Now apply θ to both clauses and then use the resolution rule to obtain the following resolvant:

$$q(a, z) \vee \neg p(g(u)) \vee r(a) \vee q(b, v).$$

   b. Choose $p(x)$ from the first clause and $\neg p(y)$ and $\neg p(g(u))$ from the second clause. An mgu for the set $\{p(x), p(y), p(g(u))\}$ is $\theta = \{x/g(u), y/g(u)\}$. Now apply θ to both clauses and then use the resolution rule to obtain the following resolvant:

$$p(a) \vee q(g(u), z) \vee r(g(u)) \vee q(b, v).$$

9. In each case, take the negation of the wff, find the clausal form, and use the clauses as premises in a resolution proof.

   a.
   1. $p(a) \vee q(b)$ — *P*
   2. $\neg p(z)$ — *P*
   3. $\neg q(z)$ — *P*
   4. $q(b)$ — 1, 2, *R*, $\{z/a\}$
   5. $\square$ — 3, 4, *R*, $\{z/b\}$

   QED.

   b.
   1. $\neg A(x, y) \vee B(x, y)$ — *P*
   2. $A(u, v)$ — *P*
   3. $\neg B(a, b)$ — *P*
   4. $B(x, y)$ — 1, 2, *R*, $\{u/x, v/y\}$
   5. $\square$ — 3, 4, *R*, $\{x/a, y/b\}$

   QED.

   c.
   1. $\neg A(x) \vee \neg B(x) \vee C(x)$ — *P*
   2. $A(y)$ — *P*
   3. $B(z)$ — *P*
   4. $\neg C(a)$ — *P*
   5. $\neg B(x) \vee C(x)$ — 1, 2, *R*, $\{y/x\}$
   6. $C(x)$ — 3, 5, *R*, $\{z/x\}$
   7. $\square$ — 4, 6, *R*, $\{x/a\}$

   QED.

9.2 Logic Programming

Learning Objectives

Be able to describe a logic program.

Be able to describe how resolution is used to execute a logic program.

Review Questions

1. What is a logic program clause?

2. What is a logic program goal or query?

3. What is a logic program?

4. What is the SLD-resolution rule?

5. What is a computation tree for a goal?

6. What is the meaning of each of the following symbols or expressions?

   a. $\leftarrow A$.

   b. $C \leftarrow A, B$.

Solved Problems

1. Given the predicates isMale, isFemale, isMotherOf, isFatherOf, and isSiblingOf, define each of the following logic program predicates. If you need other predicates, be sure to define them too.

   a. isParentOf. b. isGrandFatherOf.

   c. isAuntOf. d. isNephewOf.

2. Given the isParentOf predicate, define the logic program predicate isCousinOf predicate, where isCousinOf(n, a, b) means that a and b are nth cousins and isCousinOf(0, a, b) = isSiblingOf(a, b).

3. In Prolog, a disjunction of two atoms is represented by placing a semicolon between them. Explain, from a logical point of view, why the Prolog clause

   p :- q; r.

   is equivalent to the two Prolog clauses

   p :- q.
   p :- r.

4. Given the following logic program:

   $p(\langle\ \rangle, w)$.
   $p(x :: y, x :: z) \leftarrow p(y, z)$.

   Construct an SLD-resolution proof for the goal

   $\leftarrow p(\langle a, b\rangle, \langle a, b, c\rangle)$.

5. Given the following logic program:

   $p(x, \langle x\rangle)$.
   $p(x, x :: y) \leftarrow p(x, y)$.

   Describe the values of z that are generated by backtracking with the depth-first search strategy for the goal $\leftarrow p(b, z)$.

6. Given the following logic program:

   $p(a)$.
   $p(x) \leftarrow r(x), s(x)$.
   $r(b)$.
   $r(x) \leftarrow s(f(x)), t(x)$.
   $s(b)$.
   $s(c)$.
   $s(f(c))$.
   $t(c)$.

   Draw the SLD-tree (i.e., the computation tree) for the goal

   $\leftarrow p(y)$.

7. Given the following logic program and goal:

   $p(a)$.
   $q(a)$.
   $r(x) \leftarrow p(x), q(x)$.
   $\leftarrow r(y)$.

   a. Write an SLD-resolution proof of the given program and its goal.

   b. Write the corresponding resolution proof for the clauses represented as first-order wffs.

8. Translate the following functional definition into a logic program.

   $$f(n) = \text{if } n = 0 \text{ then } 0 \text{ else } n + f(n - 1).$$

9. Translate the following functional definition into a logic program.

   $$g(x) = \text{if } x = \langle\ \rangle \text{ then } 0 \text{ else } 1 + g(\text{tail}(x)).$$

10. Write a logic program for the predicate "sumProd" that computes the sum and product of the numbers in a list. For example, the goal

$$\leftarrow \text{sumProd}(\langle 3, 2, 5\rangle, S, P).$$

returns $S = 10$ and $P = 30$.

11. In the blocks world suppose that we are given the predicates "on" and "on_top", where on(a, b) means that a is on top of b and on_top(a) means that nothing is on top of a. Use these predicates to write a logic program to find the depth of a block, where depth(A, N) means that there are N blocks on top of A.

12. Write a logic program for the predicate "squares" that computes the list of squares of a list of numbers. For example, the goal

$$\leftarrow \text{squares}(\langle 4, 3, 7\rangle, A).$$

will return the answer $A = \langle 16, 9, 49\rangle$.

13. Find a logic program to distribute an element over a list by pairing it with each element in the list. For example, the goal

$$\leftarrow \text{dist}(a, \langle b, c, d\rangle, x)$$

should return $x = \langle (a, b), (a, c), (a, d)\rangle$.

14. Find a logic program to take an element and a list and return the list of all pairs that contain the given element in one position and an element of the list in the other position. For example, the goal

$$\leftarrow \text{makePairs}(a, \langle b, c, d\rangle, x)$$

should return $x = \langle (a, b), (b, a), (a, c), (c, a), (a, d), (d, a)\rangle$.

15. Find a logic program that takes a list and returns all possible pairs of elements in the list. For example, the goal

$$\leftarrow p(\langle a, b, c\rangle, x)$$

should return

$$x = \langle (a, a), (a, b), (b, a), (a, c), (c, a), (b, b), (b, c), (c, b), (c, c)\rangle.$$

Solutions

1. a. isParentOf(x, y) ← isMotherOf(x, y).
   isParentOf(x, y) ← isFatherOf(x, y).

   b. isGrandFatherOf(x, y) ← isMale(x),
   isParentOf(x, z),
   isParentOf(z, y).

   c. isAuntOf(x, y) ← isFemale(x), isSiblingOf(x, z), isParentOf(z, y).

   d. isNephewOf(x, y) ← isMale(x), isParentOf(z, x), isSiblingOf(z, y).

2. isCousinOf(0, x, y) $\leftarrow$ isSiblingOf(x, y).

   isCousinOf(n, x, y) $\leftarrow$ isParentOf(u, x),
   isParentOf(v, y),
   isCousinOf($n - 1$, u, v).

3. The Prolog clause

   p :- q; r.

   represents the wff

   $q \vee r \rightarrow p$.

   Using equivalences, we can transform this wff as follows:

   $$\begin{aligned} q \vee r \rightarrow p &\equiv \neg (q \vee r) \vee p \\ &\equiv (\neg q \wedge \neg r) \vee p \\ &\equiv (\neg q \vee p) \wedge (\neg r \vee p) \\ &\equiv (q \rightarrow p) \wedge (r \rightarrow p). \end{aligned}$$

   The last wff can be represented by the two Prolog clauses

   p :- q.
   p :- r.

4. | | | |
   |---|---|---|
   | 1. | $p(\langle\ \rangle, w)$ | P |
   | 2. | $p(x :: y, x :: z) \leftarrow p(y, z)$ | P |
   | 3. | $\leftarrow p(\langle a, b\rangle, \langle a, b, c\rangle)$ | P |
   | 4. | $\leftarrow p(\langle b\rangle, \langle b, c\rangle)$ | 2, 3, R, $\theta_1 = \{x/a, y/\langle b\rangle, z/\langle b, c\rangle\}$ |
   | 5. | $\leftarrow p(\langle\ \rangle, \langle c\rangle)$ | 2, 4, R, $\theta_2 = \{x/b, y/\langle\ \rangle\}$ |
   | 6. | $\square$ | 1, 5, R, $\theta_3 = \{w/\langle c\rangle\}$ |

   QED.

5. The possible values of z are $\langle b\rangle$, $\langle b, b\rangle$, $\langle b, b, b\rangle$, ... To see this, we'll give three SLD-refutations and calculate three values of z. The shortest refutation can be described as follows:

1.	$p(x, \langle x\rangle)$	P
2.	$p(x, x :: y) \leftarrow p(x, y)$	P
3.	$\leftarrow p(b, z)$	P
4.	$\square$	1, 3, R, $\theta = \{x/b, z/\langle b\rangle\}$

   QED.

   The value of z is calculated by $z\theta = \langle b\rangle$.

The next shortest refutation can be described as follows:

1.	$p(x, \langle x \rangle)$	P
2.	$p(x, x :: y) \leftarrow p(x, y)$	P
3.	$\leftarrow p(b, z)$	P
4.	$\leftarrow p(b, y_1)$	2, 3, R, $\theta_1 = \{x_1/b, z/b :: y_1\}$
5.	$\square$	1, 4, R, $\theta_2 = \{x_2/b, y_1/\langle b \rangle\}$

QED.

The value of z is calculated by $z\theta_1\theta_2 = (b :: y_1)\theta_2 = b :: \langle b \rangle = \langle b , b \rangle$.

The next shortest proof follows:

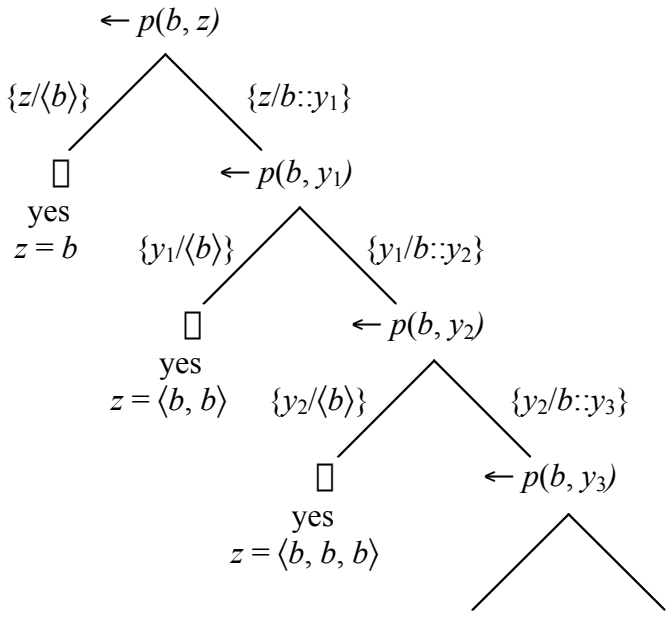
1.	$p(x, \langle x \rangle)$	P
2.	$p(x, x :: y) \leftarrow p(x, y)$	P
3.	$\leftarrow p(b, z)$	P
4.	$\leftarrow p(b, y_1)$	2, 3, R, $\theta_1 = \{x_1/b, z/b :: y_1\}$
5.	$\leftarrow p(b, y_2)$	2, 4, R, $\theta_2 = \{x_2/b, y_1/b :: y_2\}$
6.	$\square$	1, 5, R, $\theta_3 = \{x_3/b, y_2/\langle b \rangle\}$

QED.

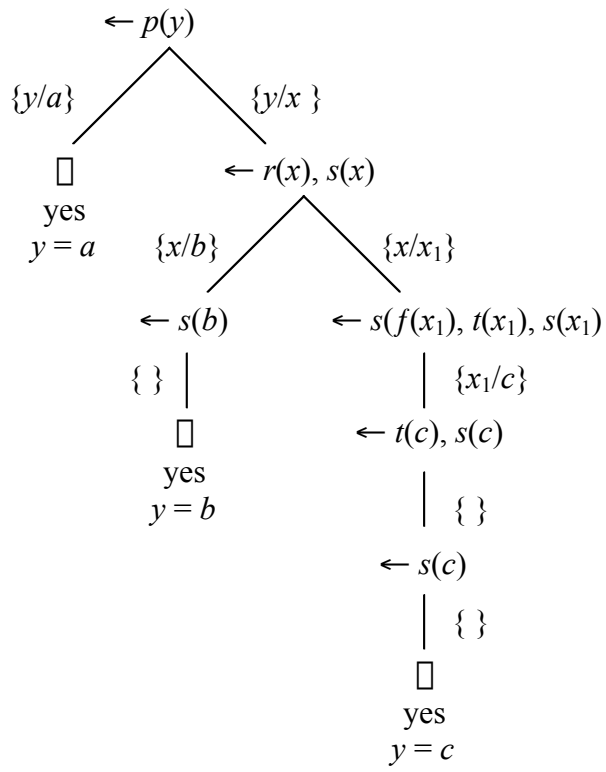
The value of z is calculated by

$z\theta_1\theta_2\theta_3 = (b :: y_1)\theta_2\theta_3 = (b :: b :: y_2)\theta_3 = b :: b :: \langle b \rangle = \langle b, b , b \rangle$.

The following infinite SLD-tree reflects these three proofs.



6. There are three different refutations, which are shown on the following SLD-tree.



7. a.

1.	$p(a)$	*P*
2.	$q(a)$	*P*
3.	$r(x) \leftarrow p(x), q(x)$	*P*
4.	$\leftarrow r(y)$	*P*
5.	$\leftarrow p(x), q(x)$	3, 4, *R*, $\theta_1 = \{y/x\}$
6.	$\leftarrow q(a)$	1, 5, *R*, $\theta_2 = \{x/a\}$
7.	□	2, 6, *R*, $\theta_3 = \{ \}$

QED.

b.

1.	$p(a)$	*P*
2.	$q(a)$	*P*
3.	$r(x) \vee \neg p(x) \vee \neg q(x)$	*P*
4.	$\neg r(y)$	*P*
5.	$\neg p(x) \vee \neg q(x)$	3, 4, *R*, $\theta_1 = \{y/x\}$
6.	$\neg q(a)$	1, 5, *R*, $\theta_2 = \{x/a\}$
7.	□	2, 6, *R*, $\theta_3 = \{ \}$

QED.

8. Letting pf be the predicate to implement f, we have the following definition for pf:

pf(0, 0).
pf(n, $n + x$) ← pf($n - 1$, x).

9. Letting pg be the predicate to implement *g*, we have the following definition for pg:

   pg(⟨ ⟩, 0).
   pg($h :: t$, $1 + x$) ← pg(t, x).

10. Assuming that an empty sum is 0 and an empty product is 1, we have the following definition for sumProd:

    sumProd(⟨ ⟩, 0, 1).
    sumProd($h :: t$, $h + x$, $h * y$) ← sumProd(t, x, y).

11. Assuming that the depth of a block that is on top is zero, we have the following definition for depth:

    depth(x, 0) ← on_top(x).
    depth(x, $n + 1$) ← on(y, x), depth(y, n).

12. squares(⟨ ⟩, ⟨ ⟩).
    squares($h :: t$, $(h * h) :: x$) ← squares(t, x).

13. dist(x, ⟨ ⟩, ⟨ ⟩).
    dist(x, $h :: t$, $(x, h) :: y$) ← dist(x, t, y).

14. makePairs(x, ⟨ ⟩, ⟨ ⟩).
    makePairs(x, $h :: t$, $(x, h) :: (h, x) :: y$) ← makePairs(x, t, y).

15. Let cat(x, y, z) mean that z is the concatenation of the lists x and y. (Note that Exercise (7b) in the book implements a logic program to concatenate lists.) Now we can write a definition for p in terms of cat and the makePairs predicate from Problem 14.

    p(⟨ ⟩, ⟨ ⟩).
    p($h :: t$, $(h, h) :: z$) ← makePairs(h, t, x), p(t, y), cat(x, y, z).

Chapter 10

Algebraic Structures and Techniques

10.1 What Is an Algebra?

Learning Objectives

Be able to answer the question "What is an algebra?"

Be able to recognize simple properties of binary operations.

Review Questions

1. What is an algebra?

2. What does the expression $\langle A; s, a \rangle$ mean?

3. What is an algebraic expression?

4. What is high school algebra?

5. What is a concrete algebra?

6. What is an abstract algebra?

Solved Problems

1. Let $\circ$ be defined over the set $\{0, 1, 2, 3\}$ by $x \circ y = (x + y) \bmod 4$. Answer each of the following questions.
   a. Does $\circ$ have a zero?
   b. Does $\circ$ have an identity?
   c. What elements, if any, have inverses?
   d. Is $\circ$ commutative?
   e. Is $\circ$ associative?

2. Let $S = \{a, b, c\}$ and let $\circ$ be a binary operation on S that is not associative. Suppose this can be demonstrated by the following two equations.

   $a \circ (b \circ c) = a \circ b = c$

   $(a \circ b) \circ c = c \circ c = a$

   a. Can $\circ$ have a zero?

   b. Can $\circ$ have an identity?

   c. Can $\circ$ be commutative?

3. Let m and n be two integers with $m < n$. Let $B = \{m, m + 1, ..., n\}$. Let "max" be the function that returns the larger of its two arguments.

   a. Does max have a zero? If so, what is it?

   b. Does max have an identity? If so, what is it?

4. Given the algebra $\langle \mathbb{Z}; +, 0 \rangle$. Use equational reasoning to prove the following statement for all integers a.

   If $a + a = a$, then $a = 0$.

5. Given an algebra $\langle A; \circ, e \rangle$, where $\circ$ is a binary operation and $e \in A$. Find some properties that the algebra must have so that we can prove the following statement for all elements $a \in A$.

   If $a \circ a = a$, then $a = e$.

6. Given an algebra $\langle A; f, a \rangle$, where f is unary and a is a constant in A and the axiom $f(f(f(x))) = x$ holds for all $x \in A$.

   a. Write down a finite set of algebraic expressions to represent the distinct elements of A.

   b. Write a recursive definition for the function "simp" that simplifies an arbitrary expression of the form $f(f(...f(a)...))$. For example,

   $\text{simp}(f(f(f(f(a))))) = f(a)$.

7. Given the algebra $\langle S; g, a, b, c \rangle$ where g is a unary operation on S and a, b, and c are constants in S. Write a recursive definition for the function "expr" that tests for expressions over S. For example, $\text{expr}(g(g(b)))$ = True and $\text{expr}(g(d)) = \text{expr}(f(a))$ = False.

Solutions

1. a. No.

   b. Yes, it is 0.

   c. 1 and 3 are inverses of each other and 2 is its own inverse.

   d. Yes, because + is commutative.

   e. Yes, because + is associative.

2. a. No. b. No. c. Yes.
3. a. Yes. The zero is n because $\max(n, x) = \max(x, n) = n$ for all $x \in B$.

   b. Yes. The identity is m because $\max(m, x) = \max(x, m) = x$ for every $x \in B$.
4. Here is an equational proof that includes the reason for each step.

$$\begin{aligned} a &= a + 0 && \text{(0 is an identity for +)} \\ &= a + (a + -a) && \text{(inverses exist)} \\ &= (a + a) + -a && \text{(+ is associative)} \\ &= a + -a && \text{(hypothesis)} \\ &= 0 && \text{(inverses exist).} \end{aligned}$$

5. This is similar to Problem 4. The properties that are sufficient to prove the statement are that e is an identity for $\circ$, every element of A has an inverse, and $\circ$ is associative. Then we can prove the statement as follows:

$$\begin{aligned} a &= a \circ e && \text{(e is an identity for $\circ$)} \\ &= a \circ (a \circ a^{-1}) && \text{(inverses exist)} \\ &= (a \circ a) \circ a^{-1} && \text{($\circ$ is associative)} \\ &= a \circ a^{-1} && \text{(hypothesis)} \\ &= e && \text{(inverses exist).} \end{aligned}$$

6. a. $\{a, f(a), f(f(a))\}$.

   b. Using equations we can define simp as follows:

   $\text{simp}(a) = a$
   $\text{simp}(f(a)) = f(a)$
   $\text{simp}(f(f(a))) = f(f(a))$
   $\text{simp}(f(x)) = \text{simp}(x)$.
7. We'll write an if-then-else definition.

   $\text{expr}(E)$ = if $(E = a)$ or $(E = b)$ or $(E = c)$ then True
   else if $E = g(x)$ then $\text{expr}(x)$
   else False.

10.2 Boolean Algebra

Learning Objectives

Be able to describe the properties of a Boolean algebra.

Be able to apply the properties to simplify Boolean expressions.

Review Questions

1. What does the symbol $\bar{x}$ mean?

2. What is a Boolean algebra?

3. How are the algebra of sets and the algebra of propositions related?

4. What is the idempotent property?

5. What are the absorption laws?

6. What is the involution law?

7. What are De Morgan's laws?

8. How are digital circuits related to Boolean algebra?

9. What does it mean to simplify a Boolean expression?

Solved Problems

1. Simplify each of the following Boolean expressions.
   a. $ac + bc + abc + bbc$.
   b. $ab + bc + abc$.
   c. $(a + b)(a + b + c + d)$.
   d. $\overline{a} + \overline{b}\,a$.
   e. $ac + a\overline{b}\,c + b$.
   f. $\overline{\left(\overline{a} + b\right)\left(\overline{b} + a\right)} + a$.
2. Let $B = \{0, 1, a, b\}$ be the carrier of a 4-element Boolean algebra. Prove that
$$\overline{a} = b.$$

Solutions

1. a. $ac + bc + abc + bbc = [(ac) + (ac)b)] + [(bc) + (bc)b] = ac + bc = (a + b)c$.
   b. $ab + bc + abc = [(ab) + (ab)c] + bc = ab + bc = (a + c)b$.
   c. $(a + b)(a + b + c + d) = (a + b)[(a + b) + (c + d)] = a + b$.
   d. $\overline{a} + \overline{b}\,a = \left(\overline{a} + \overline{b}\right)\left(\overline{a} + a\right) = \left(\overline{a} + \overline{b}\right) = \overline{ab}$.
   e. $ac + a\overline{b}\,c + b = \left(a + a\overline{b}\right)c + b = ac + b$.

f. $\overline{(\bar{a}+b)(\bar{b}+a)}+a=\overline{(\bar{a}+b)}+\overline{(\bar{b}+a)}+a$

$=a\bar{b}+b\bar{a}+a$

$=(a\bar{b}+a)+b\bar{a}$

$=a+b\bar{a}$

$=a+b.$

2. Suppose, by way of contradiction, that $\bar{a}\neq b$. Then there are three possibilities for $\bar{a}$. We'll find a contradiction in each case. If $\bar{a}=0$, then

$$a+\bar{a}=a+0=a.$$

But we also have $a+\bar{a}=1$. So we must conclude that $a = 1$, which contradicts the fact that 1 and a are distinct elements of B. If $\bar{a}=1$, then

$$a\bar{a}=a1=a.$$

But we also have $a\bar{a}=0$. So we must conclude that $a = 0$, which contradicts the fact that 0 and a are distinct elements of B. If $\bar{a}=a$, then

$$a\bar{a}=aa=a.$$

But we also have $a\bar{a}=0$. So again we must conclude that $a = 0$, which contradicts the fact that 0 and a are distinct elements of B. Since we've reached contradictions in all cases, it follows that $\bar{a}=b$.

.

10.3 Abstract Data Types as Algebras

Learning Objectives

Be able to describe an abstract data type.

Be able to apply appropriate algebraic properties to write recursive definitions for simple functions in terms of operations for abstract data types.

Review Questions

What is an abstract data type?

Solved Problems

1. Given the algebra ⟨ℕ, Boolean; 0, isZero, succ, pred⟩. Write a recursive definition for the function gt : $\mathbb{N}\times\mathbb{N}\rightarrow$ Boolean, where gt(x, y) means that $x > y$.
2. Given the algebra ⟨ℕ, Boolean; 0, isZero, succ, pred⟩. Write a recursive definition for the function le : $\mathbb{N}\times\mathbb{N}\rightarrow$ Boolean, where le(x, y) means that $x \leq y$.

3. Use only the primitive operations in the algebra of lists to give a recursive definition for the function removeAll : $A \times \text{lists}(A) \rightarrow \text{lists}(A)$ that removes all occurrences of an element from a list. For example,

$$\text{removeAll}(a, \langle b, a, c, a\rangle) = \langle b, c\rangle.$$

4. Use the primitive operations in the algebras of queues and lists to give a recursive definition for the function $f : \text{lists}(A) \rightarrow Q[A]$ that places the elements of a list into a queue, where the head of the list becomes the front of the queue.

5. Use the primitive operations in the algebra of stacks to give a recursive definition for a function to combine two stacks into a single stack by stacking one stack on the other. For example, if A and B are stacks, then combine(A, B) is the stack with top the top of A and with bottom the bottom of B.

6. Use the primitive operations in the algebras of queues and stacks to give a recursive definition for the function $f : Q[A] \rightarrow \text{Stks}[A]$ that places the elements of a queue into a stack, where the front of the queue becomes the bottom element of the stack.

7. Use the primitive operations in the algebras of priority queues and stacks to give a recursive definition for the function $f : P[A] \rightarrow \text{Stks}[A]$ that places the elements of a priority queue into a stack, where the best element of the priority queue becomes the top element of the stack.

8. Use the primitive operations in the algebras of priority queues and stacks to give a recursive definition for the function $f : P[A] \rightarrow \text{Stks}[A]$ that places the elements of a priority queue into a stack, where the best element of the priority queue becomes the bottom element of the stack.

Solutions

1. A definition with equations can be written as follows:

$$\begin{aligned}&\text{gt}(0, y) = \text{False}\\ &\text{gt}(\text{succ}(x), 0) = \text{True}\\ &\text{gt}(\text{succ}(x), \text{succ}(y)) = \text{gt}(x, y).\end{aligned}$$

   An if-then-else definition can be written as follows:

$$\begin{aligned}\text{gt}(x, y) = \ &\text{if isZero}(x) \text{ then False}\\ &\text{else if isZero}(y) \text{ then True}\\ &\text{else gt}(\text{pred}(x), \text{pred}(y)).\end{aligned}$$

2. A definition with equations can be written as follows:

$$\begin{aligned}&\text{le}(0, y) = \text{True}\\ &\text{le}(\text{succ}(x), 0) = \text{False}\\ &\text{le}(\text{succ}(x), \text{succ}(y)) = \text{le}(x, y).\end{aligned}$$

   An if-then-else definition can be written as follows:

$$\begin{aligned}\text{le}(x, y) = \ &\text{if isZero}(x) \text{ then True}\\ &\text{else if isZero}(y) \text{ then False}\\ &\text{else le}(\text{pred}(x), \text{pred}(y)).\end{aligned}$$

3. A definition with equations can be written as follows:

   removeAll(x, ⟨ ⟩) = ⟨ ⟩
   removeAll(x, x :: t) = removeAll(x, t)
   removeAll(x, h :: t) = h :: removeAll(x, t).

   An if-then-else definition can be written as follows:

   removeAll(x, L) = if isEmpty(L) then ⟨ ⟩
   else if x = head(L) then removeAll(x, tail(L))
   else head(L) :: removeAll(x, tail(L)).

4. An if-then-else definition can be written as follows, where we use the append function from Example 10.19 of the book:

   f(L) = if isEmpty(L) then emptyQ
   else if isEmpty(tail(L)) then addQ(head(L), emptyQ))
   else apQ(addQ(head(L), emptyQ)), f(tail(L))).

5. An if-then-else definition can be written as follows:

   combine(A, B) = if isEmptyStk(A) then B
   else push(top(A), combine(pop(A), B)).

6. An if-then-else definition can be written as follows, where we use the combine function from Problem 5:

   f(q) = if isEmptyQ(q) then emptyStk
   else if isEmptyQ(delQ(q)) then push(frontQ(q), emptyStk)
   else combine(f(delQ(q)), push(frontQ(q), emptyStk)).

7. An if-then-else definition can be written as follows:

   f(p) = if isEmptyP(p) then emptyStk
   else push(best(p), f(delBest(p))).

8. An if-then-else definition can be written as follows, where we use the combine function from Problem 5:

   f(p) = if isEmptyP(p) then emptyStk
   else if isEmptyP(delBest(p)) then push(best(p), emptyStk)
   else combine(f(delBest(p)), push(best(p), emptyStk)).

10.4 Computational Algebras

Learning Objectives

Be able to apply appropriate algebraic properties to write expressions to represent relations constructed in terms of operations for relational databases.

Be able to describe a functional algebra.

Review Questions

1. What is the select operation?

2. What is the projection operation?

3. What is the join operation?

4. What is the meaning of each of the following symbols or expressions?

   a. $R \triangleright\!\triangleleft S$.

   b. select(R, A, B).

   c. project(R, $\{A, B\}$).

   d. join(R, S).

5. What is a functional algebra?

6. Why is FP an important algebra?

Solved Problems

1. Let R be the family relation with attributes Person, Mother, and Father. For example, if R contains the 3-tuple

   (John, Margaret, George),

   then the person John has mother Margaret and father George. Now we can use R to answer questions about families. To shorten the notation we'll let P, M, and F stand for Person, Mother, and Father, respectively. For example, to find the children of John, we can construct the relation

   project(select(R, F, John), $\{P\}$).

   Construct relations to find the following sets of family members.

   a. The parents of John, which should return {(Margaret, George)}.

   b. The children of Margaret and George, which should have the form

      {(John), (a), (b), ...}.

2. The following two tables represent two relational databases for Farms and Harvests.

Farms

Name	*Crop*	*Acres*	*County*
Jones	corn	1500	Washington
Jones	barley	2500	Washington
Smith	wheat	600	Lincoln
Appleby	soybeans	2000	Washington
Nelson	corn	500	Jefferson
Nelson	soybeans	3500	Jefferson
Hein	hops	2000	Adams
Hein	grapes	200	Adams
Truman	corn	2500	Madison
Hill	wheat	3000	Lincoln

Harvests

Crop	*Month*
corn	July
barley	June
wheat	May
soybeans	June
grapes	September
hops	August

Construct an expression to find each of the following sets.

a. The crops that are harvested in June. In other words, the set {(barley), (soybeans)}.

b. The months that wheat is harvested. In other words, the set {(May)}.

c. The list of acreages planted in corn. In other words, the set {(1500), (500), (2500)}.

d. The months the Nelson farm harvests crops. In other words, the set {(June), (July)}.

e. The crops planted in Washington county. In other words, the set {(corn), (barley), (soybeans)}.

f. The counties that plant corn. In other words, the set {(Washington), (Jefferson), (Madison)}.

g. The harvest months in Adams and Lincoln counties. In other words, the set {(May), (August), (September)}.

3. Write an FP definition for the function seq, where seq(n) = $\langle 0, 1, ..., n \rangle$.
4. Use FP algebra to prove the following statement.

   $[f, g, h] \mathbin{@} (a \rightarrow b; c) = (a \rightarrow [f \mathbin{@} b, g \mathbin{@} b, h \mathbin{@} b]; [f \mathbin{@} c, g \mathbin{@} c, h \mathbin{@} c])$.

Solutions

1. a. project(select(R, P, John), {M, F}).
   b. project(select(select(R, F, George), M, Margaret) {P}).
2. a. project(select(Harvests, Month, June), {Crop}).
   b. project(select(Harvests, Crop, Wheat), {Month}).
   c. project(select(Farms, Crop, Corn), {Acres}).
   d. project(select(join(Farms, Harvests), Name, Nelson), {Month}).
   e. project(select(Farms, County, Washington), {Crop}).
   f. project(select(Farms, Crop, Corn), {County}).
   g. Let J = join(Farms, Harvests). Then the harvest months in Adams and Lincoln counties is obtained by the union of the following two relations:

      project(select(J, County, Adams), {Month})

      project(select(J, County, Lincoln), {Month}).
3. We'll use eq0 and sub1 from Examples 10.27 and 10.28 to write the following FP definition for seq:

   $$\text{seq} = \text{eq0} \rightarrow \sim \langle 0 \rangle; \text{apndr} \mathbin{@} [\text{seq} \mathbin{@} \text{sub1}, \text{id}].$$

4. The statement follows directly from the axioms for the FP algebra.

   $$\begin{aligned}
   & [f, g, h] \mathbin{@} (a \rightarrow b; c) \\
   = \; & [f \mathbin{@} (a \rightarrow b; c), g \mathbin{@} (a \rightarrow b; c), h \mathbin{@} (a \rightarrow b; c)] \\
   = \; & [(a \rightarrow f \mathbin{@} b; f \mathbin{@} c), (a \rightarrow g \mathbin{@} b; g \mathbin{@} c), (a \rightarrow h \mathbin{@} b; h \mathbin{@} c)] \\
   = \; & \text{id} \mathbin{@} [(a \rightarrow f \mathbin{@} b; f \mathbin{@} c), (a \rightarrow g \mathbin{@} b; g \mathbin{@} c), (a \rightarrow h \mathbin{@} b; h \mathbin{@} c)] \\
   = \; & a \rightarrow \text{id} \mathbin{@} [f \mathbin{@} b, g \mathbin{@} b, h \mathbin{@} b]; \text{id} \mathbin{@} [f \mathbin{@} c, g \mathbin{@} c, h \mathbin{@} c] \\
   = \; & a \rightarrow [f \mathbin{@} b, g \mathbin{@} b, h \mathbin{@} b]; [f \mathbin{@} c, g \mathbin{@} c, h \mathbin{@} c].
   \end{aligned}$$

10.5 Other Algebraic Ideas

Learning Objectives

Be able to describe congruences.

Be able to describe and use the RSA algorithm.

Be able to describe subalgebras and morphisms of algebras.

Review Questions

1. What does the expression $x \equiv y \pmod{n}$ mean?

2. What is a congruence?

3. How is the RSA algorithm used?

4. What is a subalgebra?

5. What is a morphism?

Solved Problems

1. Use the Chinese remainder theorem to solve each of the following sets of congruences for a unique smallest natural number x.

   a. $x \equiv 2 \pmod{5}$
      $x \equiv 3 \pmod{6}$.

   b. $x \equiv 3 \pmod{7}$
      $x \equiv 4 \pmod{11}$.

   c. $x \equiv 1 \pmod{3}$
      $x \equiv 2 \pmod{5}$
      $x \equiv 3 \pmod{7}$.

   d. $x \equiv 6 \pmod{5}$
      $x \equiv 4 \pmod{7}$
      $x \equiv 2 \pmod{11}$.

2. To construct some keys for using the RSA algorithm, let $p = 7$ and $q = 13$ so that $n = pq = 91$. Let the decryption key be $d = 25$. Find an encryption key e for the given choices of p, q, and d.

3. Given the algebra $\langle \mathbb{N}_{15}; +_{15}, 0 \rangle$, find the carriers of the subalgebras generated by each of the following sets.

   a. $\{5\}$. b. $\{3\}$. c. $\{4\}$.

4. Find the three morphisms that exist from the algebra $\langle \mathbb{N}_3; +_3, 0 \rangle$ to the algebra $\langle \mathbb{N}_6; +_6, 0 \rangle$.

5. Given the language morphism $f : \{a, b\}^* \rightarrow \{a, b\}^*$ defined by $f(a) = ba$ and $f(b) = ab$, compute the value of each of the following expressions.

   a. $f(\{b^n a \mid n \in \mathbb{N}\})$.

   b. $f^{-1}(\{b^n a \mid n \in \mathbb{N}\})$.

   c. $f^{-1}(L)$, where L is the language of the following grammar:

   $$S \rightarrow baSba \mid abSab \mid \Lambda.$$

   d. $f(L)$, where L is the language of the following grammar:

   $$S \rightarrow aSb \mid bSa \mid \Lambda.$$

Solutions

1. a. First, we must find solutions b_1 and b_2 to the two congruences

$$6b_1 \equiv 1 \pmod 5$$
$$5b_2 \equiv 1 \pmod 6$$

Two solutions are $b_1 = 1$ and $b_2 = 5$. Next, we calculate

$$x = 6 \cdot b_1 \cdot 2 + 5 \cdot b_2 \cdot 3 = 6 \cdot 1 \cdot 2 + 5 \cdot 5 \cdot 3 = 12 + 75 = 87$$

Since $87 \geq 30 = 5 \cdot 6$, we'll let $x = 87 - 2 \cdot 30 = 87 - 60 = 27$. So $x = 27$ is the smallest natural number that satisfies the two congruences.

b. First, we must find solutions b_1 and b_2 to the two congruences

$$11b_1 \equiv 1 \pmod 7$$
$$7b_2 \equiv 1 \pmod{11}$$

Two solutions are $b_1 = 2$ and $b_2 = 8$. Next, we calculate

$$x = 11 \cdot b_1 \cdot 3 + 7 \cdot b_2 \cdot 4 = 11 \cdot 2 \cdot 3 + 7 \cdot 8 \cdot 4 = 66 + 224 = 290$$

Since $290 \geq 77 = 7 \cdot 11$, we'll let $x = 290 - 3 \cdot 77 = 290 - 231 = 59$. So $x = 59$ is the smallest natural number that satisfies the two congruences.

c. First, we must find solutions b_1, b_2, and b_3 to the three congruences

$$35b_1 \equiv 1 \pmod 3$$
$$21b_2 \equiv 1 \pmod 5$$
$$15b_3 \equiv 1 \pmod 7$$

Three solutions are $b_1 = 2$, $b_2 = 1$, and $b_3 = 1$. Next, we calculate

$$x = 35 \cdot b_1 \cdot 1 + 21 \cdot b_2 \cdot 2 + 15 \cdot b_3 \cdot 3 = 35 \cdot 2 \cdot 1 + 21 \cdot 1 \cdot 2 + 15 \cdot 1 \cdot 3 = 157.$$

Since $157 \geq 105 = 3 \cdot 5 \cdot 7$, we'll let $x = 157 - 1 \cdot 105 = 52$. So $x = 52$ is the smallest natural number that satisfies the three congruences.

d. First, we must find solutions b_1, b_2, and b_3 to the three congruences

$$77b_1 \equiv 1 \pmod 5$$
$$55b_2 \equiv 1 \pmod 7$$
$$35b_3 \equiv 1 \pmod{11}$$

Three solutions are $b_1 = 3$, $b_2 = 6$, and $b_3 = 6$. Next, we calculate

$$x = 77 \cdot b_1 \cdot 6 + 55 \cdot b_2 \cdot 4 + 35 \cdot b_3 \cdot 2 = 77 \cdot 3 \cdot 6 + 55 \cdot 6 \cdot 4 + 35 \cdot 6 \cdot 2 = 3126.$$

Since $3126 \geq 385 = 5 \cdot 7 \cdot 11$, we'll let $x = 3126 - 8 \cdot 385 = 46$. So $x = 46$ is the smallest natural number that satisfies the three congruences.

2. Notice that $(p - 1)(q - 1) = 6 \cdot 12 = 72$ and $d = 25$. Since $\gcd(25, 72) = 1$, it follows that d is a proper decryption key. To find an encryption key e, we need to solve the equation $25e \bmod 72 = 1$. By reversing Euclid's algorithm, or by wits, we find that $1 = 25(-23) + 72 \cdot 8$. So $e = -23$ solves the equation. But we need e to be positive. Since we can add any multiple of 72, we'll choose $e = -23 + 72 = 49$.

3. a. $\{0, 5, 10\}$. b. $\{0, 3, 6, 9, 12\}$. c. $\mathbb{N}_{15}$.
4. The three morphisms are defined as follows: the zero function; the function f defined by $f(0) = 0$, $f(1) = 2$, $f(2) = 4$; the function g defined by $g(0) = 0$, $g(1) = 4$, $g(2) = 2$.
5. a. $\{(ab)^n ba \mid n \in \mathbb{N}\}$.

   b. $\{a\}$.

   c. The set of even palindromes over $\{a, b\}$.

   d. The set of even palindromes over $\{a, b\}$ that are derived from the grammar

$$S \rightarrow baSab \mid abSba \mid \Lambda.$$

Chapter 11

Regular Languages and Finite Automata

11.1 Regular Languages

Learning Objectives

Be able to describe regular languages and regular expressions.

Be able to use algebraic properties of regular expressions to simplify regular expressions.

Review Questions

1. What is a regular language?

2. What is a regular expression?

3. What is the meaning of a regular expression?

4. What operations are used in the algebra of regular expressions?

Solved Problems

1. Write down a regular expression for each of the following regular languages.
   a. $\{\Lambda, aa, aaaa, ..., a^{2n}, ...\}$.
   b. $\{ab, abb, ..., ab^n, ...\}$.
   c. $\{\Lambda, b, bc, bcc, ..., bc^n...\}$.
   d. $\{\Lambda, a, aa, ..., a^n, ...\} \cup \{aba, abba, ..., ab^na, ...\}$.
   e. $\{ab^n \mid n \in \mathbb{N}\} \cup \{ba^n \mid n \in \mathbb{N}\}$.
2. Describe, in words, the language of the following regular expression:
   $$(b + ab)^*(\Lambda + a).$$
3. Find a regular expression for the language of all strings over $\{a, b, c\}$ that contain exactly two c's.

4. Simplify each of the following regular expressions.
   a. $aba^* + abaa^*$.
   b. $(ba^*)^* + aa^*(ba^*)^*$.
   c. $a^*(ba^* + ca^*)^*$.

Solutions

1. a. $(aa)^*$.
   b. abb^*.
   c. $\Lambda + bc^*$.
   d. $a^* + abb^*a$.
   e. $ab^* + ba^*$.
2. All strings over $\{a, b\}$, where no a is next to another a. In other words, there are no substrings of two or more consecutive a's.
3. $(a + b)^*c(a + b)^*c\ (a + b)^*$.
4. a. $aba^* + abaa^* = ab(a^* + aa^*) = ab(\Lambda + a)a^* = aba^*$.
   b. $(ba^*)^* + aa^*(ba^*)^* = (\Lambda + aa^*)(ba^*)^* = a^*(ba^*)^* = (a + b)^*$.
   c. $a^*(ba^* + ca^*)^* = a^*((b + c)a^*)^* = (a + (b + c))^* = (a + b + c)^*$.

11.2 Finite Automata

Learning Objectives

Be able to describe DFAs and NFAs.

Be able to apply algorithms to transform between regular expressions and finite automata.

Be able to describe finite automata with output.

Review Questions

1. What is a finite automaton?

2. What does DFA mean?

3. What does NFA mean?

4. What is a Mealy machine?

5. What is a Moore machine?

6. What is the meaning of each of the following symbols or expressions?

   a. $T(i, a) = j$.

   b. $T(i, a) = \{j, k\}$.

   c. a/x.

   d. i/x.

7. What does it mean to say that DFAs are equivalent to NFAs?

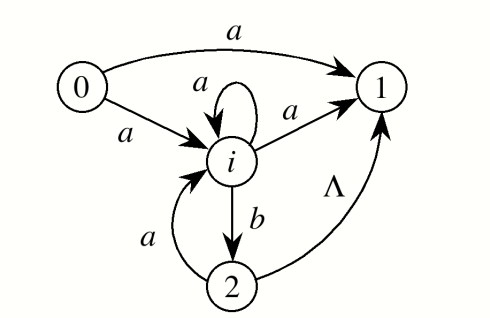
8. How do you transform a regular expression into a finite automaton?

9. How do you transform a finite automaton into a regular expression?

Solved Problems

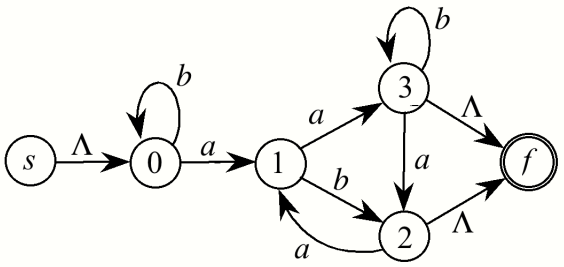
1. Find a DFA to recognize each of the following regular languages.
   a. $\{ab, aa\}$.
   b. $\{a^{2n+1} \mid n \in \mathbb{N}\}$.
   c. $\{ab^n a \mid n \in \mathbb{N}\}$.
2. Find an NFA to recognize each regular expression.
   a. $a^*b + a$.
   b. $a^*b + b^*a$.
3. Find a DFA to recognize each regular expression. In each case, try to find the DFA by your wits. Then transform the corresponding NFA solutions to Problem 2 into DFAs. Compare your results.
   a. $a^*b + a$.
   b. $a^*b + b^*a$.
4. Find an NFA to recognize the language of each of the following regular expressions.
   a. $a(ab + a)^*b$.
   b. $ab^*a + a(ab + a^*)b$.

5. The following diagram represents part of an automaton.

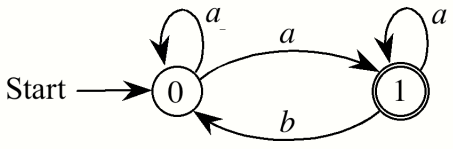


Assume that you are in the process of obtaining a regular expression for the language of the given automaton and the current task is to remove state *i*. Write down each new edge needed to remove state *i* in the form new(_, _) = *R*, where *R* is the regular expression associated with the new edge.

6. Apply the method of removing states to the following NFA to find its regular expression. Remove the states in the order 0, 1, 2, and 3.



7. Apply the method of removing states to the following NFA to find its regular expression.



a. Remove state 0 first.

b. Remove state 1 first.

c. Prove that the regular expressions from (a) and (b) are equal.

8. Suppose we want to take a nonempty string over the alphabet $\{a, b\}$ and collapse all substrings of b's to a single b. For example, if the input string is *abbaaabbbab*, then the output string is *abaaabab*.

   a. Construct a Mealy machine to solve the problem.

   b. Construct a Moore machine to solve the problem.

Solutions

1. a. There are four states, 0 (start), 1, 2 (final), and 3. The transitions are: $T(0, a) = 1$, $T(0, b) = 3$, $T(1, a) = T(1, b) = 2$, and $T(2, a) = T(2, b) = T(3, a) = T(3, b) = 3$.

   b. There are three states, 0 (start), 1 (final) and 2. The transitions are: $T(0, a) = 1$, $T(1, a) = 2$, and $T(2, a) = 1$.

   c. There are four states, 0 (start), 1, 2 (final), and 3. The transitions are: $T(0, a) = 1$, $T(0, b) = 3$, $T(1, a) = 2$, $T(1, b) = 1$, and $T(2, a) = T(2, b) = T(3, a) = T(3, b) = 3$.

2. a. There are three states, 0 (start), 1, and 2 (final). The transitions are: $T(0, a) = \{2\}$, $T(0, \Lambda) = \{1\}$, $T(1, a) = \{1\}$, and $T(1, b) = \{2\}$.

   b. There are four states, 0 (start), 1, 2, and 3 (final). The transitions are: $T(0, \Lambda) = \{1, 2\}$, $T(1, a) = \{1\}$, $T(1, b) = \{3\}$, $T(2, a) = \{3\}$, and $T(2, b) = \{2\}$.

3. a. There are five states, 0 (start), 1 (final), 2, 3 (final), and 4. The transitions are: $T(0, a) = 1$, $T(0, b) = 3$, $T(1, a) = 2$, $T(1, b) = 3$, and $T(2, a) = 2$, $T(2, b) = 3$, and $T(3, a) = T(3, b) = T(4, a) = T(4, b) = 4$.

   b. There are seven states, 0 (start), 1 (final), 2 (final), 3, 4, 5 (final), and 6. The transitions are: $T(0, a) = 1$, $T(0, b) = 2$, $T(1, a) = 3$, $T(1, b) = 5$, and $T(2, a) = 5$, $T(2, b) = 4$, $T(3, a) = 3$, $T(3, b) = 5$, $T(4, a) = 5$, $T(4, b) = 4$, $T(5, a) = T(5, b) = T(6, a) = T(6, b) = 6$.

4. a. One solution has four states, 0 (start), 1, 2, and 3 (final). The transitions are: $T(0, a) = \{1\}$, $T(1, a) = \{1, 2\}$, $T(1, b) = \{3\}$, and $T(2, b) = \{1\}$.

   b. One solution has six states, 0 (start), 1, 2, 3, 4, and 5 (final). The transitions are: $T(0, a) = \{1\}$, $T(1, a) = \{2, 5\}$, $T(1, b) = \{1\}$, $T(1, \Lambda) = \{4\}$, $T(2, b) = \{3\}$, $T(3, b) = \{5\}$, $T(4, a) = \{4\}$, and $T(4, b) = \{5\}$.

5. There are four new edges labeled with the following regular expressions:

   $\text{new}(0, 1) = a + ba^*a$.
   $\text{new}(0, 2) = \varnothing + ba^*b = ba^*b$.
   $\text{new}(2, 1) = \Lambda + aa^*a$.
   $\text{new}(2, 2) = \varnothing + aa^*b = aa^*b$.

6. If the states are removed in the order 0, 1, 2, 3, then the following regular expression is obtained.

$$b^*ab(ab)^* + b^*aab^*(\Lambda + a(ab)^*).$$

7. a. $a^*a(a + ba^*a)^*$.

   b. $(a + aa^*b)^*aa^*$.

   c. We'll start with the answer to (a) and proceed to the answer for (b), with references to the properties listed in (11.1).

$$
\begin{aligned}
a^*a(a + ba^*a)^* &= a^*a[a^*(ba^*aa^*)^*] && \text{(by 11.1(7) with } R = a, S = ba^*a) \\
&= (a^*aa^*)[b(a^*aa^*)]^* && \text{(by 11.1(2), associative property)} \\
&= (a^*aa^*b)^*a^*aa^* && \text{(by 11.1 (8) with } R = a^*aa^*, S = b) \\
&= [(a^*aa^*b)^*a^*]aa^* && \text{(by 11.1(2), associative property)} \\
&= (a + aa^*b)^*aa^* && \text{(by 11.1(7) with } R = a, S = aa^*b).
\end{aligned}
$$

8. a. One solution has two states, 0 (start) and 1. We'll write the transitions in the following form:

   T(state, input/output) = next state

   With this notation, the transitions are $T(0, a/a) = 0$, $T(0, b/b) = 1$, $T(1, b/\Lambda) = 1$, and $T(1, a/a) = 0$.

   b. One solution has four states, 0 (start), 1, 2, and 3. We'll write the transitions in the following form:

   T(state/output, input) = next state

   With this notation, the transitions are $T(0/\Lambda, a) = 1$, $T(0/\Lambda, b) = 2$, $T(1/a, a) = 1$, $T(1/a, b) = 2$, $T(2/b, a) = 1$, $T(2/b, b) = 3$, $T(3/\Lambda, a) = 1$, and $T(3/\Lambda, b) = 3$.

11.3 Constructing Efficient Finite Automata

Learning Objectives

Be able to apply algorithms to transform NFAs to DFAs and DFAs to minimum-state DFAs.

Review Questions

1. What is the meaning of the expression $\lambda(s)$?

2. What is the meaning of the equation $T_{\min}([s], a) = [T(s, a)]$?

3. How do you transform an NFA to a DFA?

4. How do you transform a DFA into a minimum-state DFA?

5. What is the lambda closure of a set of states?

Solved Problems

1. Find a DFA that is equivalent to the following NFA.

		a	b	Λ
start	0	$\varnothing$	$\varnothing$	{1}
	1	{2,3}	$\varnothing$	$\varnothing$
	2	$\varnothing$	{3}	{1}
	3	{4}	$\varnothing$	{2,4}
final	4	$\varnothing$	$\varnothing$	$\varnothing$

2. Find a DFA that is equivalent to the following NFA.

		a	b	Λ
start	0	{1,2}	$\varnothing$	{7}
	1	{3}	$\varnothing$	$\varnothing$
	2	$\varnothing$	{4}	$\varnothing$
	3	$\varnothing$	{5}	$\varnothing$
	4	{6}	$\varnothing$	$\varnothing$
	5	$\varnothing$	$\varnothing$	{1,7}
	6	$\varnothing$	$\varnothing$	{2,7}
final	7	$\varnothing$	$\varnothing$	$\varnothing$

3. Find a DFA that is equivalent to the following NFA.

		a	b	Λ
start	0	{1,2}	$\varnothing$	{2}
	1	$\varnothing$	{0}	$\varnothing$
final	2	{1}	{1}	{1}

4. Find a DFA that is equivalent to the following NFA.

		a	b	Λ
start	0	{1}	{2}	{2}
	1	$\varnothing$	{2}	$\varnothing$
final	2	{2}	$\varnothing$	$\varnothing$

5. Transform the following NFA into a DFA, where 0 is the start state and 2 is the final state.

$$T(0, a) = \{1, 2\},\ T(0, \Lambda) = T(2, \Lambda) = \{1\},\ T(1, b) = T(2, a) = \{2\}.$$

6. Transform the following NFA into a DFA: The states of the NFA are 0 (start), 1, 2 (final), and 3. The state transitions are $T(0, a) = \{1\}$, $T(0, \Lambda) = \{3\}$, $T(1, b) = \{2\}$, and $T(3, a) = \{2, 3\}$.

7. Suppose that the set of states for a DFA is $S = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$, where 0 is the start state and the final states are 5, 7, and 8.

   a. Write down the set E_0 to start the process of finding equivalent states.

   b. Suppose that for some k we calculate the following equality:

   $$E_k = E_{k+1} = \{\{0, 1\}, \{0, 3\}, \{0, 4\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \{2, 6\}, \{5, 7\}\}.$$

   Write down the states (i.e., the equivalence classes that partition S) in a minimum-state DFA and note the start state and final states.

8. Find a minimum-state DFA for the following DFA: The states are 0 (start), 1 (final), 2 final), 3 (final), and 4, where the state transitions are defined by $T(0, a) = 3$, $T(0, b) = 1$, $T(1, a) = T(2, a) = 2$, $T(1, b) = T(2, b) = T(3, b) = 4$, and $T(3, a) = T(4, a) = T(4, b) = 4$.

9. Find a minimum-state DFA for the following DFA.

		a	b
start	0	1	2
	1	4	3
	2	3	4
final	3	3	3
final	4	4	4

Solutions

1. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	b
start	$\{0,1\}$	$\{1,2,3,4\}$	$\varnothing$
final	$\{1,2,3,4\}$	$\{1,2,3,4\}$	$\{1,2,3,4\}$
	$\varnothing$	$\varnothing$	$\varnothing$

or

		a	b
start	0	1	2
final	1	1	1
	2	2	2

2. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	b
start, final	$\{0,7\}$	$\{1,2\}$	$\varnothing$
	$\{1,2\}$	$\{3\}$	$\{4\}$
	$\{3\}$	$\varnothing$	$\{1,5,7\}$
	$\{4\}$	$\{2,6,7\}$	$\varnothing$
final	$\{1,5,7\}$	$\{3\}$	$\varnothing$
final	$\{2,6,7\}$	$\varnothing$	$\{4\}$
	$\varnothing$	$\varnothing$	$\varnothing$

or

		a	b
start, final	0	1	6
	1	2	3
	2	6	4
	3	5	6
final	4	2	6
final	5	6	3
	6	6	6

3. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	*b*
start, final	{0,1,2}	{1,2}	{0,1,2}
final	{1,2}	{1}	{0,1,2}
	{1}	∅	{0,1,2}
	∅	∅	∅

or

		a	*b*
start, final	0	1	0
final	1	2	0
	2	3	0
	3	3	3

4. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	*b*
start, final	{0,2}	{1,2}	{2}
final	{1,2}	{2}	{2}
final	{2}	{2}	∅
	∅	∅	∅

or

		a	*b*
start, final	0	1	2
final	1	2	2
final	2	2	3
	3	3	3

5. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	*b*
start	{0,1}	{1,2}	{1,2}
final	{1,2}	{1,2}	{1,2}

or

		a	*b*
start	0	1	1
final	1	1	1

6. We can construct the DFA table by (11.8). Here is the DFA table to gether with a simplification obtained by renaming states.

		a	*b*
start	{0,3}	{1,2,3}	∅
final	{1,2,3}	{2,3}	{2}
final	{2,3}	{2,3}	∅
final	{2}	∅	∅
	∅	∅	∅

or

		a	*b*
start	0	1	4
final	1	2	3
final	2	2	4
final	3	4	4
	4	4	4

7. a. E_0 = {{0, 1}, {0, 2}, {0, 3}, {0, 4}, {0, 6}, {1, 2}, {1, 3}, {1, 4}, {1, 6}, {2, 3}, {2, 4}, {2, 6}, {3, 4}, {3, 6}, {4, 6}, {5, 7}, {5, 8}, {7, 8}}.

   b. [0] = {0, 1, 3, 4}
   [2] = {2, 6}
   [5] = {5, 7}
   [8] = {8}.

   The start state is [0] and the final states are [5] and [8].

8. To find the equivalence classes, we start with E_0:

   $E_0 = \{\{0, 4\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\}$
   $E_1 = \{\{1, 2\}\}$
   $E_2 = \{\{1, 2\}\}$.

   Therefore the states 1 and 2 are equivalent. It follows that the set of states {0, 1, 2, 3, 4} is partitioned into the four classes

   [0] = {0}
   [1] = {1, 2}
   [3] = {3}
   [4] = {4}

   The resulting minimum-state DFA table and its simplification after renaming states are

		a	*b*
start	[0]	[3]	[1]
final	[1]	[1]	[4]
final	[3]	[4]	[4]
	[4]	[4]	[4]

or

		a	*b*
start	0	2	1
final	1	1	3
final	2	3	3
	3	3	3

9. To find the equivalence classes, we start with E_0:

   $E_0 = \{\{0, 1\}, \{0, 2\}, \{1, 2\}, \{3, 4\}\}$
   $E_1 = \{\{1, 2\}, \{3, 4\}\}$
   $E_2 = \{\{1, 2\}, \{3, 4\}\}$.

   Therefore the set of states {0, 1, 2, 3, 4} is partitioned into the three classes

   [0] = {0}
   [1] = {1, 2}
   [3] = {3, 4}

   The resulting minimum-state DFA table and its simplification after renaming states are

		a	*b*
start	[0]	[1]	[1]
	[1]	[3]	[3]
final	[3]	[3]	[3]

or

		a	*b*
start	0	1	1
	1	2	2
final	2	2	2

11.4 Regular Language Topics

Learning Objectives

Be able to construct regular grammars for simple languages.

Be able to transform between regular grammars and NFAs.

Be able to describe and apply the pumping lemma for regular languages.

Review Questions

1. What is a regular grammar?

2. How do you transform a regular grammar into a finite automaton?

3. How do you transform a finite automaton into a regular grammar?

4. What is the pumping lemma?

5. How is the pumping lemma used?

Solved Problems

1. Find a regular grammar for the language of each of the following regular expressions.
   a. $a^*bc + b^*$.
   b. a^*b^*c.
   c. aa^*cbb^*d.
   d. $a^*b(ab)^* + aa^*b$.
2. Find a regular grammar for each of the following languages.
   a. $\{ba^m c^n \mid m, n \in \mathbb{N}\}$.
   b. $\{a^k bc^m de^n \mid k, m, n \in \mathbb{N}\}$.
3. Use (11.11) to transform the following NFA into a regular grammar: The states are 0 (start), 1, and 2 (final). The state transitions are given by $T(0, a) = \{0, 1\}$, $T(0, b) = \{2\}$, $T(0, \Lambda) = \{2\}$, $T(1, b) = \{1, 2\}$.
4. Use (11.12) to transform each of the following regular grammars into an NFA.
   a. $S \to abS \mid aS \mid ba$
   b. $S \to \Lambda \mid c \mid aS \mid abS$.

5. Show that the language $\{ab^ncd^{3n} \mid n \in \mathbb{N}\}$ is not regular by using a pumping lemma argument.
6. Show that the language $\{a^nb^{2n} \mid n \in \mathbb{N}\}$ is not regular by using the following two facts: $\{a^nb^n \mid n \in \mathbb{N}\}$ is not regular and there is a morphism $f : \{a, b\}^* \to \{a, b\}^*$ defined by $f(a) = a$ and $f(b) = bb$.

Solutions

1. a. $S \to A \mid B$
   $A \to aA \mid bc$
   $B \to bB \mid \Lambda$.

   b. $S \to aS \mid B$
   $B \to bB \mid c$.

   c. $S \to aS \mid acB$
   $B \to bB \mid bd$.

   d. $S \to T \mid R$
   $S \to aS \mid bT$
   $T \to abT \mid \Lambda$
   $R \to aR \mid ab$.

2. a. $S \to bT$
   $T \to aT \mid U$
   $U \to cU \mid \Lambda$.

   b. $S \to aS \mid bC$
   $C \to cC \mid dE$
   $E \to eE \mid \Lambda$.

3. First, rename the states to capital letters with S, I, and F representing 0, 1, and 2, respectively. With these states as nonterminals, the grammar can be written as follows:

   $S \to aS \mid aI \mid bF \mid F$
   $I \to bI \mid b$

4. a. First transform the grammar to the following grammar, where the right side of each production has at most one terminal followed by a nonterminal.

   $S \to aB \mid aS \mid bA$
   $B \to bS$
   $A \to a$.

   Now the states of the NFA are S (start), A, B, and F (final) and the transitions are

   $T(S, a) = \{B, S\}$, $T(S, b) = \{A\}$, $T(B, b) = \{S\}$, and $T(A, a) = \{F\}$.

b. First transform the grammar to the following grammar, where the right side of each production has at most one terminal followed by a nonterminal.

$S \rightarrow aS \mid aB \mid c \mid \Lambda$
$B \rightarrow bS.$

Now the states of the NFA are S (start and final), B, and F (final) and the transitions are: $T(S, a) = \{S, B\}$, $T(S, c) = \{F\}$, and $T(B, b) = \{S\}$.

5. Let L be the language and assume, by way of contradiction, that L is regular. Then let $s = ab^m cd^{3m}$, where $m > 0$ is the integer from the pumping lemma (11.13). It then follows from (11.13) that $s = ab^m cd^{3m} = xyz$, where $y \neq \Lambda$, $|xy| \leq m$, and $xy^k z \in L$ for all $k \geq 0$. Since $|xy| \leq m$, it follows that either $y = ab^i$ (in which case $x = \Lambda$) with $0 \leq i < m$ or $y = b^j$ with $1 \leq j < m$.

Now we can obtain a contradiction is several ways. For example, if we let $k = 0$, then (11.13) tells us that $xz \in L$. Since either $y = ab^i$ or $y = b^j$, it follows that either $xz = b^{m-i}cd^{3m}$ or $xz = ab^{m-j}cd^{3m}$ with $j \geq 1$. But neither of these representations of xz fits the form of strings in L, which contradicts the pumping lemma's assertion that $xz \in L$. So L is not regular.

An alternative contradiction can be obtained by letting $k = 2$ and considering the string $xy^2z \in L$. Since either $y = ab^i$ or $y = b^j$, it follows that either $xy^2z = ab^iab^ib^{m-i}cd^{3m}$ or $xy^2z = ab^{m+j}cd^{3m}$ with $j \geq 1$. But neither of these representations of xy^2z fits the form of strings in L, which contradicts the pumping lemma's assertion that $xy^2z \in L$. So L is not regular.

6. Let $L = \{a^n b^{2n} \mid n \in \mathbb{N}\}$ and assume, by way of contradiction, that L is regular. By the definition of f, it follows that $f(a^n b^n) = a^n b^{2n}$ for all $n \in \mathbb{N}$. So if we let $M = \{a^n b^n \mid n \in \mathbb{N}\}$, then $f^{-1}(L) = M$. Since we are assuming that L is regular, we can use (11.15b) to conclude that $f^{-1}(L)$ must also be regular. But $f^{-1}(L) = M$ and we already know by Example 11.25 that M is not regular. This contradiction implies that L is not regular.

Chapter 12

Context-Free Languages and Pushdown Automata

12.1 Context-Free Languages

Learning Objectives

Be able to construct context-free grammars for simple languages.

Review Questions

1. What is a context-free grammar?

2. What is a context-free language?

Solved Problems

1. Find a context-free grammar for each of the following languages.
   a. $\{a^{n+2}bc^n \mid n \in \mathbb{N}\}$.
   b. $\{a^n b \mid n \in \mathbb{N}\}^*$.
   c. $(\{a^n(bc)^n \mid n \geq 0\} \cup \{c^n \mid n \geq 1\})\{a^n b^{n+1} \mid n \geq 0\}^*$.

Solutions

1. a. Notice that an arbitrary string of the language can be put in the following form:

   $$a^{n+2}bc^n = aa(a^n bc^n).$$

   This makes it easier to discover the following grammar.

   $$S \to aaT$$
   $$T \to aTc \mid b.$$

   b. The language has the form L^* where $L = \{a^n b \mid n \in \mathbb{N}\}^*$. If we let T be the start symbol for a grammar for L, then by (12.3) a grammar for L^* takes the form

   $$S \to TS \mid \Lambda.$$

   A grammar for L has the form $T \to aT \mid b$. So a grammar for L^* is

   $$S \to TS \mid \Lambda.$$
   $$T \to aT \mid b.$$

c. The language has the form $(L \cup M)(N^*)$ where

$L = \{a^n(bc)^n \mid n \geq 0\}$
$M = \{c^n \mid n \geq 1\}$
$N = \{a^n b^{n+1} \mid n \geq 0\}^*$.

If T, U, and V are the start symbols for grammars for L, M, and N, respectively, then by (12.3) a grammar for $(L \cup M)(N^*)$ takes the form

$S \rightarrow TV \mid UV$
$T \rightarrow aTbc \mid \Lambda$
$U \rightarrow cU \mid c$
$V \rightarrow RV \mid \Lambda$.
$R \rightarrow aRb \mid b$.

12.2 Pushdown Automata

Learning Objectives

Be able to describe a PDA.

Be able to apply algorithms to transform between PDAs that accept by final state and those that accept by empty stack.

Be able to apply algorithms to transform between context-free grammars and PDAs that accept by empty stack.

Review Questions

1. What does the expression (i, b, C, pop, j) mean?

2. What is a pushdown automaton (PDA)?

3. What is acceptance by final state?

4. What is acceptance by empty stack?

5. How do you transform a final state PDA into an empty stack PDA?

6. How do you transform an empty stack PDA into a final state PDA?

7. How do you transform a context-free grammar into a PDA?

8. How do you transform a PDA into a context-free grammar?

9. What is a deterministic PDA?

10. What is a nondeterministic PDA?

11. Do deterministic and nondeterministic PDAs have the same power?

Solved Problems

1. Find a pushdown automaton for the language of all palindromes over the alphabet $\{a, b\}$.
2. Use (12.7) to transform each of the following context-free grammars into a single-state PDA that accepts by empty stack, where the PDA instructions may contain multiple stack operations.
   a. $S \rightarrow \Lambda \mid c \mid aSb$.
   b. $S \rightarrow bBS \mid aB$
      $B \rightarrow bbB \mid \Lambda$.
   c. $S \rightarrow aSb \mid BC$
      $B \rightarrow b \mid bB$
      $C \rightarrow \Lambda \mid cC$.
3. Assume that the following instruction is part of the instruction set for a single-state PDA that accepts by empty stack and uses multiple stack operations in some instructions.

   $$(0, \Lambda, S, \langle \text{pop}, \text{push}(b), \text{push}(S), \text{push}(a) \rangle, 0).$$

   Transform the instruction into a set of PDA instructions that accomplish the same purpose, where each instruction has one stack operation but may use states other than 0. Assume that the PDA uses stack symbols a, b, A, B, S, and X and that it starts with all six symbols on the stack with X on top.
4. Given the following empty-stack PDA with start state 0 and starting stack symbol X.

   $(0, a, X, \text{push}(A), 0)$
   $(0, a, A, \text{nop}, 1)$
   $(1, a, A, \text{push}(A), 1)$
   $(1, b, A, \text{pop}, 1)$
   $(1, b, X, \text{pop}, 1)$.

   a. Use (12.8) to transform the PDA to a context-free grammar. DO NOT simplify the resulting grammar. In other words, list all the productions generated by the algorithm.
   b. Simplify the result of Part (a).
   c. Try to describe the language of the PDA and corresponding grammar.

Solutions

1. Let ? stand for any stack symbol. Then an empty-stack PDA to accept all palindromes over $\{a, b\}$ can be written as follows:

$(0, a, X, \text{pop}, 0)$	(accept a)
$(0, b, X, \text{pop}, 0)$	(accept b)
$(0, \Lambda, X, \text{pop}, 0)$	(accept Λ)
$(0, a, ?, \text{push}(a), 0)$	
$(0, b, ?, \text{push}(b), 0)$	
$(0, \Lambda, ?, \text{nop}, 1)$	(transition for even palindrome)
$(0, a, ?, \text{nop}, 1)$	(transition for odd palindrome)
$(0, b, ?, \text{nop}, 1)$	(transition for odd palindrome)
$(1, \Lambda, X, \text{pop}, 1)$	
$(1, a, a, \text{pop}, 1)$	
$(1, b, b, \text{pop}, 1)$.	

2. a. The terminals a, b, and c give us the following three PDA instructions:

$(0, a, a, \text{pop}, 0)$
$(0, b, b, \text{pop}, 0)$.
$(0, c, c, \text{pop}, 0)$.

The three productions give us the following three instructions:

Production	*Corresponding PDA instruction*
$S \rightarrow aSb$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(b), \text{push}(S), \text{push}(a) \rangle, 0)$.
$S \rightarrow c$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(c) \rangle, 0)$.
$S \rightarrow \Lambda$	$(0, \Lambda, S, \text{pop}, 0)$.

b. From the terminals a and b we construct the following two PDA instructions:

$(0, a, a, \text{pop}, 0)$
$(0, b, b, \text{pop}, 0)$.

From each production we construct a corresponding PDA instruction as shown in the following table:

Production	*Corresponding PDA instruction*
$S \rightarrow bBS$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(S), \text{push}(B), \text{push}(b) \rangle, 0)$.
$S \rightarrow aB$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(B), \text{push}(a) \rangle, 0)$.
$B \rightarrow bbB$	$(0, \Lambda, B, \langle \text{pop}, \text{push}(B), \text{push}(b), \text{push}(b) \rangle, 0)$.
$B \rightarrow \Lambda$	$(0, \Lambda, B, \text{pop}, 0)$.

c. The terminals a, b, and c give us the following three PDA instructions:

$(0, a, a, \text{pop}, 0)$
$(0, b, b, \text{pop}, 0)$.
$(0, c, c, \text{pop}, 0)$.

From each production we construct a corresponding PDA instruction as shown in the following table:

Production	*Corresponding PDA instruction*
$S \to aSb$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(b), \text{push}(S), \text{push}(a)\rangle, 0)$.
$S \to BC$	$(0, \Lambda, S, \langle \text{pop}, \text{push}(C), \text{push}(B)\rangle, 0)$.
$B \to bB$	$(0, \Lambda, B, \langle \text{pop}, \text{push}(B), \text{push}(b)\rangle, 0)$.
$B \to b$	$(0, \Lambda, B, \langle \text{pop}, \text{push}(b)\rangle, 0)$.
$C \to cC$	$(0, \Lambda, C, \langle \text{pop}, \text{push}(C), \text{push}(c)\rangle, 0)$.
$C \to \Lambda$	$(0, \Lambda, C, \text{pop}, 0)$.

3. The following instructions will do the job:

$(0, \Lambda, S, \text{pop}, 1)$
$(1, \Lambda, a, \text{push}(b), 2)$
$(1, \Lambda, b, \text{push}(b), 2)$
$(1, \Lambda, A, \text{push}(b), 2)$
$(1, \Lambda, B, \text{push}(b), 2)$
$(1, \Lambda, S, \text{push}(b), 2)$
$(1, \Lambda, X, \text{push}(b), 2)$
$(2, \Lambda, b, \text{push}(S), 3)$
$(3, \Lambda, S, \text{push}(a), 0)$.

4. a. Since the starting stack symbol is X and the start state is 0, the first productions are

$S \to X_{00} \mid X_{01}$

Next, we'll list the productions that are constructed from each of the PDA instructions.

PDA Instruction	*Corresponding Grammar Productions*
$(1, b, X, \text{pop}, 1)$	$X_{11} \to b$
$(1, b, A, \text{pop}, 1)$	$A_{11} \to b$
$(0, a, A, \text{nop}, 1)$	Productions $A_{0i} \to aA_{1i}$ for each state i: $A_{00} \to aA_{10}$ $A_{01} \to aA_{11}$
$(0, a, X, \text{push}(A), 0)$	Productions $X_{0j} \to aA_{0i}X_{ij}$ for all states i and j: $X_{00} \to aA_{00}X_{00} \mid aA_{01}X_{10}$ $X_{01} \to aA_{00}X_{01} \mid aA_{01}X_{11}$
$(1, a, A, \text{push}(A), 1)$	Productions $A_{1j} \to aA_{1i}A_{ij}$ for all states i and j: $A_{10} \to aA_{10}A_{00} \mid aA_{11}A_{10}$ $A_{11} \to aA_{10}A_{01} \mid aA_{11}A_{11}$

b. Notice that X_{10} does not occur on the left side of any production. So we can omit the production $X_{00} \rightarrow aA_{01}X_{10}$. Notice that each of the following productions is recursive but does not have a basis case:

$X_{00} \rightarrow aA_{00}X_{00}$
$X_{01} \rightarrow aA_{00}X_{01}$
$A_{10} \rightarrow aA_{10}A_{00}$
$A_{10} \rightarrow aA_{11}A_{10}$

Therefore these four productions can be omitted. Now we can omit the production $S \rightarrow X_{00}$ and because X_{00} does not appear on the right side of any remaining production and thus can't be reached in a derivation. Similarly, we can omit the productions $A_{00} \rightarrow aA_{10}$ and $A_{11} \rightarrow aA_{10}A_{01}$ because A_{10} does not appear on the left side of any production. The remaining productions can be listed as follows:

$S \rightarrow X_{01}$
$X_{11} \rightarrow b$
$A_{11} \rightarrow b$
$A_{01} \rightarrow aA_{11}$
$X_{01} \rightarrow aA_{01}X_{11}$
$A_{11} \rightarrow aA_{11}A_{11}$

Since there are no alternative choices for X_{11}, A_{01} and X_{01}, we can replace the productions $X_{11} \rightarrow b$, $A_{01} \rightarrow aA_{11}$, and $X_{01} \rightarrow aA_{01}X_{11}$ by substitution to obtain the following productions:

$S \rightarrow aaA_{11}b$
$A_{11} \rightarrow aA_{11}A_{11} \mid b$

Now we'll replace A_{11} by T to obtain the following grammar:

$S \rightarrow aaTb$
$T \rightarrow aTT \mid b$.

c. The language is the set of all strings over $\{a, b\}$ that have the following properties:

1. Each string has the same number of a's and b's.
2. Each string starts with aa and ends with bb.
3. For each occurrence of a after the prefix aa, there are at least two more a's than b's to its left.

12.3 Context-Free Parsing

Learning Objectives

Be able to describe LL(k) grammars.

Be able to perform factorization, if possible, to reduce the size of k.

Be able to write recursive descent procedures.

Be familiar with the parsing process for simple LL(1) grammars.

Be able to describe the properties of LR(k) grammars.

Be familiar with the parsing process for LR(1) grammars.

Review Questions

1. What is a deterministic context-free language?
2. What is top-down parsing?
3. What is bottom-up parsing?
4. What is an LL(k) grammar?
5. What is recursive descent parsing?
6. What are the characteristics of an LR(k) grammar?
7. What is a handle?
8. What is the meaning of the expression ($A \rightarrow y$, p)?

Solved Problems

1. Given the following LL(k) grammar:

   $$S \rightarrow aaS \mid ab \mid b.$$

   a. What is the smallest value of k for this grammar?

   b. Transform the LL(k) grammar into an LL(1) grammar.

2. Given the following LL(k) grammar:

   $S \rightarrow abA \mid abbB$
   $A \rightarrow \Lambda \mid aA$
   $B \rightarrow b \mid bB$.

   a. What is the smallest value of k for this grammar?

   b. Transform the LL(k) grammar into an LL(1) grammar.

3. Remove the left recursion from each of the following grammars and note whether the resulting grammar is LL(k).

   a. $S \rightarrow SaC \mid SbC \mid C \mid Cb \mid a$
      $C \rightarrow c \mid cC$.

   b. $S \rightarrow SaB \mid Bc$
      $B \rightarrow BbC \mid C$
      $C \rightarrow Cc \mid \Lambda$.

   c. $S \rightarrow Aaa \mid aB \mid c$
      $A \rightarrow Sb \mid b$
      $B \rightarrow bB \mid \Lambda$.

4. Write down the recursive descent procedures to parse strings in the language defined by the following LL(1) grammar.

   $S \rightarrow aSb \mid bBCc$
   $B \rightarrow bB \mid \Lambda$
   $C \rightarrow cC \mid d$.

5. Given the following LL(1) parse table:

	a	b	c	d	\$
S	$S \rightarrow aSbT$	$S \rightarrow TR$	$S \rightarrow TR$	$S \rightarrow TR$	
T		$T \rightarrow \Lambda$	$T \rightarrow cT$	$T \rightarrow \Lambda$	$T \rightarrow \Lambda$
R		$R \rightarrow b$		$R \rightarrow dR$	

   Use the table to parse the string *acdbbc* by showing at each step the contents of the stack, the input, and the actions to perform.

6. Describe for any $k \geq 1$ an LL($k + 1$) grammar for $\{a^n b \mid n \in \mathbb{N}\}$ that is not LL(k).

7. Given the following LR(0) grammar.

   $S \rightarrow aTb \mid c$
   $T \rightarrow aSb \mid c$.

   Find the handle for each of the following sentential forms.

   a. *aacbb*. b. *aaacbbb*. c. *aaSbb*. d. *aaaTbbb*.

8. Given the following LR(1) grammar.

   $S \rightarrow dSB \mid cAb$
   $A \rightarrow aA \mid a$
   $B \rightarrow b$.

   Complete the following table of sentential forms and handles, where the handle is used to determine a rightmost derivation in reverse.

Sentential Form	*Handle*
ddcaaabbb	($S \rightarrow a$, 6)

Solutions

1. a. $k = 2$. To see this, notice that two letters of lookahead are sufficient to determine which production to use for a derivation step.

   b. The two productions $S \rightarrow aaS \mid ab$ can be factored to give the following LL(1) grammar fragment:

   $S \rightarrow aT$
   $T \rightarrow aS \mid b$.

   So the desired LL(1) grammar is

   $S \rightarrow aT \mid b$
   $T \rightarrow b \mid aS$.

2. a. $k = 3$. To see this, notice that three letters of lookahead are sufficient to determine which production to use for a derivation step.

   b. The two productions $S \rightarrow abA \mid abbB$ can be factored to give the following LL(1) grammar fragment:

   $S \rightarrow abT$
   $T \rightarrow A \mid bB$.

   The two productions $B \rightarrow b \mid bB$ can also be factored to give the following LL(1) fragment:

   $B \rightarrow bC$
   $C \rightarrow \Lambda \mid B$.

So the transformed LL(1) grammar can be written as follows:

$S \rightarrow abT$
$T \rightarrow A \mid bB$
$B \rightarrow bC$
$A \rightarrow \Lambda \mid aA$
$C \rightarrow \Lambda \mid B$.

3. a. $S \rightarrow CT \mid CbT \mid aT$
$T \rightarrow aCT \mid bCT \mid \Lambda$
$C \rightarrow c \mid cC$.

   b. $S \rightarrow BcT$
$T \rightarrow aBT \mid \Lambda$
$B \rightarrow CR$
$R \rightarrow bCR \mid \Lambda$
$C \rightarrow Cc \mid \Lambda$.

   c. We can remove the indirect left recursion by replacing A in $S \rightarrow Aaa$ by $A \rightarrow Sb \mid b$ to obtain the following grammar.

$S \rightarrow Sbaa \mid baa \mid aB \mid c$
$B \rightarrow bB \mid \Lambda$.

Now remove the direct left recursion to obtain the following grammar.

$S \rightarrow baaT \mid aBT \mid cT$
$T \rightarrow baaT \mid \Lambda$
$B \rightarrow bB \mid \Lambda$.

4. The procedures for S, B, and C can be written as follows:

```
procedure S
    if lookahead = a then
        match(a);
        S;
        match(b)
    else
        match(b)
        B;
        C;
        match(c)
    fi

procedure B
    if lookahead = b then
        match(b);
        B
    fi
```

```
procedure C
    if lookahead = c then
        match(c);
        C
    else
        match(d)
    fi
```

5.

Stack	Input	Actions to Perform
$*S*	*acdbbc*$	pop, push *T*, push *b*, push *S*, push *a*
$*TbSa*	*acdbbc*$	pop, consume
$*TbS*	*cdbbc*$	pop, push *R*, push *T*
$*TbRT*	*cdbbc*$	pop, push *T*, push *c*
$*TbRTc*	*cdbbc*$	pop, consume
$*TbRT*	*dbbc*$	pop
$*TbR*	*dbbc*$	pop, push *R*, push *d*
$*TbRd*	*dbbc*$	pop, consume
$*TbR*	*bbc*$	pop, push *b*
$*Tbb*	*bbc*$	pop, consume
$*Tb*	*bc*$	pop, consume
$*T*	*c*$	pop, push *T*, push *c*
$*Tc*	*c*$	pop, consume
$*T*	$	pop
$	$	accept.

6. Let $L = \{a^n b \mid n \in \mathbb{N}\}$. To get the idea, we'll write a couple of examples. The following grammar for *L* is LL(2) but not LL(1):

   $S \rightarrow b \mid ab \mid aaS.$

   For another example, the following grammar for *L* is LL(3) but not LL(2):

   $S \rightarrow b \mid ab \mid aab \mid aaaS.$

   So a grammar for *L* that is LL(k + 1) but not LL(k) takes the following form, where *a*...*aS* is a sentential form with a string of k *a*'s followed by *S*.

   $S \rightarrow b \mid ab \mid aab \mid ... \mid a...aS.$

7. a. $(S \rightarrow c, 3)$.

   b. $(T \rightarrow c, 4)$.

   c. $(T \rightarrow aSb, 4)$.

   d. $(S \rightarrow aTb, 5)$.

8.

Sentential Form	Handle
ddcaaabbb	($A \rightarrow a$, 6)
ddcaaAbbb	($A \rightarrow aA$, 6)
ddcaAbbb	($A \rightarrow aA$, 5)
ddcAbbb	($S \rightarrow cAb$, 5)
ddSbb	($B \rightarrow b$, 4)
ddSBb	($S \rightarrow dSB$, 4)
dSb	($B \rightarrow b$, 3)
dSB	($S \rightarrow dSB$, 3)
S.	

12.4 Context-Free Language Topics

Learning Objectives

Be able to transform grammars by removing all left recursion and by removing all possible productions that have the empty string on the right side.

Be able to describe and to apply the pumping lemma for context-free languages.

Review Questions

1. How do you remove Λ-productions from a grammar?

2. What is the Chomsky normal form?

3. What is the Greibach normal form?

4. What is the pumping lemma for context-free languages?

5. How is the pumping lemma used?

Solved Problems

1. Transform the following grammar into an equivalent grammar that does not have Λ-productions.

$S \rightarrow aAB \mid bC$
$A \rightarrow aA \mid \Lambda$
$B \rightarrow bB \mid \Lambda$
$C \rightarrow A \mid B \mid cC \mid \Lambda$.

2. Transform the following grammar into an equivalent grammar in Chomsky normal form.

$$S \to bABc \mid cBA$$
$$A \to aA \mid a$$
$$B \to bcB \mid \Lambda.$$

3. Transform the following grammar into an equivalent grammar in Greibach normal form.

$$S \to ABc \mid cBA$$
$$A \to Aab \mid a$$
$$B \to bcB \mid \Lambda.$$

4. Outline a proof that uses the pumping lemma for context-free languages to show that the language $\{(ab)^n c^n a^n \mid n \in \mathbb{N}\}$ is not context-free.

5. Outline a proof that uses morphisms for context-free languages to show that the language $\{(ab)^n c^n d^n \mid n \in \mathbb{N}\}$ is not context-free.

Solutions

1. The nonterminals that derive Λ are A, B, and C. So we construct new productions by removing one or more occurrences of A, B, and C from the right sides of existing productions. This process gives us the following new productions:

$$S \to aB \mid aA \mid a \mid b$$
$$A \to a$$
$$B \to b$$
$$C \to c.$$

Now we combine this productions with the original grammar but without the original Λ-productions to obtain the following desired grammar:

$$S \to aAB \mid bC \mid aB \mid aA \mid a \mid b$$
$$A \to aA \mid a$$
$$B \to bB \mid b$$
$$C \to A \mid B \mid cC \mid c.$$

2. First, we need to remove all Λ-productions. Since B is the only nonterminal that derives Λ, we add the following new productions by removing B from the right sides of existing productions:

$$S \to bAc \mid cA$$
$$B \to bc$$

Combining these productions with the original productions without the production $B \to \Lambda$, we obtain the following grammar:

$$S \to bABc \mid cBA \mid bAc \mid cA$$
$$A \to aA \mid a$$
$$B \to bcB \mid bc.$$

Now we introduce new nonterminals to replace the terminals a, b, and c that occur on right sides with two or more symbols. This gives us the following grammar:

$S \rightarrow DABC \mid CBA \mid DAC \mid CA$
$A \rightarrow EA \mid a$
$B \rightarrow DCB \mid DC$
$C \rightarrow c$
$D \rightarrow b$
$E \rightarrow a.$

Lastly, introduce new nonterminals so that no right side has more than two nonterminals. This gives the following grammar:

$S \rightarrow DF \mid CH \mid DK \mid CA$
$F \rightarrow AG$
$G \rightarrow BC$
$H \rightarrow BA$
$K \rightarrow AC$
$A \rightarrow EA \mid a$
$B \rightarrow DL \mid DC$
$L \rightarrow CB$
$C \rightarrow c$
$D \rightarrow b$
$E \rightarrow a.$

3. First we must remove all left recursion, which gives us the following grammar:

$S \rightarrow ABc \mid cBA$
$A \rightarrow aC$
$C \rightarrow abC \mid \Lambda$
$B \rightarrow bcB \mid \Lambda.$

Next, we must remove all Λ-productions to obtain the following grammar:

$S \rightarrow ABc \mid cBA \mid Ac \mid cA$
$A \rightarrow aC \mid a$
$C \rightarrow abC \mid ab$
$B \rightarrow bcB \mid bc.$

Now we need to make substitutions to get each production into the proper form. We'll use the productions $A \rightarrow aC \mid a$ to replace A in $S \rightarrow ABc$ to obtain $S \rightarrow aCBc \mid aBc$. We'll also use $A \rightarrow aC \mid a$ to replace A in $S \rightarrow Ac$ to obtain $S \rightarrow aCc \mid ac$. This gives us the following grammar:

$S \rightarrow aCBc \mid aBc \mid cBA \mid aCc \mid ac \mid cA$
$A \rightarrow aC \mid a$
$C \rightarrow abC \mid ab$
$B \rightarrow bcB \mid bc.$

Now we can introduce new nonterminals to replace terminals that don't appear on the left end of a right side. This gives the following desired grammar:

$S \to aCBD \mid aBD \mid cBA \mid aCD \mid aD \mid cA$
$A \to aC \mid a$
$C \to aEC \mid aE$
$B \to bDB \mid bD$
$D \to c$
$E \to b.$

4. Let L be the language and assume, by way of contradiction, that L is context-free. Then we can apply the pumping lemma (12.19). Let $z = (ab)^m c^m a^m \in L$, where m is the integer given by the pumping lemma. Then z has the form $z = (ab)^m c^m a^m = uvwxy$, where $|vx| \geq 1$, $|vwx| \leq m$, and $uv^k wx^k y \in L$ for all $k \geq 0$.

   Now we can try to find contradictions by considering the structure of v and x. Since $|vx| \geq 1$, it follows that at least one of v and x is not the empty string. Since $|vwx| \leq m$, it follows that v and x can't have the form $...ab...c...a...$ because any substring of z having this form has length greater than m. Another reason for this is that if v or x has the form $...ab...c...a...$ then v^2 or x^2 has the form

   $$...ab...c...a.....ab...c...a... .$$

   But this pattern can't occur as a substring of any element of L, contrary to $uv^2wx^2y \in L$.

   We also obtain contradictions if we assume that v or x has one of the forms $(ab)^i c^j$ or $c^j a^k$, where $i > 0$, $j > 0$, and $k > 0$. In this case, either v^2 or x^2 has the form $(ab)^i c^j (ab)^i c^j$ or $c^j a^k c^j a^k$. But neither of these patterns can occur as a substring of any element of L, contrary to the pumping property $uv^2wx^2y \in L$.

   Now we're left with the possibility that v or x has one of the forms $(ab)^i$, c^j, or a^k, where $i > 0$, $j > 0$, and $k > 0$. In these cases, the string uv^2wx^2y would not be in L because one of the three substrings $(ab)^i$, c^j, and a^k would be missing. For example, if $v = (ab)^i$ and $x = c^j$, then $uv^2wx^2y = (ab)^{m+i} c^{m+j} a^m$, which is not in L.

   We have obtained contradictions for all possible choices for the form of v and x. Therefore our original assumption was wrong and it follows that L is not context-free.

5. Define the function $f : \{a, b, c\}^* \to \{a, b, c, d\}^*$ as follows: for any string $s \in \{a, b, c\}^*$, let $f(s)$ be defined by applying the following rules to each letter of s:

   $$f(\Lambda) = \Lambda,\ f(a) = ab,\ f(b) = c, \text{ and } f(c) = d.$$

   This definition makes f into a morphism and we can apply the results of (12.23). Let $L = \{(ab)^n c^n d^n \mid n \in \mathbb{N}\}$. Then $f^{-1}(L) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ and we know this language is not context-free by Example 12.24. Now if L is context-free, then (12.23) tells us that $f^{-1}(L)$ is context-free, which is a contradiction. Therefore L is not context-free.

Chapter 13

Turing Machines and Equivalent Models

13.1 Turing Machines

Learning Objectives

Be able to describe a Turing machine.

Be able to write Turing machines (single-tape and muli-tape) to solve simple problems.

Review Questions

1. What is a Turing machine?

2. What is the meaning of the expression (i, a, b, L, j)?

3. How does a Turing machine recognize a string?

4. Is there a difference in the power of deterministic Turing machines and nondeterministic Turing machines?

Solved Problems

1. Write down the instructions for a single-tape Turing machine that accepts the language $\{ab^n ac^n \mid n \in \mathbb{N}\}$.
2. Write down the instructions for a single-tape Turing machine that searches for the leftmost occurrence of the string *abc* on the tape. Assume that the tape contains a string over the alphabet $\{a, b, c\}$. If *abc* is found, the machine halts with a 1 in the current cell. Otherwise the machine halts with a 0 in the current cell.
3. Construct a 2-tape to test divisibility of nonzero natural numbers represented in unary notation. Assume that the two tapes contain the numbers A and B with the tape heads initially at the left end of each number. If A divides B, then write Y in the current cell of A and if not, then write N in the current cell of A.

4. Construct a 2-tape Turing machine to test the relationship between two binary numbers. Assume that the tapes contain binary representations of the two numbers A and B, and the tape heads are initially at the rightmost bits of the numbers. The Turing machine should halt with the current pair of cells containing one of the following three values with the associated meanings.

   (A, B) means $A > B$
   (B, A) means $A < B$
   (E, E) means $A = B$.

Solutions

1. Assume that the tape head is at the left end of the input string. The following Turing machine keeps track of the b's and c's by replacing them with X's and Y's to indicate that a pair has been matched.

   Start by checking whether the first letter is an a. If so, then look for a b.

   (0, a, a, R, 1)

   If there are b's to mark, then mark the next b and then go find a matching c.

   (1, b, X, R, 2)
   (1, a, a, R, 5) Done with b's. Go check to see that c's are done.

   Scan right to find a c to mark.

   (2, b, b, R, 2)
   (2, a, a, R, 3)
   (3, Y, Y, R, 3)
   (3, c, Y, L, 4)

   Scan left to find another b to mark.

   (4, a, a, L, 4)
   (4, Y, Y, L, 4)
   (4, b, b, L, 4)
   (4, X, X, R, 1)

   Final check to see that there are no more c's than b's.

   (5, Y, Y, R, 5)
   (5, Λ, Λ, S, halt) Success.

2. Look for the letter a.

   (0, a, a, R, 1)
   (0, b, b, R, 0)
   (0, c, c, R, 0)
   (0, Λ, 0, S, Halt) *abc* does not occur in the string.

Look for the letter *b* in a substring *ab*.

(1, *a*, *a*, *R*, 1)
(1, *b*, *b*, *R*, 2)
(1, *c*, *c*, *R*, 0)
(1, Λ, 0, *S*, Halt) *abc* does not occur in the string.

Look for the letter *c* in a substring *abc*.

(2, *a*, *a*, *R*, 1)
(2, *b*, *b*, *R*, 0)
(2, *c*, 1, *S*, Halt) Found leftmost substring *abc*.
(2, Λ, 0, *S*, Halt) *abc* does not occur in the string.

3. Scan right on both tapes until the end of either number is reached. If the end of *A* is reached, reset the tape head for *A* to the left end and scan right on both tapes as before. Continue this process until both ends are reached at the same time, which means that *B* is a multiple of *A* (i.e., *A* divides *B*), or until the end of *B* is reached without reaching the end of *A*, which means that *B* is not a multiple of *A* (i.e., *A* does not divide *B*). Here is the algorithm, where (*x*, *y*) means that *x* is for *A* and *y* is *B*.

   Scan right to see if *B* is a multiple of *A*.

   (0, (1, 1), (1, 1), (*R*, *R*), 0)
   (0, (1, Λ), (*N*, Λ), (*S*, *S*), Halt)
   (0, (Λ, 1), (Λ, 1), (*L*, *S*), 1)
   (0, (Λ, Λ), (*Y*, Λ), (*S*, *S*), Halt)

   Reset the tape head for *A* to left end.

   (1, (1, 1), (1, 1), (*L*, *S*), 1)
   (1, (Λ, 1), (Λ, 1), (*R*, *S*), 0).

4. Here is a 3-state solution, where each state represents the current relationship of the numbers. The main idea is to compare two bits, move both tape heads left, and enter the appropriate state. Repeat this process until an ending condition involving Λ is found.

   The equality state

   (0, (0, 0), (0, 0), (*L*, *L*), 0)
   (0, (1, 1), (1, 1), (*L*, *L*), 0)
   (0, (0, 1), (0, 1), (*L*, *L*), 1)
   (0, (1, 0), (1, 0), (*L*, *L*), 2)
   (0, (0, Λ), (0, Λ), (*L*, *S*), 0)
   (0, (Λ, 0), (Λ, 0), (*S*, *L*), 0)
   (0, (1, Λ), (*A*, *B*), (*L*, *S*), Halt) $A > B$
   (0, (Λ, 1), (*B*, *A*), (*S*, *L*), Halt) $A < B$
   (0, (Λ, Λ), (*E*, *E*), (*S*, *S*), Halt) $A = B$

The less-than state (first < second)

(1, (0, 0), (0, 0), (*L*, *L*), 1)
(1, (1, 1), (1, 1), (*L*, *L*), 1)
(1, (0, 1), (0, 1), (*L*, *L*), 1)
(1, (1, 0), (1, 0), (*L*, *L*), 2)
(1, (0, Λ), (0, Λ), (*L*, *S*), 1)
(1, (Λ, 0), (*B*, *A*), (*S*, *S*), Halt) $A < B$
(1, (1, Λ), (*A*, *B*), (*L*, *S*), Halt) $A > B$
(1, (Λ, 1), (*B*, *A*), (*S*, *L*), Halt) $A < B$
(1, (Λ, Λ), (*E*, *E*), (*S*, *S*), Halt) $A < B$

The greater-than state (first > second)

(2, (0, 0), (0, 0), (*L*, *L*), 2)
(2, (1, 1), (1, 1), (*L*, *L*), 2)
(2, (0, 1), (0, 1), (*L*, *L*), 1)
(2, (1, 0), (1, 0), (*L*, *L*), 2)
(2, (0, Λ), (0, Λ), (*L*, *S*), Halt) $A > B$
(2, (Λ, 0), (Λ, 0), (*S*, *L*), 2)
(2, (1, Λ), (*A*, *B*), (*L*, *S*), Halt) $A > B$
(2, (Λ, 1), (*B*, *A*), (*S*, *L*), Halt) $A < B$
(2, (Λ, Λ), (*E*, *E*), (*S*, *S*), Halt) $A < B$.

13.2 The Church-Turing Thesis

Learning Objectives

Be able to state the Church-Turing Thesis.

Be able to solve simple problems with each of the following models of computation: simple while-loop programs; partial recursive functions; Markov algorithms; Post algorithms; and Post systems.

Review Questions

1. What does the Church-Turing thesis say?

2. What is a partial recursive function?

3. What is a Markov algorithm?

4. What does the Markov expression $x \rightarrow y$ mean?

5. What is a Post algorithm?

6. What does the Post expression $x \rightarrow y$ mean?

7. What is a Post system?

8. What does the Post system expression $s, t \rightarrow u$ mean?

Solved Problems

1. Write a simple program to test whether one natural number divides another. Assume that the output values 0 and 1 denote False and True, respectively. You may use macros from either the text or exercises in Section 13.2 of the book.
2. Show that multiplication is a partial recursive function by defining it by primitive recursion and using the add operation from Figure 13.3 of the book.
3. Show that each of the following logic functions is partial recursive by writing it in terms of known partial recursive functions. Assume that P and Q return the values 0 and 1, which denote False and True, respectively. You may use any previously defined partial recursive function.
   a. $P(x) \wedge Q(x)$.
   b. $\neg P(x)$.
   c. $P(x) \vee Q(x)$.
4. Suppose the function f is defined by the following if-then-else statement:

   $$f(x) = \text{if } B(x) \text{ then } C(x) \text{ else } D(x),$$

   where B, C, and D are partial recursive functions and $B(x)$ takes only the values 0 and 1, which we interpret as False and True, respectively. Show that f is partial recursive. You may use any previously defined partial recursive function from the text or these problems.
5. Given the function

   $$f(x, y) = \min(z, \text{monus}(y, \text{mult}(x, z)) = 0).$$

   Try to describe f for each of the following conditions.
   a. $f(0, y)$.
   b. $f(x, 0)$.
   c. $f(x, y)$ when $0 < x < y$.
   d. $f(x, y)$ when $0 < y < x$.
6. Find a Markov algorithm to transform any string of the form $a^n c$ into ca^n, where $n \in \mathbb{N}$.
7. Find a Markov algorithm that transforms any string of the form a^n for $n \in \mathbb{N}$ into a string of the form $a^n b$.

8. Find a Markov algorithm to transform any string of the form $a^n b^m$ into a string of the form $b^n a^m$, where $m, n \in \mathbb{N}$.
9. Find a Post algorithm to transform any string of the form $a^n c$ into ca^n, where $n \in \mathbb{N}$.
10. Find a Post algorithm that transforms any string of the form $a^m b^n$, for $m, n \in \mathbb{N}$, into a string of the form $a^m c b^n$.
11. Find a Post algorithm to transform any string of the form $a^n c$ into $c^n a$, where $n \in \mathbb{N}$.
12. Find a Post algorithm to transform any string of the form $a^n b^m$ into a string of the form $b^n a^m$, where $m, n \in \mathbb{N}$.
13. Find a Post system that generates the language $\{(ab)^m c a^n \mid m, n \in \mathbb{N}\}$.
14. Find a Post system that generates the language $\{a^n b c^n \mid n \in \mathbb{N}\} \cup \{c^n b a^n \mid n \in \mathbb{N}\}$.

Solutions

1. The algorithm will test whether A divides B by finding out if some multiple of A can be subtracted from B to yield the value zero. The output of the program will be contained in the variable Z. The first if-then statement makes sure that divisors can only be positive. We'll use macros that are already defined. Here is the algorithm.

```
if A ≠ 0 then
    while A < B do
        B := B monus A
    od
else
    A := B + 1;
fi;
X := A monus B;
if X ≠ 0 then Z := 0 else Z := 1 fi.
```

2. Let mult(x, y) denote the product of two natural numbers x and y. Then mult has the following definition:

   mult(x, 0) = 0
   mult(x, succ(y)) = add(x, mult(x, y)).

3. a. $P(x) \wedge Q(x) = \text{mult}(P(x), Q(x))$.

   b. $\neg P(x) = \text{monus}(1, P(x))$

   c. $P(x) \vee Q(x) = \text{add}(P(x), \text{mult}(\text{monus}(1, P(x)), Q(x)))$.

4. We can define f informally as follows:

   $f(x) = B(x)C(x) + (1 - B(x))D(x)$.

   This definition can be represented by the following composition of known partial recursive functions.

   $f(x) = \text{add}(\text{mult}(B(x), C(x)), \text{mult}(\text{monus}(1, B(x)), D(x)))$.

5. a. $f(0, y)$ = if $y = 0$ then 0 else ∞.

   b. $f(x, 0) = 0$.

   c. If $0 < x < y$, then $f(x, y)$ = ceiling(y/x).

   d. If $0 < y < x$, then $f(x, y) = 1$, which is also ceiling(y/x).

6. One instruction will do the job.

   1. $ac \rightarrow ca$.

7. One solution is to attach a symbol x to the left end of the input string. Then move x to the right until it reaches b and remove them both. Here is the algorithm.

   1. $xa \rightarrow ax$
   2. $xb \rightarrow \Lambda$ (halt)

8. One solution is to attach a symbol x to the left end of the input string. Then move x to the right, while changing a's to b's and b's to a's. Here is the algorithm.

   1. $xa \rightarrow bx$
   2. $xb \rightarrow xa$
   3. $x \rightarrow a$ (halt)
   4. $\Lambda \rightarrow x$.

9. One instruction will do the job.

   1. $Xc \rightarrow cX$.

10. Find the point where a meets b and place c between them.

    $XabY \rightarrow XacbY$
    $Xa \rightarrow Xac$
    $bX \rightarrow cbX$.

11. One deterministic solution is to place markers around the string of a's and replace the c by a. Then replace each a by c, always keeping the markers around the a's that are yet to be replaced. Then remove the markers when there is nothing between them. Here's the algorithm.

    $Xc \rightarrow @X\#a$

    $@aX\#a \rightarrow c@X\#a$

    $X@aY\#a \rightarrow Xc@Y\#a$

    $X@\#a \rightarrow Xa$ (halt).

12. One solution places two markers at the left end and then moves one of them to the right end, changing *a*'s to *b*'s and *b*'s to *a*'s as it goes. Here is the algorithm.

$aX \rightarrow @\#aX$
$bX \rightarrow @\#bX$
$X\#aY \rightarrow Xb\#Y$
$X\#bY \rightarrow Xa\#Y$
$@X\# \rightarrow X$ (halt).

13. Axiom: *c*.
Inference rules: $XcY \rightarrow XabcY$
$XcY \rightarrow XcaY$.

14. Axioms: *b*, *abc*, *cba*.
Inference rules: $XabcY \rightarrow XaabccY$
$XcbaY \rightarrow XccbaaY$

Chapter 14

Computational Notions

14.1 Computability

Learning Objectives

Be able to describe the concepts of undecidable and partially decidable.

Be able to state the halting problem and prove that it is undecidable and partially decidable.

Be able to use diagonalization to prove that the set of total computable functions cannot be enumerated.

Review Questions

1. What is a decidable problem?

2. What is a partially decidable problem?

3. What is an undecidable problem?

4. What is the halting problem?

5. What is the total problem?

6. What is the equivalence problem?

7. What is the Post correspondence problem?

Solved Problems

1. Suppose we are given the following list of computable functions, each of which halts on its own index.

   $$f_0, f_1, f_2, ..., f_n, \ldots .$$

   In other words, $f_n(n) \in \mathbb{N}$ for all $n \in \mathbb{N}$. Show that there is some computable function that halts on its own index, but is not in the list.

2. Show that the problem of deciding whether two regular grammars derive the same language is decidable.
3. Show that the problem of deciding whether two regular expressions are equal is decidable.
4. Find a solution to the following instance of the Post correspondence problem, where the tuples are numbered 1, 2, 3, and 4.

$$(a, ab), (a, ba), (baa, a), (b, a).$$

Solutions

1. Let $h(x) = f_x(x) + 1$. Then h is computable and is total, so it must halt on its own index. If h is in the list, then $h = f_n$ for some $n \in \mathbb{N}$. But now we'll obtain a contradiction when we calculate the value $h(n)$. If we use the definition of h, then $h(n) = f_n(n) + 1$. But since $h = f_n$, we also have $h(n) = f_n(n)$. In other words, we have $f_n(n) = f_n(n) + 1$, which is a contradiction. Therefore h halts on its own index and is not in the list.
2. Use (11.12) to transform each grammar into an NFA. Then use (11.8) to transform each NFA into a DFA. Then use (11.10) to transform each DFA into a minimum-state DFA. Now compare the two DFA tables to see whether they have the same number of states and, if so, whether they have the same start and final states (after possibly renaming states), and whether the table entries are the same.
3. Use either (11.4) or (11.7) to transform each regular expression into an NFA. Then proceed in the same way as outlined in the preceding problem.
4. One solution is 1, 4, 2, 3, which produces $ababaa = ababaa$.

14.2 A Hierarchy of Languages

Learning Objectives

Be able to describe the hierarchy of languages and to give an example of a language at each level of the hierarchy that does not belong in a lower level.

Review Questions

1. What is a context-sensitive grammar?
2. What is a monotonic grammar?
3. What is a linear bounded automaton?
4. What is the hierarchy of languages?

Solved Problems

1. Use the following monotonic grammar to write down a derivation for the string *aabaaabb*.

   $S \rightarrow aSbA \mid Ab$
   $A \rightarrow bB \mid a$
   $bA \rightarrow aaAB \mid bb$
   $B \rightarrow bB \mid bb$.

2. The language of the following context-sensitive grammar is actually a regular language.

   $S \rightarrow AS \mid a$
   $Aa \rightarrow aba$.

   Describe the language with a regular grammar.

3. The language of the following monotonic grammar is actually a context-free language.

   $S \rightarrow AS \mid ab$
   $Aa \rightarrow aaB$
   $Ba \rightarrow aB$
   $Bb \rightarrow bb$.

   Describe the language with a context-free grammar.

Solutions

1. Here is a leftmost derivation of the string.

   $S \Rightarrow aSbA \Rightarrow aAbbA \Rightarrow aabbA \Rightarrow aabbaaAB \Rightarrow aabbaaaB \Rightarrow aabbaaabb$.

2. The following few derivations give the pattern for the strings derived by the grammar.

   $S \Rightarrow a$,
   $S \Rightarrow AS \Rightarrow Aa \Rightarrow aba$,
   $S \Rightarrow AS \Rightarrow AAS \Rightarrow AAa \Rightarrow Aaba \Rightarrow ababa$,
   $S \Rightarrow AS \Rightarrow AAS \Rightarrow AAAS \Rightarrow AAAa \Rightarrow AAaba \Rightarrow Aababa \Rightarrow abababa$.

   The grammar derives the language $\{(ab)^n a \mid n \in \mathbb{N}\}$, and a regular grammar for this language can be written as follows:

   $S \rightarrow abS \mid a$.

3. The following few derivations give the pattern for the strings derived by the grammar.

   $S \Rightarrow ab$,
   $S \Rightarrow AS \Rightarrow Aab \Rightarrow aaBb \Rightarrow aabb$,
   $S \Rightarrow AS \Rightarrow AAS \Rightarrow AAab \Rightarrow AaaBb \Rightarrow aaBaBb \Rightarrow aaaBBb$
   $\Rightarrow aaaBbb \Rightarrow aaabbb$,

$$S \Rightarrow AS \Rightarrow AAS \Rightarrow AAAS \Rightarrow AAAab \Rightarrow AAaaBb \Rightarrow AaaBaBb$$
$$\Rightarrow aaBaBaBb \Rightarrow aaaBBaBb \Rightarrow aaaBaBBb \Rightarrow aaaaBBBb$$
$$\Rightarrow aaaaBBbb \Rightarrow aaaaBbbb \Rightarrow aaaabbbb.$$

The grammar derives the language $\{a^n b^n \mid n \geq 1\}$, and a context-free grammar for this language can be written as follows:

$S \rightarrow aSb \mid ab$.

14.3 Complexity Classes

Learning Objectives

Be able to describe the complexity classes *P*, *NP*, and *PSPACE*.

Be able to describe polynomially reducible.

Be able to describe what it means for a class to be *NP*-complete.

Review Questions

1. What is the class *P*?

2. What is the class *NP*?

3. What is the class *PSPACE*?

4. What is the known relationship between the three classes *P*, *NP*, and *PSPACE*?

Solved Problems

1. Transform the following CNF into an equivalent CNF for the 3-satisfiability problem.

$$(a \vee b \vee \neg c \vee d) \wedge (\neg a \vee \neg b \vee c \vee \neg d \vee e).$$

2. Find the truth value for each of the following wffs in Presburger arithmetic.
   a. $\forall x \, \forall y \, (x + y = y + x)$.
   b. $\exists y \, \forall x \, (x + y = x)$.
   c. $\forall x \, \forall y \, \exists z \, (\neg (x = y) \rightarrow (x + z = y))$.
   d. $\forall x \, \forall y \, \forall z \, (x + (y + z) = (x + y) + z)$.
   e. $\forall x \, \forall y \, ((x + x = x) \rightarrow (x + y = y))$.

3. Finish each of the following sentences.
   a. If $p \in P$ and p is polynomially reducible to problem q, then_____.
   b. If p is *NP*-complete and $p \in P$, then_____.

c. If $p \in NP$ and p is intractable, then_____.

d. If p is *NP*-complete and p is intractable, then_____.

4. Label each decision problem with those classes from the following list to which it is known to belong.

P, *NP*, *PSPACE*, Intractable.

a. Traveling salesman's problem.

b. Quantified Boolean formula problem.

c. Presburger arithmetic problem.

d. Generalized regular expression problem.

e. CNF-satisfiability problem.

f. 3-satisfiability problem.

g. 2-satisfiability problem.

5. Which decision problems listed in Problem 4 are *NP*-complete?

Solutions

1. Since the fundamental disjunction $(a \vee b \vee \neg c \vee d)$ has four literals, we create three new variables x_1, x_2, and x_3 and then construct the following CNF:

$$(a \vee b \vee x_1) \wedge (\neg c \vee \neg x_1 \vee x_2) \wedge (d \vee \neg x_2 \vee x_3).$$

Since there are five literals in $(\neg a \vee \neg b \vee c \vee \neg d \vee e)$, we create four new variables y_1, y_2, y_3 , and y_4 and then construct the following CNF:

$$(\neg a \vee \neg b \vee y_1) \wedge (c \vee \neg y_1 \vee y_2) \wedge (\neg d \vee \neg y_2 \vee y_3) \wedge (e \vee \neg y_3 \vee y_4).$$

The desired CNF is the conjunction of the preceding two CNFs.

2. a. True, because of the commutative law for addition.

b. True, because $y = 0$ is the identity for addition.

c. False. For example, no natural number z satisfies $1 + z = 0$.

d. True, because of the associative law for addition.

e. True. To see this, notice that if $x + x = x$, then $x = 0$. For if $x > 0$, then it follows that $x + x > x$, which is contrary to hypothesis. So if $x + x = x$, it follows that $x = 0$ and thus $x + y = 0 + y = y$.

3. a. $q \in P$.

b. $P = NP$.

c. $P \neq NP$.

d. $P \neq NP$ and all *NP*-complete problems are intractable.

4. a. *NP*. b. PSPACE. c. Intractable. d. Intractable. e. *NP*. f. *NP*. g. *P*.

5. Traveling salesman, CNF-satisfiability, and 3-satisfiability.