Rules of Inference

Section 1.6

Section Summary

- Valid Arguments
- Inference Rules for Propositional Logic
- Using Rules of Inference to Build Arguments
- Rules of Inference for Quantified Statements
- Building Arguments for Quantified Statements

Revisiting the Socrates Example

- We have the two premises:
  - “All men are mortal.”
  - “Socrates is a man.”
- And the conclusion:
  - “Socrates is mortal.”
- How do we get the conclusion from the premises?

The Argument

- We can express the premises (above the line) and the conclusion (below the line) in predicate logic as an argument:

$$\forall x(Man(x) \rightarrow Mortal(x))$$

$$Man(Socrates)$$

$$\therefore \quad Mortal(Socrates)$$

- We will see shortly that this is a valid argument.

Valid Arguments

- We will show how to construct valid arguments in two stages; first for propositional logic and then for predicate logic. The rules of inference are the essential building block in the construction of valid arguments.

1. Propositional Logic

   Inference Rules

2. Predicate Logic

   Inference rules for propositional logic plus additional inference rules to handle variables and quantifiers.

Arguments in Propositional Logic

- A *argument* in propositional logic is a sequence of propositions. All but the final proposition are called *premises*. The last statement is the *conclusion*.
- The argument is valid if the premises imply the conclusion. An *argument form* is an argument that is valid no matter what propositions are substituted into its propositional variables.
- If the premises are $p_1, p_2, \ldots, p_n$ and the conclusion is q then $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow q$ is a tautology.
- Inference rules are all argument simple argument forms that will be used to construct more complex argument forms.

Rules of Inference for Propositional Logic: Modus Ponens

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

Corresponding Tautology:

$(p \wedge (p \rightarrow q)) \rightarrow q$

Example:

Let p be "It is snowing."

Let q be "I will study discrete math."

"If it is snowing, then I will study discrete math."

"It is snowing."

"Therefore , I will study discrete math."

Modus Tollens

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Corresponding Tautology:
$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$

Example:
Let p be "it is snowing."
Let q be "I will study discrete math."

"If it is snowing, then I will study discrete math."
"I will not study discrete math."

"Therefore , it is not snowing."

Hypothetical Syllogism

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Corresponding Tautology:
$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$

Example:
Let p be "it snows."
Let q be "I will study discrete math."
Let r be "I will get an A."

"If it snows, then I will study discrete math."
"If I study discrete math, I will get an A."

"Therefore , If it snows, I will get an A."

Disjunctive Syllogism

$$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$

Corresponding Tautology:
$(\neg p \wedge (p \vee q)) \rightarrow q$

Example:
Let p be "I will study discrete math."
Let q be "I will study English literature."

"I will study discrete math or I will study English literature."
"I will not study discrete math."

"Therefore , I will study English literature."

Addition

$$\frac{p}{\therefore p \vee q}$$

Corresponding Tautology:
$p \rightarrow (p \vee q)$

Example:
Let p be "I will study discrete math."
Let q be "I will visit Las Vegas."

"I will study discrete math."

"Therefore, I will study discrete math or I will visit Las Vegas."

Simplification

$$\frac{p \wedge q}{\therefore q}$$

Corresponding Tautology:
$(p \wedge q) \rightarrow p$

Example:
Let p be "I will study discrete math."
Let q be "I will study English literature."

"I will study discrete math and English literature"

"Therefore, I will study discrete math."

Conjunction

$$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$$

Corresponding Tautology:
$((p) \wedge (q)) \rightarrow (p \wedge q)$

Example:
Let p be "I will study discrete math."
Let q be "I will study English literature."

"I will study discrete math."
"I will study English literature."

"Therefore, I will study discrete math and I will study English literature."

Resolution

Resolution plays an important role in AI and is used in Prolog.

$$\begin{array}{l} \neg p \vee r \\ p \vee q \\ \hline \therefore q \vee r \end{array}$$

Corresponding Tautology:
$((\neg p \vee r) \wedge (p \vee q)) \rightarrow (q \vee r)$

Example:
Let *p* be "I will study discrete math."
Let *r* be "I will study English literature."
Let q be "I will study databases."

"I will not study discrete math or I will study English literature."
"I will study discrete math or I will study databases."

"Therefore, I will study databases or I will study English literature."

Using the Rules of Inference to Build Valid Arguments

- A *valid argument* is a sequence of statements. Each statement is either a premise or follows from previous statements by rules of inference. The last statement is called conclusion.
- A valid argument takes the following form:

$$
\begin{array}{l}
S_1 \\
S_2 \\
\vdots \\
S_n \\
\hline
\therefore\ C
\end{array}
$$

Valid Arguments

Example 1: From the single proposition

$$p \wedge (p \rightarrow q)$$

Show that q is a conclusion.

Solution:

Step	**Reason**
1. $p \wedge (p \rightarrow q)$	Premise
2. p	Simplification using (1)
3. $p \rightarrow q$	Simplification using (1)
4. q	Modus Ponens using (2) and (3)

Valid Arguments

Example 2:

- With these hypotheses:

  "It is not sunny this afternoon and it is colder than yesterday."

  "We will go swimming only if it is sunny."

  "If we do not go swimming, then we will take a canoe trip."

  "If we take a canoe trip, then we will be home by sunset."

- Using the inference rules, construct a valid argument for the conclusion:

  "We will be home by sunset."

Solution:

1. Choose propositional variables:

   p : "It is sunny this afternoon." r : "We will go swimming." t : "We will be home by sunset."

   q : "It is colder than yesterday." s : "We will take a canoe trip."

2. Translation into propositional logic:

$$\text{Hypotheses: } \neg p \wedge q,\ r \rightarrow p,\ \neg r \rightarrow s,\ s \rightarrow t$$
$$\text{Conclusion: } t$$

Continued on next slide →

Valid Arguments

3. Construct the Valid Argument

Step	**Reason**
1. $\neg p \wedge q$	Premise
2. $\neg p$	Simplification using (1)
3. $r \rightarrow p$	Premise
4. $\neg r$	Modus tollens using (2) and (3)
5. $\neg r \rightarrow s$	Premise
6. s	Modus ponens using (4) and (5)
7. $s \rightarrow t$	Premise
8. t	Modus ponens using (6) and (7)

Handling Quantified Statements

- Valid arguments for quantified statements are a sequence of statements. Each statement is either a premise or follows from previous statements by rules of inference which include:
  - Rules of Inference for Propositional Logic
  - Rules of Inference for Quantified Statements
- The rules of inference for quantified statements are introduced in the next several slides.

Universal Instantiation (UI)

$$\frac{\forall x P(x)}{\therefore P(c)}$$

Example:

Our domain consists of all dogs and Fido is a dog.

"All dogs are cuddly."

"Therefore, Fido is cuddly."

Universal Generalization (UG)

$$\frac{P(c) \text{ for an arbitrary } c}{\therefore \forall x P(x)}$$

Used often implicitly in Mathematical Proofs.

Existential Instantiation (EI)

$$\frac{\exists x P(x)}{\therefore P(c) \text{ for some element } c}$$

Example:

"There is someone who got an A in the course."
"Let's call her *a* and say that *a* got an A"

Existential Generalization (EG)

$$\frac{P(c) \text{ for some element } c}{\therefore \exists x P(x)}$$

Example:

"Michelle got an A in the class."
"Therefore, someone got an A in the class."

Using Rules of Inference

Example 1: Using the rules of inference, construct a valid argument to show that

"John Smith has two legs"

is a consequence of the premises:

"Every man has two legs." "John Smith is a man."

Solution: Let $M(x)$ denote "x is a man" and $L(x)$ " x has two legs" and let John Smith be a member of the domain.

Valid Argument:

Step	**Reason**
1. $\forall x(M(x) \rightarrow L(x))$	Premise
2. $M(J) \rightarrow L(J)$	UI from (1)
3. $M(J)$	Premise
4. $L(J)$	Modus Ponens using (2) and (3)

Using Rules of Inference

Example 2: Use the rules of inference to construct a valid argument showing that the conclusion

"Someone who passed the first exam has not read the book."

follows from the premises

"A student in this class has not read the book."

"Everyone in this class passed the first exam."

Solution: Let $C(x)$ denote "x is in this class," $B(x)$ denote "x has read the book," and $P(x)$ denote "x passed the first exam."

First we translate the premises and conclusion into symbolic form.

$$\begin{array}{l} \exists x(C(x) \wedge \neg B(x)) \\ \forall x(C(x) \rightarrow P(x)) \\ \hline \therefore\ \exists x(P(x) \wedge \neg B(x)) \end{array}$$

Continued on next slide →

Using Rules of Inference

Valid Argument:

Step	**Reason**
1. $\exists x(C(x) \wedge \neg B(x))$	Premise
2. $C(a) \wedge \neg B(a)$	EI from (1)
3. $C(a)$	Simplification from (2)
4. $\forall x(C(x) \rightarrow P(x))$	Premise
5. $C(a) \rightarrow P(a)$	UI from (4)
6. $P(a)$	MP from (3) and (5)
7. $\neg B(a)$	Simplification from (2)
8. $P(a) \wedge \neg B(a)$	Conj from (6) and (7)
9. $\exists x(P(x) \wedge \neg B(x))$	EG from (8)

Returning to the Socrates Example

$$\forall x(Man(x) \rightarrow Mortal(x))$$

$$\frac{Man(Socrates)}{\therefore \quad Mortal(Socrates)}$$

Solution for Socrates Example

Valid Argument

Step	**Reason**
1. $\forall x(Man(x) \rightarrow Mortal(x))$	Premise
2. $Man(Socrates) \rightarrow Mortal(Socrates)$	UI from (1)
3. $Man(Socrates)$	Premise
4. $Mortal(Socrates)$	MP from (2) and (3)

Universal Modus Ponens

Universal Modus Ponens combines universal instantiation and modus ponens into one rule.

$$\frac{\begin{array}{l}\forall x(P(x) \rightarrow Q(x)) \\ P(a), \text{where } a \text{ is a particular} \\ \quad \text{element in the domain}\end{array}}{\therefore Q(a)}$$

This rule could be used in the Socrates example.

Introduction to Proofs

Section 1.7

Section Summary

- Mathematical Proofs
- Forms of Theorems
- Direct Proofs
- Indirect Proofs
  - Proof of the Contrapositive
  - Proof by Contradiction

Proofs of Mathematical Statements

- A *proof* is a valid argument that establishes the truth of a statement.
- In math, CS, and other disciplines, informal proofs which are generally shorter, are generally used.
  - More than one rule of inference are often used in a step.
  - Steps may be skipped.
  - The rules of inference used are not explicitly stated.
  - Easier for to understand and to explain to people.
  - But it is also easier to introduce errors.
- Proofs have many practical applications:
  - verification that computer programs are correct
  - establishing that operating systems are secure
  - enabling programs to make inferences in artificial intelligence
  - showing that system specifications are consistent

Definitions

- A *theorem* is a statement that can be shown to be true using:
  - definitions
  - other theorems
  - *axioms* (statements which are given as true)
  - rules of inference
- A *lemma* is a 'helping theorem' or a result which is needed to prove a theorem.
- A *corollary* is a result which follows directly from a theorem.
- Less important theorems are sometimes called *propositions*.
- A *conjecture* is a statement that is being proposed to be true. Once a proof of a conjecture is found, it becomes a theorem. It may turn out to be false.

Forms of Theorems

- Many theorems assert that a property holds for all elements in a domain, such as the integers, the real numbers, or some of the discrete structures that we will study in this class.
- Often the universal quantifier (needed for a precise statement of a theorem) is omitted by standard mathematical convention.

For example, the statement:

"If $x > y$, where x and y are positive real numbers, then $x^2 > y^2$ "

really means

"For all positive real numbers x and y, if $x > y$, then $x^2 > y^2$."

Proving Theorems

- Many theorems have the form:

$$\forall x(P(x) \rightarrow Q(x))$$

- To prove them, we show that where *c* is an arbitrary element of the domain, $P(c) \rightarrow Q(c)$
- By universal generalization the truth of the original formula follows.
- So, we must prove something of the form: $p \rightarrow q$

Proving Conditional Statements: $p \rightarrow q$

- *Trivial Proof*: If we know q is true, then
  $p \rightarrow q$ is true as well.

"If it is raining then 1=1."

- *Vacuous Proof*: If we know p is false then
  $p \rightarrow q$ is true as well.

"If I am both rich and poor then $2 + 2 = 5$."

[Even though these examples seem silly, both trivial and vacuous proofs are often used in mathematical induction, as we will see in Chapter 5)]

Even and Odd Integers

Definition: The integer n is even if there exists an integer k such that $n = 2k$, and n is odd if there exists an integer k, such that $n = 2k + 1$. Note that every integer is either even or odd and no integer is both even and odd.

We will need this basic fact about the integers in some of the example proofs to follow. We will learn more about the integers in Chapter 4.

Proving Conditional Statements: $p \rightarrow q$

- *Direct Proof*: Assume that p is true. Use rules of inference, axioms, and logical equivalences to show that q must also be true.

Example: Give a direct proof of the theorem "If n is an odd integer, then n^2 is odd."

Solution: Assume that n is odd. Then $n = 2k + 1$ for an integer k. Squaring both sides of the equation, we get:

$$n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1 = 2r + 1,$$

where $r = 2k^2 + 2k$, an integer.

We have proved that if n is an odd integer, then n^2 is an odd integer. ◀

(◀ marks the end of the proof. Sometimes **QED** is used instead.)

Proving Conditional Statements: $p \rightarrow q$

Definition: The real number r is *rational* if there exist integers p and q where $q \neq 0$ such that $r = p/q$

Example: Prove that the sum of two rational numbers is rational.

Solution: Assume r and s are two rational numbers. Then there must be integers p, q and also t, u such that

$$r = p/q, \quad s = t/u, \quad u \neq 0, \; q \neq 0$$

$$r + s = \frac{p}{q} + \frac{t}{u} = \frac{pu+qt}{qu} = \frac{v}{w}$$

where $v = pu + qt$
$w = qu \neq 0$

Thus the sum is rational. ◀

Proving Conditional Statements: $p \rightarrow q$

- *Proof by Contraposition*: Assume $\neg q$ and show $\neg p$ is true also. This is sometimes called an *indirect proof* method. If we give a direct proof of $\neg q \rightarrow \neg p$ then we have a proof of $p \rightarrow$ q.

  Why does this work?

Example: Prove that if n is an integer and $3n + 2$ is odd, then n is odd.

Solution: Assume n is even. So, $n = 2k$ for some integer k. Thus

$3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1) = 2j$ for $j = 3k + 1$

Therefore $3n + 2$ is even. Since we have shown $\neg q \rightarrow \neg p$, $p \rightarrow q$ must hold as well. If n is an integer and $3n + 2$ is odd (not even), then n is odd (not even). ◀

Proving Conditional Statements: $p \rightarrow q$

Example: Prove that for an integer n, if n^2 is odd, then n is odd.

Solution: Use proof by contraposition. Assume n is even (i.e., not odd). Therefore, there exists an integer k such that $n = 2k$. Hence,

$$n^2 = 4k^2 = 2\,(2k^2)$$

and n^2 is even(i.e., not odd).

We have shown that if n is an even integer, then n^2 is even. Therefore by contraposition, for an integer n, if n^2 is odd, then n is odd. ◀

Proving Conditional Statements: $p \rightarrow q$

- *Proof by Contradiction*: (AKA *reductio ad absurdum*).

  To prove p, assume $\neg p$ and derive a contradiction such as $p \wedge \neg p$. (an indirect form of proof). Since we have shown that $\neg p \rightarrow \mathbf{F}$ is true , it follows that the contrapositive $\mathbf{T} \rightarrow p$ also holds.

- We can prove that p is true if we can show *that*…,$p \rightarrow (r \wedge \neg r)$ is true for some proposition r

  Example: Prove that if you pick 22 days from the calendar, at least 4 must fall on the same day of the week.

  Solution: Assume that no more than 3 of the 22 days fall on the same day of the week. Because there are 7 days of the week, we could only have picked 21 days. This contradicts the assumption that we have picked 22 days. ◄

Proof by Contradiction

- A preview of Chapter 4.

Example: Use a proof by contradiction to give a proof that $\sqrt{2}$ is irrational.

Solution: Suppose $\sqrt{2}$ is rational. Then there exists integers a and b with $\sqrt{2} = a/b$, where $b \neq 0$ and a and b have no common factors (see Chapter 4). Then

$$2 = \frac{a^2}{b^2} \qquad 2b^2 = a^2$$

Therefore a^2 must be even. If a^2 is even then a must be even (an exercise). Since a is even, $a = 2c$ for some integer c. Thus,

$$2b^2 = 4c^2 \qquad b^2 = 2c^2$$

Therefore b^2 is even. Again then b must be even as well.

But then 2 must divide both a and b. This contradicts our assumption that a and b have no common factors. We have proved by contradiction that our initial assumption must be false and therefore $\sqrt{2}$ is irrational .

Proof by Contradiction

- A preview of Chapter 4.

  Example: Prove that there is no largest prime number.

  Solution: Assume that there is a largest prime number. Call it p_n. Hence, we can list all the primes 2,3,.., p_n. Form

$$r = p_1 \times p_2 \times \ldots \times p_n + 1$$

  None of the prime numbers on the list divides *r*. Therefore, by a theorem in Chapter 4, either *r* is prime or there is a smaller prime that divides *r*. This contradicts the assumption that there is a largest prime. Therefore, there is no largest prime. ◄

Theorems that are Biconditional Statements

- To prove a theorem that is a biconditional statement, that is, a statement of the form $p \leftrightarrow q$, we show that $p \rightarrow q$ and $q \rightarrow p$ are both true.

Example: Prove the theorem: "If n is an integer, then n is odd if and only if n^2 is odd."

Solution: We have already shown (previous slides) that both $p \rightarrow q$ and $q \rightarrow p$. Therefore we can conclude $p \leftrightarrow q$.

Sometimes *iff* is used as an abbreviation for "if an only if," as in

"If n is an integer, then n is odd iif n^2 is odd."

What is wrong with this?

"Proof" that *1 = 2*

Step	**Reason**
1. $a = b$	Premise
2. $a^2 = a \times b$	Multiply both sides of (1) by a
3. $a^2 - b^2 = a \times b - b^2$	Subtract b^2 from both sides of (2)
4. $(a - b)(a + b) = b(a - b)$	Algebra on (3)
5. $a + b = b$	Divide both sides by $a - b$
6. $2b = b$	Replace a by b in (5) because $a = b$
7. $2 = 1$	Divide both sides of (6) by b

Solution: Step 5. a - b = 0 by the premise and division by 0 is undefined.

Looking Ahead

- If direct methods of proof do not work:
  - We may need a clever use of a proof by contraposition.
  - Or a proof by contradiction.
  - In the next section, we will see strategies that can be used when straightforward approaches do not work.
  - In Chapter 5, we will see mathematical induction and related techniques.
  - In Chapter 6, we will see combinatorial proofs

Induction and recursion

Chapter 5

With Question/Answer Animations

Chapter Summary

- Mathematical Induction
- Strong Induction
- Well-Ordering
- Recursive Definitions
- Structural Induction
- Recursive Algorithms
- Program Correctness (*not yet included in overheads*)

Mathematical Induction

Section 5.1

Section Summary

- Mathematical Induction
- Examples of Proof by Mathematical Induction
- Mistaken Proofs by Mathematical Induction
- Guidelines for Proofs by Mathematical Induction

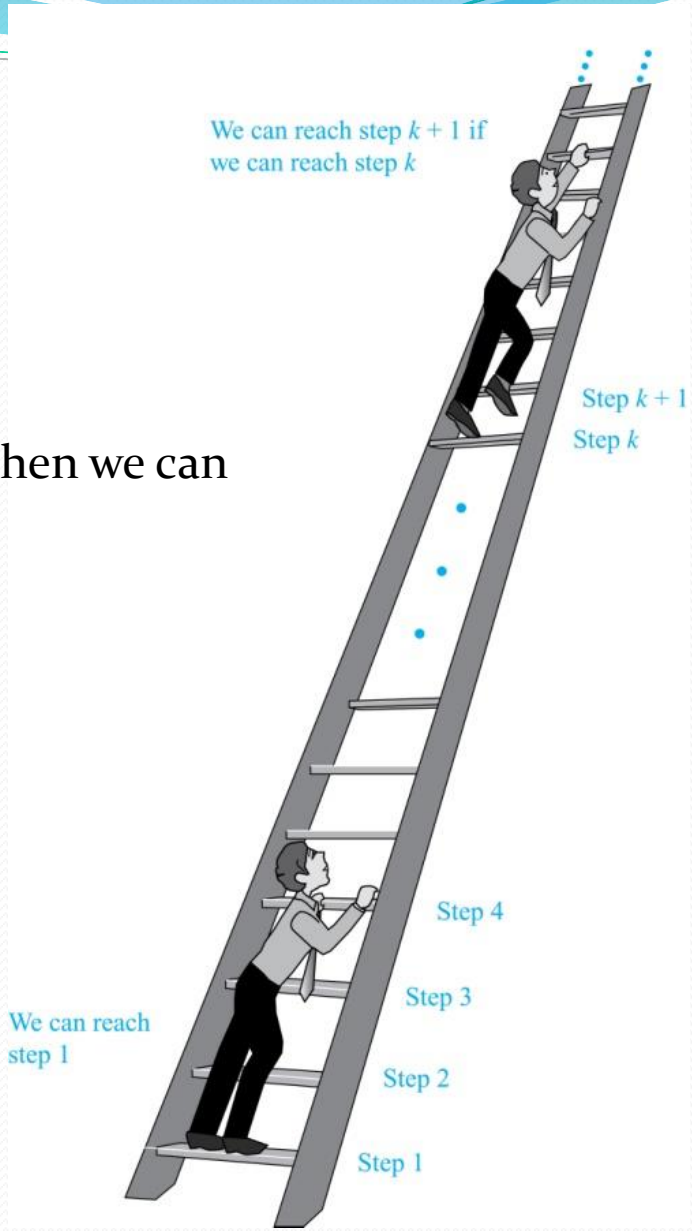
Climbing an Infinite Ladder

Suppose we have an infinite ladder:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

From (1), we can reach the first rung. Then by applying (2), we can reach the second rung. Applying (2) again, the third rung. And so on. We can apply (2) any number of times to reach any particular rung, no matter how high up.

This example motivates proof by mathematical induction.



Principle of Mathematical Induction

Principle of Mathematical Induction: To prove that $P(n)$ is true for all positive integers n, we complete these steps:

- *Basis Step*: Show that $P(1)$ is true.
- *Inductive Step*: Show that $P(k) \rightarrow P(k+1)$ is true for all positive integers k.

To complete the inductive step, assuming the *inductive hypothesis* that $P(k)$ holds for an arbitrary integer k, show that must $P(k+1)$ be true.

Climbing an Infinite Ladder Example:

- BASIS STEP: By (1), we can reach rung 1.
- INDUCTIVE STEP: Assume the inductive hypothesis that we can reach rung k. Then by (2), we can reach rung $k+1$.

Hence, $P(k) \rightarrow P(k+1)$ is true for all positive integers k. We can reach every rung on the ladder.

◀

Important Points About Using Mathematical Induction

- Mathematical induction can be expressed as the rule of inference

$$(P(1) \land \forall k\,(P(k) \rightarrow P(k+1))) \rightarrow \forall n\,P(n),$$

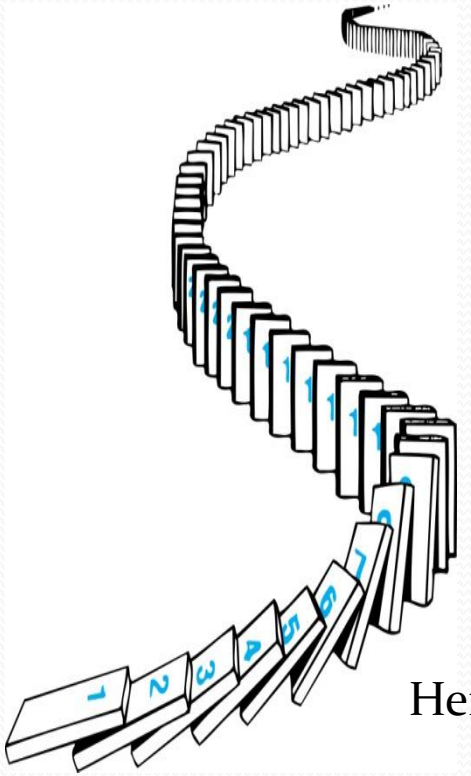
where the domain is the set of positive integers.

- In a proof by mathematical induction, we don't assume that $P(k)$ is true for all positive integers! We show that if we assume that $P(k)$ is true, then $P(k+1)$ must also be true.
- Proofs by mathematical induction do not always start at the integer 1. In such a case, the basis step begins at a starting point b where b is an integer. We will see examples of this soon.

Remembering How Mathematical Induction Works

Consider an infinite sequence of dominoes, labeled 1,2,3, …, where each domino is standing.

Let $P(n)$ be the proposition that the nth domino is knocked over.



We know that the first domino is knocked down, i.e., $P(1)$ is true .

We also know that if whenever the kth domino is knocked over, it knocks over the $(k + 1)$st domino, i.e, $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.

Hence, all dominos are knocked over.

$P(n)$ is true for all positive integers n.

Well-Ordering Property

- *Well-ordering property*: Every nonempty set of nonnegative integers has a least element.
- The well-ordering property is one of the axioms of the positive integers listed in Appendix 1.
- The well-ordering property can be used directly in proofs, as the next example illustrates.
- The well-ordering property can be generalized.
  - **Definition:** A set is *well ordered if every subset has a least element*.
    - **N** is well ordered under ≤.
    - The set of finite strings over an alphabet using lexicographic ordering is well ordered.
  - We will see a generalization of induction to sets other than the integers in the next section.

Conjecturing and Proving Correct a Summation Formula

Example: Conjecture and prove correct a formula for the sum of the first n positive odd integers. Then prove your conjecture.

Solution: We have: $1 = 1, 1 + 3 = 4, 1 + 3 + 5 = 9, 1 + 3 + 5 + 7 = 16, 1 + 3 + 5 + 7 + 9 = 25.$

- We can conjecture that the sum of the first n positive odd integers is n^2,

$$1 + 3 + 5 + \cdots + (2n - 1) + (2n + 1) = n^2.$$

- We prove the conjecture is proved correct with mathematical induction.
- BASIS STEP: $P(1)$ is true since $1^2 = 1$.
- INDUCTIVE STEP: $P(k) \rightarrow P(k + 1)$ for every positive integer k.

  Assume the inductive hypothesis holds and then show that $P(k + 1)$ holds has well.

  Inductive Hypothesis: $1 + 3 + 5 + \cdots + (2k - 1) = k^2$

- So, assuming $P(k)$, it follows that:

$$\begin{aligned} 1 + 3 + 5 + \cdots + (2k - 1) + (2k + 1) &= [1 + 3 + 5 + \cdots + (2k - 1)] + (2k + 1) \\ &= k^2 + (2k + 1) \quad (\textit{by the inductive hypothesis}) \\ &= k^2 + 2k + 1 \\ &= (k + 1)^2 \end{aligned}$$

- Hence, we have shown that $P(k + 1)$ follows from $P(k)$. Therefore the sum of the first n positive odd integers is n^2.

Proving Inequalities

Example: Use mathematical induction to prove that $n < 2^n$ for all positive integers n.

Solution: Let $P(n)$ be the proposition that $n < 2^n$.

- BASIS STEP: $P(1)$ is true since $1 < 2^1 = 2$.
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $k < 2^k$, for an arbitrary positive integer k.
- Must show that $P(k + 1)$ holds. Since by the inductive hypothesis, $k < 2^k$, it follows that:

$$k + 1 < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$$

Therefore $n < 2^n$ holds for all positive integers n. ◀

Proving Divisibility Results

Example: Use mathematical induction to prove that $n^3 - n$ is divisible by 3, for every positive integer n.

Solution: Let $P(n)$ be the proposition that $n^3 - n$ is divisible by 3.

- BASIS STEP: $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $k^3 - k$ is divisible by 3, for an arbitrary positive integer k. To show that $P(k+1)$ follows:

$$(k+1)^3 - (k+1) = (k^3 + 3k^2 + 3k + 1) - (k+1)$$
$$= (k^3 - k) + 3(k^2 + k)$$

By the inductive hypothesis, the first term $(k^3 - k)$ is divisible by 3 and the second term is divisible by 3 since it is an integer multiplied by 3. So by part (i) of Theorem 1 in Section 4.1 , $(k+1)^3 - (k+1)$ is divisible by 3.

Therefore, $n^3 - n$ is divisible by 3, for every integer positive integer n. ◀

Example

- Question: Use mathematical induction to show that $1+2+2^2+.....+2^n = 2^{n+1}-1$ for all nonnegative integers n.
- *Solution:* Let P(n) be the proposition that this formula is correct for integer n.
- *BASIS STEP:* P(0) is true since $2^0 = 1 = 2^1-1$.
- *INDUCTIVE STEP:* Assume that P(n) is true. To carry out the inductive step using this assumption, it must be shown that P(n+1) is true, namely,
- $1+2+2^2+.....+2^n+2^{n+1} = (1+2+2^2+...+2^n)+2^{n+1}$
- $= (2^{n+1}-1)+2^{n+1} = 2.2^{n+1}-1 = 2^{n+2}-1$
- This finishes the inductive step, which completes the proof.
- To use mathematical induction to show that P(n) is true for n = k, k+1, k+2, …. , where k is an integer other than 1, we show that P(k) is true [the basis step] and then show that the implication $P(n) \rightarrow P(n+1)$ is true for n = k, k+1, k+2,… [the inductive step]. Note that k can be negative, zero or positive.

Guidelines: Mathematical Induction Proofs

Template for Proofs by Mathematical Induction

1. Express the statement that is to be proved in the form "for all $n \geq b$, $P(n)$" for a fixed integer b.
2. Write out the words "Basis Step." Then show that $P(b)$ is true, taking care that the correct value of b is used. This completes the first part of the proof.
3. Write out the words "Inductive Step."
4. State, and clearly identify, the inductive hypothesis, in the form "assume that $P(k)$ is true for an arbitrary fixed integer $k \geq b$."
5. State what needs to be proved under the assumption that the inductive hypothesis is true. That is, write out what $P(k + 1)$ says.
6. Prove the statement $P(k + 1)$ making use the assumption $P(k)$. Be sure that your proof is valid for all integers k with $k \geq b$, taking care that the proof works for small values of k, including $k = b$.
7. Clearly identify the conclusion of the inductive step, such as by saying "this completes the inductive step."
8. After completing the basis step and the inductive step, state the conclusion, namely that by mathematical induction, $P(n)$ is true for all integers n with $n \geq b$.

Strong Induction and Well-Ordering

Section 5.2

Section Summary

- Strong Induction
- Example Proofs using Strong Induction
- Using Strong Induction in Computational Geometry (*not yet included in overheads*)
- Well-Ordering Property

Strong Induction

- *Strong Induction*: To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, complete two steps:
  - *Basis Step*: Verify that the proposition $P(1)$ is true.
  - *Inductive Step*: Show the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ holds for all positive integers k.

Strong Induction is sometimes called the *second principle of mathematical induction* or *complete induction*.

Strong Induction and the Infinite Ladder

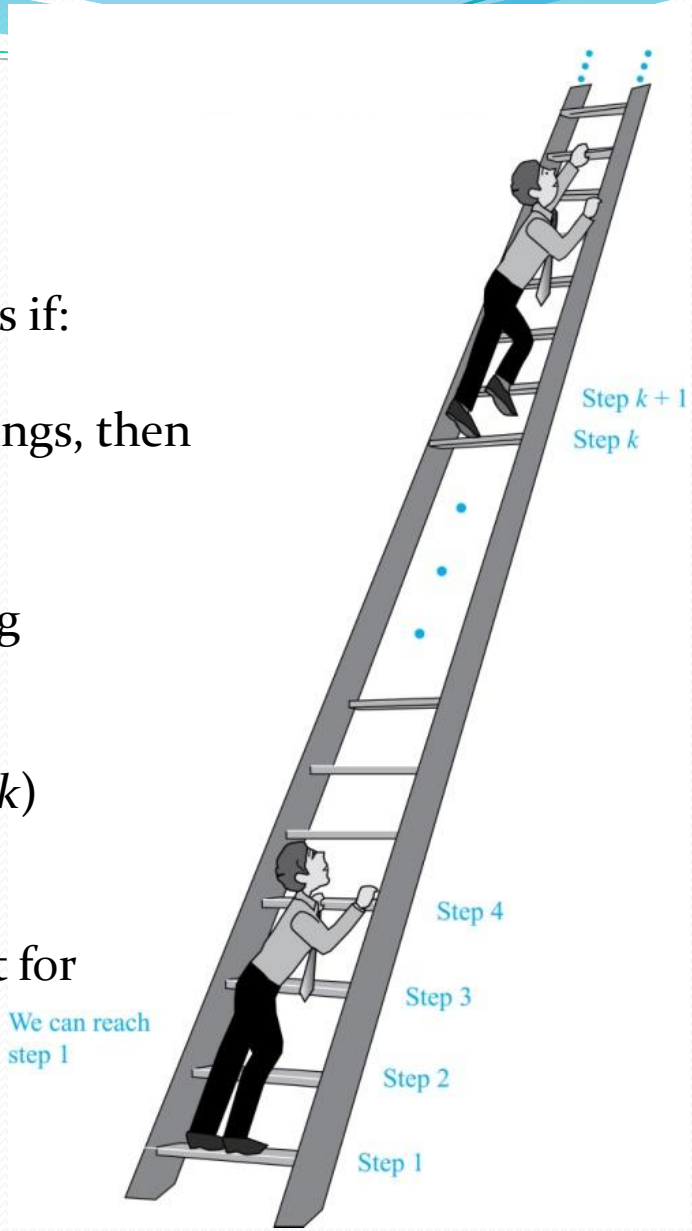
Strong induction tells us that we can reach all rungs if:

1. We can reach the first rung of the ladder.
2. For every integer k, if we can reach the first k rungs, then we can reach the $(k + 1)$st rung.

To conclude that we can reach every rung by strong induction:

- BASIS STEP: $P(1)$ holds
- INDUCTIVE STEP: Assume $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ holds for an arbitrary integer k, and show that $P(k + 1)$ must also hold.

We will have then shown by strong induction that for every positive integer n, $P(n)$ holds, i.e., we can reach the nth rung of the ladder.



Completion of the proof of the Fundamental Theorem of Arithmetic

Example: Show that if n is an integer greater than 1, then n can be written as the product of primes.

Solution: Let $P(n)$ be the proposition that n can be written as a product of primes.

- BASIS STEP: $P(2)$ is true since 2 itself is prime.
- INDUCTIVE STEP: The inductive hypothesis is $P(j)$ is true for all integers j with $2 \leq j \leq k$. To show that $P(k + 1)$ must be true under this assumption, two cases need to be considered:
  - If $k + 1$ is prime, then $P(k + 1)$ is true.
  - Otherwise, $k + 1$ is composite and can be written as the product of two positive integers a and b with $2 \leq a \leq b < k + 1$. By the inductive hypothesis a and b can be written as the product of primes and therefore $k + 1$ can also be written as the product of those primes.

Hence, it has been shown that every integer greater than 1 can be written as the product of primes.

◀

(*uniqueness proved in Section* 4.3)

Proof using Strong Induction

Example: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

- BASIS STEP: $P(12)$, $P(13)$, $P(14)$, and $P(15)$ hold.
  - $P(12)$ uses three 4-cent stamps.
  - $P(13)$ uses two 4-cent stamps and one 5-cent stamp.
  - $P(14)$ uses one 4-cent stamp and two 5-cent stamps.
  - $P(15)$ uses three 5-cent stamps.
- INDUCTIVE STEP: The inductive hypothesis states that $P(j)$ holds for $12 \leq j \leq k$, where $k \geq 15$. Assuming the inductive hypothesis, it can be shown that $P(k + 1)$ holds.
- Using the inductive hypothesis, $P(k - 3)$ holds since $k - 3 \geq 12$. To form postage of $k + 1$ cents, add a 4-cent stamp to the postage for $k - 3$ cents.

Hence, $P(n)$ holds for all $n \geq 12$. ◀

Proof of Same Example using Mathematical Induction

Example: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

- BASIS STEP: Postage of 12 cents can be formed using three 4-cent stamps.
- INDUCTIVE STEP: The inductive hypothesis $P(k)$ for any positive integer k is that postage of k cents can be formed using 4-cent and 5-cent stamps. To show $P(k + 1)$ where $k \geq 12$, we consider two cases:
  - If at least one 4-cent stamp has been used, then a 4-cent stamp can be replaced with a 5-cent stamp to yield a total of k + 1 cents.
  - Otherwise, no 4-cent stamp have been used and at least three 5-cent stamps were used. Three 5-cent stamps can be replaced by four 4-cent stamps to yield a total of k + 1 cents.

Hence, $P(n)$ holds for all $n \geq 12$. ◀

Well-Ordering Property

Example: Use the well-ordering property to prove the division algorithm, which states that if a is an integer and d is a positive integer, then there are unique integers q and r with $0 \leq r < d$, such that $a = dq + r$.

Solution: Let S be the set of nonnegative integers of the form $a - dq$, where q is an integer. The set is nonempty since $-dq$ can be made as large as needed.

- By the well-ordering property, S has a least element $r = a - dq_0$. The integer r is nonnegative. It also must be the case that $r < d$. If it were not, then there would be a smaller nonnegative element in S, namely, $a - d(q_0 + 1) = a - dq_0 - d = r - d > 0$.
- Therefore, there are integers q and r with $0 \leq r < d$.

(*uniqueness of q and r is Exercise* 37)

Recursive Definitions and Structural Induction

Section 5.3

Section Summary

- Recursively Defined Functions
- Recursively Defined Sets and Structures
- Structural Induction
- Generalized Induction

Recursively Defined Functions

Definition: A *recursive* or *inductive definition* of a function consists of two steps.

- BASIS STEP: Specify the value of the function at zero.
- RECURSIVE STEP: Give a rule for finding its value at an integer from its values at smaller integers.

- A function $f(n)$ is the same as a sequence $a_0, a_1, \ldots,$ where a_i, where $f(i) = a_i$. This was done using recurrence relations in Section 2.4.

Recursively Defined Functions

Example: Suppose f is defined by:

$f(0) = 3,$

$f(n + 1) = 2f(n) + 3$

Find $f(1), f(2), f(3), f(4)$

Solution:

- $f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$
- $f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$
- $f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$
- $f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$

Example: Give a recursive definition of the factorial function $n!$:

Solution:

$f(0) = 1$

$f(n + 1) = (n + 1) \cdot f(n)$

Recursively Defined Functions

Example: Give a recursive definition of:

$$\sum_{k=0}^{n} a_k.$$

Solution: The first part of the definition is

$$\sum_{k=0}^{0} a_k = a_0.$$

The second part is

$$\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^{n} a_k\right) + a_{n+1}.$$

Fibonacci
(1170- 1250)

Fibonacci Numbers

Example : The Fibonacci numbers are defined as follows:

$f_0 = 0$

$f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$

Find f_2, f_3, f_4, f_5.

- $f_2 = f_1 + f_0 = 1 + 0 = 1$
- $f_3 = f_2 + f_1 = 1 + 1 = 2$
- $f_4 = f_3 + f_2 = 2 + 1 = 3$
- $f_5 = f_4 + f_3 = 3 + 2 = 5$

In Chapter 8, we will use the Fibonacci numbers to model population growth of rabbits. This was an application described by Fibonacci himself.

Next, we use strong induction to prove a result about the Fibonacci numbers.

Fibonacci Numbers

Example 4: Show that whenever $n \geq 3, f_n > \alpha^{n-2}$, where $\alpha = (1+\sqrt{5})/2$.

Solution: Let $P(n)$ be the statement $f_n > \alpha^{n-2}$. Use strong induction to show that $P(n)$ is true whenever $n \geq 3$.

- BASIS STEP: $P(3)$ holds since $\alpha < 2 = f_3$

  $P(4)$ holds since $\alpha^2 = (3+\sqrt{5})/2 < 3 = f_4$.
- INDUCTIVE STEP: Assume that $P(j)$ holds, i.e., $f_j > \alpha^{j-2}$ for all integers j with $3 \leq j \leq k$, where $k \geq 4$. Show that $P(k+1)$ holds, i.e., $f_{k+1} > \alpha^{k-1}$.
  - Since $\alpha^2 = \alpha + 1$ (because α is a solution of $x^2 - x - 1 = 0$),

    $$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1)\cdot\alpha^{k-3} = \alpha\cdot\alpha^{k-3} + 1\cdot\alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}$$
  - By the inductive hypothesis, because $k \geq 4$ we have

    $$f_{k-1} > \alpha^{k-3}, \qquad f_k > \alpha^{k-2}.$$
  - Therefore, it follows that

    $$f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}.$$
  - Hence, $P(k+1)$ is true.

Why does this equality hold?

Gabriel Lamé
(1795-1870)

Lamé's Theorem

Lamé's Theorem: Let a and b be positive integers with $a \geq b$. Then the number of divisions used by the Euclidian algorithm to find gcd(a,b) is less than or equal to five times the number of decimal digits in b.

Proof: When we use the Euclidian algorithm to find gcd(a,b) with $a \geq b$,

- n divisions are used to obtain (with $a = r_0, b = r_1$):

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1,\\
r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2,\\
&\vdots\\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1},\\
r_{n-1} &= r_n q_n.
\end{aligned}
$$

- Since each quotient $q_1, q_2, \ldots, q_{n-1}$ is at least 1 and $q_n \geq 2$:

$$
\begin{aligned}
r_n &\geq 1 = f_2,\\
r_{n-1} &\geq 2\, r_n \geq 2 f_2 = f_3,\\
r_{n-2} &\geq r_{n-1} + r_n \geq f_3 + f_2 = f_4,\\
&\vdots\\
r_2 &\geq r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n,\\
b = r_1 &\geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}.
\end{aligned}
$$

continued →

Lamé's Theorem

- It follows that if n divisions are used by the Euclidian algorithm to find gcd(a,b) with $a \geq b$, then $b \geq f_{n+1}$. By Example 4, $f_{n+1} > \alpha^{n-1}$, for $n > 2$, where $\alpha = (1 + \sqrt{5})/2$. Therefore, $b > \alpha^{n-1}$.
- Because $\log_{10} \alpha \approx 0.208 > 1/5$, $\log_{10} b > (n-1) \log_{10} \alpha > (n-1)/5$. Hence,

$$n-1 < 5 \cdot \log_{10} b.$$

- Suppose that b has k decimal digits. Then $b < 10^k$ and $\log_{10} b < k$. It follows that $n - 1 < 5k$ and since k is an integer, $n \leq 5k$.

- As a consequence of Lamé's Theorem, $O(\log b)$ divisions are used by the Euclidian algorithm to find gcd(a,b) whenever $a > b$.
  - By Lamé's Theorem, the number of divisions needed to find gcd(a,b) with $a > b$ is less than or equal to 5 ($\log_{10} b + 1$) since the number of decimal digits in b (which equals $[\log_{10} b] + 1$) is less than or equal to $\log_{10} b + 1$.

Lamé's Theorem was the first result in computational complexity

Recursively Defined Sets and Structures

Recursive definitions of sets have two parts:

- The *basis step* specifies an initial collection of elements.
- The *recursive step* gives the rules for forming new elements in the set from those already known to be in the set.

- Sometimes the recursive definition has an *exclusion rule*, which specifies that the set contains nothing other than those elements specified in the basis step and generated by applications of the rules in the recursive step.
- We will always assume that the exclusion rule holds, even if it is not explicitly mentioned.
- We will later develop a form of induction, called *structural induction*, to prove results about recursively defined sets.

Recursively Defined Sets and Structures

Example : Subset of Integers S:

BASIS STEP: $3 \in S$.

RECURSIVE STEP: If $x \in S$ and $y \in S$, then $x + y$ is in S.

- Initially 3 is in S, then $3 + 3 = 6$, then $3 + 6 = 9$, etc.

Example: The natural numbers **N**.

BASIS STEP: $0 \in \mathbf{N}$.

RECURSIVE STEP: If n is in **N**, then $n + 1$ is in **N**.

- Initially 0 is in S, then $0 + 1 = 1$, then $1 + 1 = 2$, etc.

Strings

Definition: The set Σ^* of *strings* over the alphabet Σ:

BASIS STEP: $\lambda \in \Sigma^*$ (λ is the empty string)

RECURSIVE STEP: If w is in Σ^* and x is in Σ, then $wx \in \Sigma^*$.

Example: If $\Sigma = \{0,1\}$, the strings in in Σ^* are the set of all bit strings, λ,0,1, 00,01,10, 11, etc.

Example: If $\Sigma = \{a,b\}$, show that aab is in Σ^*.

- Since $\lambda \in \Sigma^*$ and $a \in \Sigma$, $a \in \Sigma^*$.
- Since $a \in \Sigma^*$ and $a \in \Sigma$, $aa \in \Sigma^*$.
- Since $aa \in \Sigma^*$ and $b \in \Sigma$, $aab \in \Sigma^*$.

String Concatenation

Definition: Two strings can be combined via the operation of *concatenation*. Let Σ be a set of symbols and Σ^* be the set of strings formed from the symbols in Σ. We can define the concatenation of two strings, denoted by $\cdot$, recursively as follows.

BASIS STEP: If $w \in \Sigma^*$, then $w \cdot \lambda = w$.

RECURSIVE STEP: If $w_1 \in \Sigma^*$ and $w_2 \in \Sigma^*$ and $x \in \Sigma$, then $w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x$.

- Often $w_1 \cdot w_2$ is written as $w_1 w_2$.
- If $w_1 = abra$ and $w_2 = cadabra$, the concatenation $w_1 w_2 = abracadabra$.

Length of a String

Example: Give a recursive definition of $l(w)$, the length of the string w.

Solution: The length of a string can be recursively defined by:

$l(\lambda) = 0$;

$l(wx) = l(w) + 1$ if $w \in \Sigma^*$ and $x \in \Sigma$.

Balanced Parentheses

Example: Give a recursive definition of the set of balanced parentheses P.

Solution:

BASIS STEP: $() \in P$

RECURSIVE STEP: If $w \in P$, then $()w \in P$, $(w) \in P$ and $w() \in P$.

- Show that $((())())$ is in P.
- Why is $))((()$ not in P?

Well-Formed Formulae in Propositional Logic

Definition: The set of *well-formed formulae* in propositional logic involving **T**, **F**, propositional variables, and operators from the set $\{\neg,\wedge,\vee,\rightarrow,\leftrightarrow\}$.

BASIS STEP: **T**,**F**, and *s*, where *s* is a propositional variable, are well-formed formulae.

RECURSIVE STEP: If *E* and *F* are well formed formulae, then $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$, $(E \leftrightarrow F)$, are well-formed formulae.

Examples: $((p \vee q) \rightarrow (q \wedge \mathbf{F}))$ is a well-formed formula.

$pq \wedge$ is not a well formed formula.

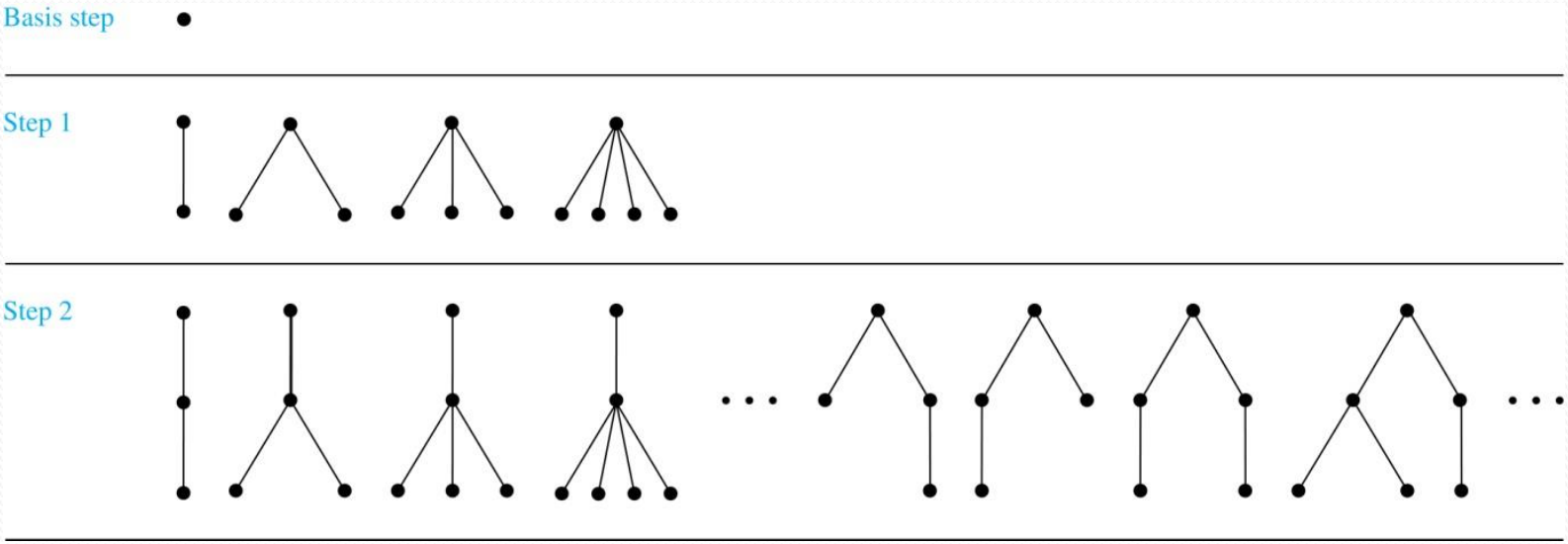
Rooted Trees

Definition: The set of *rooted trees*, where a rooted tree consists of a set of vertices containing a distinguished vertex called the *root*, and edges connecting these vertices, can be defined recursively by these steps:

BASIS STEP: A single vertex r is a rooted tree.

RECURSIVE STEP: Suppose that $T_1, T_2, \ldots, T_n$ are disjoint rooted trees with roots $r_1, r_2, \ldots, r_n$, respectively. Then the graph formed by starting with a root r, which is not in any of the rooted trees $T_1, T_2, \ldots, T_n$, and adding an edge from r to each of the vertices $r_1, r_2, \ldots, r_n$, is also a rooted tree.

Building Up Rooted Trees



- Trees are studied extensively in Chapter 11.
- Next we look at a special type of tree, the full binary tree.

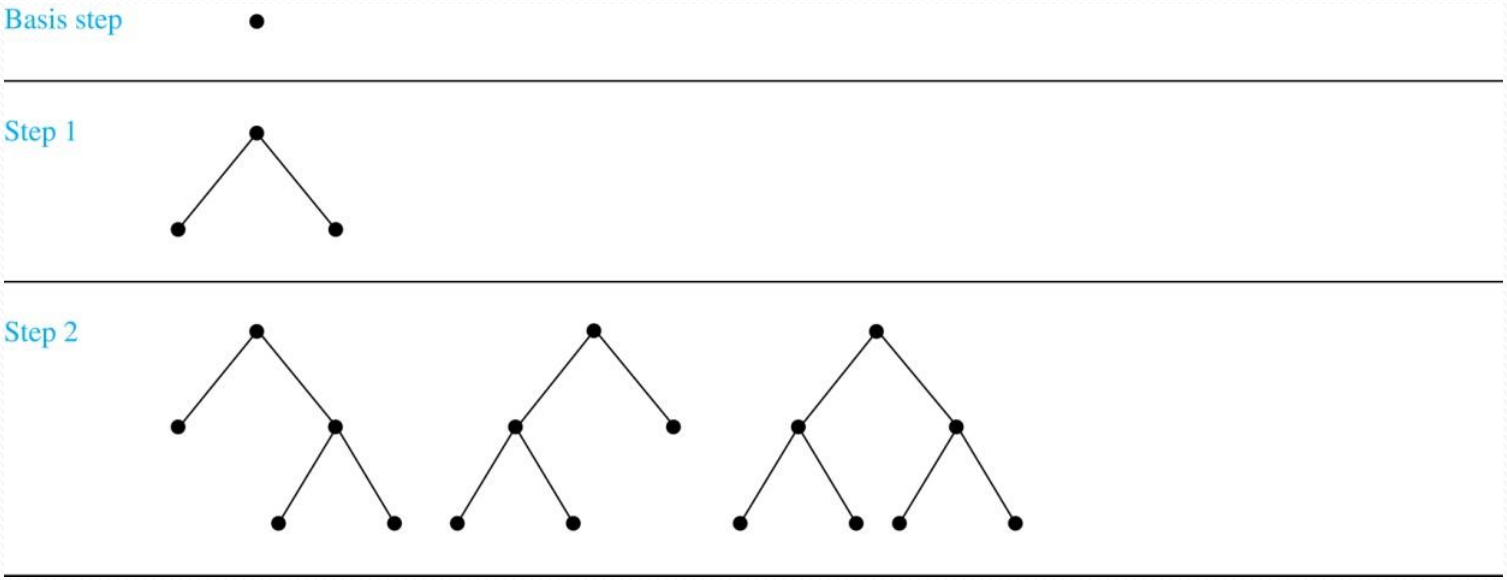
Full Binary Trees

Definition: The set of *full binary trees* can be defined recursively by these steps.

BASIS STEP: There is a full binary tree consisting of only a single vertex r.

RECURSIVE STEP: If T_1 and T_2 are disjoint full binary trees, there is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtree T_2.

Building Up Full Binary Trees



Induction and Recursively Defined Sets

Example: Show that the set S defined by specifying that $3 \in S$ and that if $x \in S$ and $y \in S$, then $x + y$ is in S, is the set of all positive integers that are multiples of 3.

Solution: Let A be the set of all positive integers divisible by 3. To prove that $A = S$, show that A is a subset of S and S is a subset of A.

- $A \subset S$: Let $P(n)$ be the statement that $3n$ belongs to S.

  BASIS STEP: $3 \cdot 1 = 3 \in S$, by the first part of recursive definition.

  INDUCTIVE STEP: Assume $P(k)$ is true. By the second part of the recursive definition, if $3k \in S$, then since $3 \in S$, $3k + 3 = 3(k + 1) \in S$. Hence, $P(k + 1)$ is true.

- $S \subset A$:

  BASIS STEP: $3 \in S$ by the first part of recursive definition, and $3 = 3 \cdot 1$.

  INDUCTIVE STEP: The second part of the recursive definition adds $x + y$ to S, if both x and y are in S. If x and y are both in A, then both x and y are divisible by 3. By part (i) of Theorem 1 of Section 4.1, it follows that $x + y$ is divisible by 3.

- We used mathematical induction to prove a result about a recursively defined set. Next we study a more direct form induction for proving results about recursively defined sets.

Structural Induction

Definition: To prove a property of the elements of a recursively defined set, we use *structural induction*.

BASIS STEP: Show that the result holds for all elements specified in the basis step of the recursive definition.

RECURSIVE STEP: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

- The validity of structural induction can be shown to follow from the principle of mathematical induction.

Full Binary Trees

Definition: The *height* $h(T)$ of a full binary tree T is defined recursively as follows:

- BASIS STEP: The height of a full binary tree T consisting of only a root r is $h(T) = 0$.
- RECURSIVE STEP: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has height $h(T) = 1 + \max(h(T_1), h(T_2))$.

- The number of vertices $n(T)$ of a full binary tree T satisfies the following recursive formula:
  - **BASIS STEP**: The number of vertices of a full binary tree T consisting of only a root r is $n(T) = 1$.
  - **RECURSIVE STEP**: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has the number of vertices
    $$n(T) = 1 + n(T_1) + n(T_2).$$

Structural Induction and Binary Trees

Theorem: If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$.

Proof: Use structural induction.

- BASIS STEP: The result holds for a full binary tree consisting only of a root, $n(T) = 1$ and $h(T) = 0$. Hence, $n(T) = 1 \leq 2^{0+1} - 1 = 1$.
- RECURSIVE STEP: Assume $n(T_1) \leq 2^{h(T_1)+1} - 1$ and also $n(T_2) \leq 2^{h(T_2)+1} - 1$ whenever T_1 and T_2 are full binary trees.

$$
\begin{aligned}
n(T) &= 1 + n(T_1) + n(T_2) && (\textit{by recursive formula of } n(T)) \\
&\leq 1 + (2^{h(T_1)+1} - 1) + (2^{h(T_2)+1} - 1) && (\textit{by inductive hypothesis}) \\
&\leq 2 \cdot \max(2^{h(T_1)+1}, 2^{h(T_2)+1}) - 1 \\
&= 2 \cdot 2^{\max(h(T_1),h(T_2))+1} - 1 && (\max(2^x, 2^y) = 2^{\max(x,y)}) \\
&= 2 \cdot 2^{h(t)} - 1 && (\textit{by recursive definition of } h(T)) \\
&= 2^{h(t)+1} - 1
\end{aligned}
$$

Generalized Induction

- *Generalized induction* is used to prove results about sets other than the integers that have the well-ordering property. (*explored in more detail in Chapter* 9)
- For example, consider an ordering on **N**× **N**, ordered pairs of nonnegative integers. Specify that (x_1, y_1) is less than or equal to (x_2, y_2) if either $x_1 < x_2$, or $x_1 = x_2$ and $y_1 < y_2$. This is called the *lexicographic ordering*.
- Strings are also commonly ordered by a *lexicographic ordering*.
- The next example uses generalized induction to prove a result about ordered pairs from **N**× **N**.

Generalized Induction

Example: Suppose that $a_{m,n}$ is defined for $(m,n) \in \mathbf{N} \times \mathbf{N}$ by $a_{0,0} = 0$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + n & \text{if } n > 0 \end{cases}.$$

Show that $a_{m,n} = m + n(n+1)/2$ is defined for all $(m,n) \in \mathbf{N} \times \mathbf{N}$.

Solution: Use generalized induction.

BASIS STEP: $a_{0,0} = 0 = 0 + (0 \cdot 1)/2$

INDUCTIVE STEP: Assume that $a_{m',n'} = m' + n'(n'+1)/2$ whenever (m',n') is less than (m,n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$.

- If $n = 0$, by the inductive hypothesis we can conclude
$$a_{m,n} = a_{m-1,n} + 1 = m - 1 + n(n+1)/2 + 1 = m + n(n+1)/2 .$$
- If $n > 0$, by the inductive hypothesis we can conclude
$$a_{m,n} = a_{m-1,n} + 1 = m + n(n-1)/2 + n = m + n(n+1)/2 .$$

Recursive Algorithms

Section 5.4

Section Summary

- Recursive Algorithms
- Proving Recursive Algorithms Correct
- Recursion and Iteration (*not yet included in overheads*)
- Merge Sort

Recursive Algorithms

Definition: An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.

- For the algorithm to terminate, the instance of the problem must eventually be reduced to some initial case for which the solution is known.

Recursive Factorial Algorithm

Example: Give a recursive algorithm for computing $n!$, where n is a nonnegative integer.

- **Solution**: Use the recursive definition of the factorial function.

procedure *factorial*(n: nonnegative integer)
if $n = 0$ **then return** 1
else return $n \cdot factorial\ (n - 1)$
{output is $n!$}

Recursive Exponentiation Algorithm

Example: Give a recursive algorithm for computing a^n, where a is a nonzero real number and n is a nonnegative integer.

Solution: Use the recursive definition of a^n.

procedure *power*(a: nonzero real number, n: nonnegative integer)
if $n = 0$ **then return** 1
else return $a \cdot power\,(a, n-1)$
{output is a^n}

Recursive GCD Algorithm

Example: Give a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$.

Solution: Use the reduction

$$\gcd(a,b) = \gcd(b \bmod a, a)$$

and the condition $\gcd(0,b) = b$ when $b > 0$.

```
procedure gcd(a,b: nonnegative integers
                          with a < b)
if a = 0 then return b
else return gcd (b mod a, a)
{output is gcd(a, b)}
```

Recursive Modular Exponentiation Algorithm

Example: Devise a a recursive algorithm for computing b^n **mod** m, where b, n, *and* m are integers with $m \geq 2$, $n \geq 0$, and $1 \leq b \leq m$.

- **Solution**: (*see text for full explanation*)

procedure *mpower*(b,m,n: integers with $b > 0$ and $m \geq 2$, $n \geq 0$)
if $n = 0$ **then**
    return 1
else if *n is even* **then**
    return $mpower(b,n/2,m)^2$ **mod** m
else
    return $(mpower(b,\lfloor n/2 \rfloor,m)^2$ **mod** $m \cdot b$ **mod** $m)$ **mod** m
{output is b^n **mod** m}

Recursive Binary Search Algorithm

Example: Construct a recursive version of a binary search algorithm.

Solution: Assume we have $a_1, a_2, \ldots, a_n$, an increasing sequence of integers. Initially i is 1 and j is n. We are searching for x.

procedure *binary search*(i, j, x : integers, $1 \leq i \leq j \leq n$)
$m := \lfloor (i + j)/2 \rfloor$
if $x = a_m$ **then**
    return m
else if ($x < a_m$ and $i < m$) **then**
    return *binary search*($i, m-1, x$)
else if ($x > a_m$ and $j > m$) **then**
    return *binary search*($m+1$,j,x)
else return 0
{output is location of x in $a_1, a_2, \ldots, a_n$ if it appears, otherwise 0}

Proving Recursive Algorithms Correct

- Both mathematical and strong induction are useful techniques to show that recursive algorithms always produce the correct output.

Example: Prove that the algorithm for computing the powers of real numbers is correct.

procedure *power*(a: nonzero real number, n: nonnegative integer)
if $n = 0$ **then return** 1
else return $a \cdot power\ (a, n - 1)$
{output is a^n}

Solution: Use mathematical induction on the exponent n.

BASIS STEP: $a^0 = 1$ for every nonzero real number a, and $power(a,0) = 1$.

INDUCTIVE STEP: The inductive hypothesis is that $power(a,k) = a^k$, for all $a \neq 0$.
Assuming the inductive hypothesis, the algorithm correctly computes a^{k+1}, since

$$power(a,k + 1) = a \cdot power\ (a, k) = a \cdot a^k = a^{k+1}\ .$$

Merge Sort

- *Merge Sort* works by iteratively splitting a list (with an even number of elements) into two sublists of equal length until each sublist has one element.
- Each sublist is represented by a balanced binary tree.
- At each step a pair of sublists is successively merged into a list with the elements in increasing order. The process ends when all the sublists have been merged.
- The succession of merged lists is represented by a binary tree.

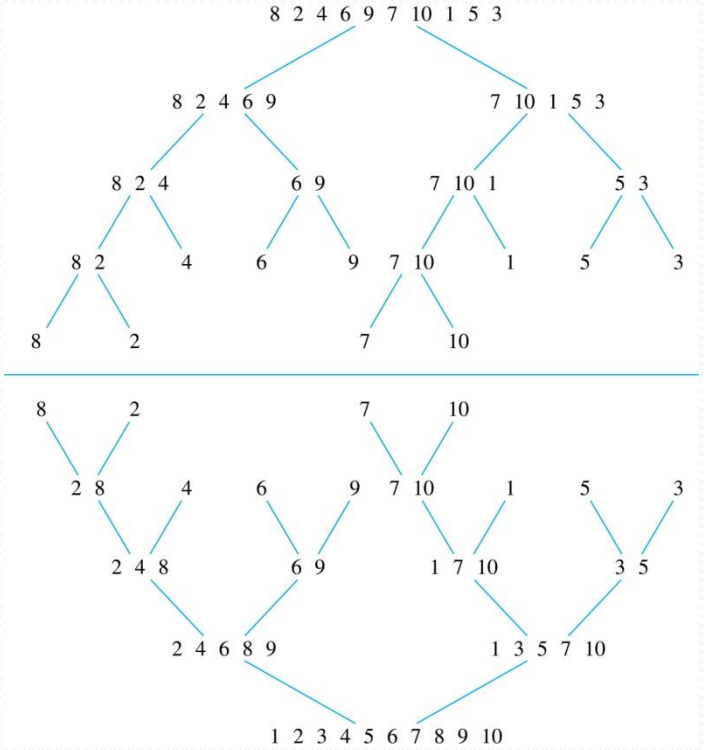
Merge Sort

Example: Use merge sort to put the list

8,2,4,6,9,7,10, 1, 5, 3

into increasing order.

Solution:



Recursive Merge Sort

Example: Construct a recursive merge sort algorithm.

Solution: Begin with the list of n elements L.

procedure $mergesort(L = a_1, a_2,\ldots,a_n)$
if $n > 1$ **then**
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, a_2,\ldots,a_m$
    $L_2 := a_{m+1}, a_{m+2},\ldots,a_n$
    $L := merge(mergesort(L_1), mergesort(L_2))$
{L is now sorted into elements in increasing order}

continued →

Recursive Merge Sort

- Subroutine *merge*, which merges two sorted lists.

```
procedure merge(L1, L2 :sorted lists)
L := empty list
while L1 and L2 are both nonempty
    remove smaller of first elements of L1 and L2 from its list;
          put at the right end of L
    if this removal makes one list empty
       then remove all elements from the other list and append them to L
return L {L is the merged list with the elements in increasing order}
```

Complexity of Merge: Two sorted lists with m elements and n elements can be merged into a sorted list using no more than $m + n - 1$ comparisons.

Merging Two Lists

Example: Merge the two lists 2,3,5,6 and 1,4.

Solution:

TABLE 1 Merging the Two Sorted Lists 2, 3, 5, 6 and 1, 4.

First List	*Second List*	*Merged List*	*Comparison*
2 3 5 6	1 4		1 < 2
2 3 5 6	4	1	2 < 4
3 5 6	4	1 2	3 < 4
5 6	4	1 2 3	4 < 5
5 6		1 2 3 4	
		1 2 3 4 5 6	

Complexity of Merge Sort

Complexity of Merge Sort: The number of comparisons needed to merge a list with n elements is $O(n \log n)$.

- For simplicity, assume that n is a power of 2, say 2^m.
- At the end of the splitting process, we have a binary tree with m levels, and 2^m lists with one element at level m.
- The merging process begins at level m with the pairs of 2^m lists with one element combined into 2^{m-1} lists of two elements. Each merger takes two one comparison.
- The procedure continues , at each level ($k = m$, $m-1$, $m-1$,...,3,2,1) 2^k lists with 2^{m-k} elements are merged into 2^{k-1} lists, with 2^{m-k+1} elements at level $k-1$.
  - We know (by the complexity of the merge subroutine) that each merger takes at most $2^{m-k} + 2^{m-k} - 1 = 2^{m-k+1} - 1$ comparisons.

continued →

Complexity of Merge Sort

- Summing over the number of comparisons at each level, shows that

$$\sum_{k=1}^{m} 2^{k-1}(2^{m-k+1}-1) = \sum_{k=1}^{m} 2^m - \sum_{k=1}^{m} 2^{k-1} = m2^m - (2^m - 1) = n \log n - n + 1,$$

because $m = \log n$ and $n = 2^m$.

(The expression $\sum_{k=1}^{m} 2^{k-1}$ in the formula above is evaluated as $2^m - 1$ using the formula for the sum of the terms of a geometric progression, from Section 2.4.)

- In Chapter 11, we'll see that the fastest comparison-based sorting algorithms have $O(n \log n)$ time complexity. So, merge sort achieves the best possible big-O estimate of time complexity.

Proving Inequalities

Example: Use mathematical induction to prove that $2^n < n!$, for every integer $n \geq 4$.

Solution: Let $P(n)$ be the proposition that $2^n < n!$.

- BASIS STEP: $P(4)$ is true since $2^4 = 16 < 4! = 24$.
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $2^k < k!$ for an arbitrary integer $k \geq 4$. To show that $P(k + 1)$ holds:

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k \\ &< 2 \cdot k! \qquad \textit{(by the inductive hypothesis)} \\ &< (k+1)k! \\ &= (k+1)! \end{aligned}$$

Therefore, $2^n < n!$ holds, for every integer $n \geq 4$. ◀

Note that here the basis step is $P(4)$, since $P(0)$, $P(1)$, $P(2)$, and $P(3)$ are all false.