Mat 2345

Week 10

Fall 2013

Student Responsibilities — Week 10

- **Reading**: Textbook, Section 3.7, 4.1, & 5.2
- **Assignments**: from Sections 4.1, 4.2, and Chapter 4 Supplementary Exercises (Pg 329–330)
- **Attendance**: Compellingly Encouraged

Week 10 Overview

- Sec 4.2. Strong Induction and Well–Ordering
- Sec 4.3. Recursive Definitions and Structural Induction

Use Induction to Prove the Theorem

A plane divided into regions by any number of distinct straight lines in **standard position** (no three of which intersect at the same point and no two of which are parallel), can be painted with black and white paint in such a way that any two regions having a common boundary will be painted in different colors.

Suppose the sequence $(s_0, s_1, s_2, \dots)$ satisfies the conditions $s_0 = a$ and $s_n = 2s_{n-1} + b$ for some constants a and b, and $\forall n \in \mathbb{N}$. Can we find a **closed form** (formula) to describe s_n?

Strong Induction

- Another form of proof by induction, **strong induction**, is similar to weak induction, but we assume $P(i)$ is true **for all** $1 \leq i \leq k$, and show $P(k+1)$ is true.

- Strong induction is used when $P(i)$ is defined in terms of more than just $P(i-1)$, e.g., when $P(i)$ is defined by combinations of previous $P_j,\ 1 \leq j < i$.

The Match Game

- The **Match Game**:

  1. Consists of **two players** and **two equal piles** of n matches
  2. The goal is to **remove the last match**
  3. The rule is that the players take turns removing any positive number of matches they want from **one** of the two piles

- We are going to use strong induction to prove:

  $P(n)$ — the second player can win when there are initially n matches in each pile.

The Match Game

BC Let $n = 1$.
The first player has only one choice, to remove the single match from one of the piles, leaving one match (in the other pile) for player two, and player two wins. $\surd$

IH Assume $P(j)$ is true $\forall \; 1 \leq j \leq k$

IS Show $P(k+1)$: that player two can win when there are initially $k+1$ matches in each pile — is true.

Match Game Proof, Cont.

- So suppose that there are $k+1$ matches in each of the two piles at the start of the game
- Further, suppose that player one removes r matches $(1 \leq r \leq k)$ from one of the piles, leaving $k+1-r$ matches in that pile
- The second player now removes r matches from the other pile, also leaving $k+1-r$ matches in that pile

Match Game Proof, Cont.

- Because $1 \leq k + 1 - r \leq k$, the second player can always win by the inductive hypothesis (and the fact that if player one takes all the matches in one pile, player two can do likewise).

Thus, the second player can always win when there are n matches in both piles.

Postage Problem

Theorem. Every amount of postage of 12 cents or more can be formed using just 4–cent and 5–cent stamps.

$$\boxed{\begin{array}{l} P(n) \text{ — postage of } n \text{ cents can be formed using} \\ \qquad\qquad \text{4–cent and 5–cent stamps } \forall \;\; n \geq 12. \end{array}}$$

Proof

BC $P(12)$ = three 4–cent stamps
$P(13)$ = two 4–cent and one 5–cent stamps
$P(14)$ = one 4–cent and two 5–cent stamps
$P(15)$ = three 5–cent stamps
This shows that $P(12), P(13), P(14)$, and $P(15)$ are true.

IH Assume $P(j)$ is true $\forall\ 12 \leq j \leq k, \quad k \geq 15$

Postage Proof, Cont.

IS Show $P(k+1)$ is true

- Using the inductive hypothesis, we can assume $P(k-3)$ is true, because $k-3 \geq 12$.
- To form postage of $k+1$ cents, we need only add another 4–cent stamp to the stamps used to form postage of $k-3$ cents.
- We have shown that if the inductive hypothesis is true, then $P(k+1)$ is true.
- Thus, $P(n)$ — postage of n cents can be formed using 4–cent and 5–cent stamps — is true $\forall \ n \geq 12$.

The Well-Ordering Property

Well–Ordering Property: Every nonempty set of non-negative integers has a least element.

This also means that any set of objects on which "less than" ($<$) is defined, has a "least" element — one that is "less than" all the other objects in the set.

Section 4.3 — Recursive Definitions

Recursive or **inductive definitions** of sets, and functions on recursively defined sets are similar:

- **Basis step**:
  - For **sets** — state the basic building blocks (BBBs) of the set
  - For **functions** — state the values of the function on the BBBs.
- **Inductive** or **recursive** step:
  - For **sets** — show how to build new things from old with some construction rules
  - For **functions** — show how to compute the value of a function on the new things that can be built knowing the value of the old things.

- **Extremal Clause**
  - For **sets** — if you can't build it with a finite number of applications of steps 1 and 2, then it isn't in the set.
  - For **functions** — a function defined on a recursively defined set does not require an extremal clause.

Note: To prove something is **in** the set, you must show how to **construct** it with a **finite number** of applications of the basis and inductive steps.

To prove something is **not** in the set is often more difficult.

Example: A Recursive Definition of $\mathbb{N}$

1. **Basis**: 0 is in $\mathbb{N}$ — 0 is the BBB
2. **Inductive definition**: if n is in $\mathbb{N}$, then so is $n + 1$
3. **Extremal clause**: If you can't construct it with a finite number of applications of steps 1 and 2, it isn't in $\mathbb{N}$.

Recursive Function Definitions

Now, given the recursive definition of $\mathbb{N}$, we can give recursive definitions of functions on $\mathbb{N}$.

One example is the **factorial** function, $f(n)$:

1. $f(0) = 1$
   the **initial condition** or the value of the function on the BBBs.

2. $f(n+1) = (n+1) * f(n)$
   the **recurrence** equation, how to define f on the new objects based on its value on old objects.

f is the **factorial function**: $f(n) = n!$ — note how it follows the recursive definition of $\mathbb{N}$

Proving Assertions

Proofs of assertions about inductively defined objects usually involve a **proof by induction**.

- Prove the assertion is true for the BBBs in the basis step.
- Prove that if the assertion is true for the old objects, it must be true for the new objects you can build from the old objects.
- Conclude the assertion must be true for all objects.

Example: $a^n, \quad n \in \mathbb{N}$

We define a^n inductively, where $n \in \mathbb{N}$:

- Basis: $a^0 = 1$
- Inductive definition: $a^{(n+1)} = a^n \times a$

Theorem. $\forall m \forall n [a^m a^n = a^{m+n}]$

Proof. Since the powers of a have been defined inductively, we must use a proof by induction somewhere.

Get rid of the first quantifier m by Universal Instantiation: assume m is arbitrary.

Now prove the remaining quantified assertion:
$\forall n [a^m a^n = a^{m+n}]$

Induction Proof: $\forall m \forall n [a^m a^n = a^{m+n}]$

- **Base Case**. Let $n = 0$.
  LHS. $a^m a^0 = a^m(1) = a^m$
  RHS. $a^{m+0} = a^m \quad \surd$
  Hence, the two sides are equal.

- **Inductive Hypothesis**. Assume for some arbitrary $k \geq 0$, that $a^m a^k = a^{m+k}$.

Induction Proof: $\forall m \forall n [a^m a^n = a^{m+n}]$

- **Inductive Step**. Show that $a^m a^{k+1} = a^{m+k+1}$

$$\begin{array}{rcll} a^m a^{k+1} & = & a^m(a^k a) & \text{inductive definition of } a^{n+1} \\ & = & (a^m a^k)a & \text{associativity of multiplication} \\ & = & a^{m+k} a & \text{IH, substitution} \\ & = & a^{(m+k)+1} & \text{inductive definition of powers of } a \\ & = & a^{m+k+1} & \text{associativity of addition} \end{array}$$

Thus, $a^m a^n = a^{m+n} \ \ \forall n \in \mathbb{N}$

Since m was arbitrary, by Universal Generalization,

$$\forall m \forall n [a^m a^n = a^{m+n}]$$

Example: the Fibonacci Sequence

The **Fibonacci sequence** is defined recursively:

1. **Basis**. $f(0) = f(1) = 1$
   two initial conditions

2. **Inductive definition**. $f(n+1) = f(n) + f(n-1)$
   the recurrence equation

The sequence begins: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Example: The Set of Strings over an Alphabet

- A **finite alphabet**, Σ, is defined as a finite set of symbols
- A **string** (over an alphabet, Σ) is a sequence of 0 or more characters from Σ.
- The empty or null string is denoted: λ (lambda)
- We can define the **set of all strings** over a **finite alphabet** Σ recursively.
- This set is called Σ^* (Sigma star), and is equivalent to $\Sigma^+ \cup \lambda$

Recursive Definition of Strings

- The recursive definition of Σ^*:

1. **Basis**: The empty string, λ, is in Σ^*
2. **Inductive definition**: If w is in Σ^* and a is a symbol in Σ, then wa is in Σ^* — (a string + a letter)

   Note: we can concatenate a on the right or left of w, but it makes a difference in proofs, since concatenation is not commutative.
3. **Extremal clause**: infinitely long strings cannot be in Σ^*. (Why not?)

Example: Let $\Sigma = \{a, b\}$. Then $aab \in \Sigma^*$

Proof: We construct *aab* with a finite number of applications of the basis and inductive steps in the definition of Σ^*.

1. $\lambda \in \Sigma^*$ by the basis step
2. By step 1, the induction clause in the definition of Σ^* and the fact that a is in Σ, we can conclude that $\lambda a = a \in \Sigma^*$.
3. Since $a \in \Sigma^*$ from step 2, and $a \in \Sigma$, applying the induction clause again, we conclude that $aa \in \Sigma^*$.
4. Since $aa \in \Sigma^*$ from step 3, and $b \in \Sigma$, applying the induction clause again, we conclude that $aab \in \Sigma^*$.

We have shown $aab \in \Sigma^*$ with a finite number of applications of the basis and induction clauses in the definition, thus we are done.

Example: Well Formed Parenthesis Strings

An inductive definition of the well–formed parenthesis strings P:

1. **Basis**: $(\ ) \in P$
2. **Inductive definition**: if $w \in P$, then so are $(\ )w$, (w), and $w(\ )$
3. **Extremal clause**: must be able to construct such strings in a finite application of steps 1 & 2

Prove: $(()()) \in P$

1. $()\in P$ by basis clause
2. $()() \in P$ by step 1 and the induction clause
3. $(()()) \in P$ by step 2 and the induction clause

QED.

Note: $))(() \notin P$ — why not? Can you prove it? (hint: what can we say about the length of strings in P? How can we order the strings in P?)

Bit Strings

The set S of bit strings with no more than a single 1 in them can be defined recursively:

1. **Basis**: λ, 0, 1 $\in S$
2. **Inductive definition**: if w is in S, then so are $0w$ and $w0$
3. **Extremal clause**: must be able to construct such strings in a finite application of steps 1 & 2

Can we prove that $0010 \in S$?