# CSE 311 Foundations of Computing I

Lecture 16

Induction and Recursive Definitions

Autumn 2012



# Announcements

- Reading assignments
  - Today:
    - 5.2, 5.3 $7^{th}$ Edition
    - 4.2, 4.3 $6^{th}$ Edition
    - 3.3, 3.4 $5^{th}$ Edition
- Midterm Friday, Nov 2
  - Closed book, closed notes
  - Practice midterm available on the Web
  - Cover class material up to and including induction.
- Extra office hours Thursday (midterm review)
  - 3:30 pm, Dan Suciu, Gowen 201
  - 4:30 pm, Richard Anderson, Gowen 201



# Highlights from last lecture

- Mathematical Induction

$$\begin{array}{l} P(0) \\ \underline{\forall k\geq 0\ (P(k) \rightarrow P(k+1))} \\ \therefore\ \forall n\geq 0\ \ P(n) \end{array}$$

- Induction proof layout:
  1. By induction we will show that P(n) is true for every n≥0
  2. Base Case: Prove P(0)
  3. Inductive Hypothesis: Assume that P(k) is true for some arbitrary integer k ≥ 0
  4. Inductive Step: Prove that P(k+1) is true using Inductive Hypothesis that P(k) is true
  5. Conclusion: Result follows by induction



# Harmonic Numbers

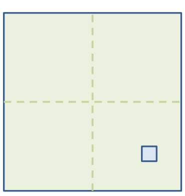
$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots \frac{1}{n} = \sum_{k=1}^{n} \frac{1}{k}$$

Prove $H_{2^n} \geq 1 + \frac{n}{2}$ for all $n \geq 1$



# Cute Application: Checkerboard tiling with Trinominos

Prove that a $2^n \times 2^n$ checkerboard with one square removed can be tiled with: 





# Strong Induction

$$\begin{array}{l} P(0) \\ \underline{\forall k\ ((P(0) \wedge P(1) \wedge P(2) \wedge \ldots \wedge P(k)) \rightarrow P(k+1))} \\ \therefore\ \forall n\ P(n) \end{array}$$

Follows from ordinary induction applied to
$Q(n) = P(0) \wedge P(1) \wedge P(2) \wedge \ldots \wedge P(n)$

## Strong Induction English Proofs

1. By induction we will show that P(n) is true for every $n \geq 0$
2. Base Case: Prove P(0)
3. Inductive Hypothesis:
   Assume that for some arbitrary integer $k \geq 0$, P(j) is true for every j from 0 to k
4. Inductive Step:
   Prove that P(k+1) is true using the Inductive Hypothesis (that P(j) is true for all values $\leq k$)
5. Conclusion: Result follows by induction



## Every integer ≥ 2 is the product of primes



## Recursive Definitions of Functions

- $F(0) = 0$; $F(n + 1) = F(n) + 1$;
- $G(0) = 1$; $G(n + 1) = 2 \times G(n)$;
- $0! = 1$; $(n+1)! = (n+1) \times n!$
- $H(0) = 1$; $H(n + 1) = 2^{H(n)}$



## Fibonacci Numbers

- $f_0 = 0$; $f_1 = 1$; $f_n = f_{n-1} + f_{n-2}$



## Bounding the Fibonacci Numbers

- Theorem: $2^{n/2-1} \leq f_n < 2^n$ for $n \geq 2$



## Fibonacci numbers and the running time of Euclid's algorithm

- Theorem: Suppose that Euclid's algorithm takes n steps for gcd($a$,$b$) with $a$>$b$, then $a \geq f_{n+1}$
- Set $r_{n+1}=a$, $r_n=b$ then Euclid's alg. computes

$$r_{n+1} = q_n r_n + r_{n-1}$$
$$r_n = q_{n-1} r_{n-1} + r_{n-2}$$
$$\vdots$$
$$r_3 = q_2 r_2 + r_1$$
$$r_2 = q_1 r_1$$

each quotient $q_i \geq 1$
$r_1 \geq 1$

## Recursive Definitions of Sets

- Recursive definition
  - Basis step: $0 \in S$
  - Recursive step: if $x \in S$, then $x + 2 \in S$
  - Exclusion rule: Every element in S follows from basis steps and a finite number of recursive steps



## Recursive definitions of sets

Basis: $6 \in S$; $15 \in S$;
Recursive: if $x, y \in S$, then $x + y \in S$;

Basis: $[1, 1, 0] \in S$, $[0, 1, 1] \in S$;
Recursive:
if $[x, y, z] \in S$, $\alpha$ in R, then $[\alpha x, \alpha y, \alpha z] \in S$
if $[x_1, y_1, z_1], [x_2, y_2, z_2] \in S$
then $[x_1 + x_2, y_1 + y_2, z_1 + z_2]$

Powers of 3



## Strings

- The set $\Sigma^*$ of strings over the alphabet $\Sigma$ is defined
  - Basis: $\lambda \in S$ ($\lambda$ is the empty string)
  - Recursive: if $w \in \Sigma^*$, $x \in \Sigma$, then $wx \in \Sigma^*$



## Function definitions on recursively defined sets

$\text{Len}(\lambda) = 0$;
$\text{Len}(wx) = 1 + \text{Len}(w)$; for $w \in \Sigma^*$, $x \in \Sigma$

$\text{Concat}(w, \lambda) = w$ for $w \in \Sigma^*$
$\text{Concat}(w_1,w_2x) = \text{Concat}(w_1,w_2)x$ for $w_1$, $w_2$ in $\Sigma^*$, $x \in \Sigma$