Fall 2015
Lecture 17: Strong induction & Recursive definitions

YOUR PARTY ENTERS THE TAVERN.

I GATHER EVERYONE AROUND A TABLE. I HAVE THE ELVES START WHITTLING DICE AND GET OUT SOME PARCHMENT FOR CHARACTER SHEETS.

HEY, NO RECURSING.
Midterm review session Sunday @ 1:00 pm (EEB 105)

MIDTERM MONDAY (IN THIS ROOM, USUAL TIME)

No office hours on Monday/Wednesday

Closed book.
One page (front and back) of notes allowed.

Exam includes induction!
Homework #5 is due on Friday, Nov 13th.
review: strong induction

\[ P(0) \]
\[ \forall k \left( (P(0) \land P(1) \land P(2) \land \cdots \land P(k)) \rightarrow P(k + 1) \right) \]
\[ \therefore \forall n \ P(n) \]

Follows from ordinary induction applied to

\[ Q(n) = P(0) \land P(1) \land P(2) \land \cdots \land P(n) \]
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
Fibonacci numbers

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \\
  f_n &= f_{n-1} + f_{n-2} \quad \text{for all } n \geq 2
\end{align*}
\]
bounding the Fibonacci numbers

Theorem: \( f_n < 2^n \) for all \( n \geq 2 \)

\[ P(n) = \text{“} f_n < 2^n \text{“} \]

Base case: \( P(0): 0 = f_0 < 2^0 = 1 \)  
\( P(1): 1 = f_1 < 2^1 = 2 \)

IH: Assume for some \( k \geq 1 \) 
\( P(j) \) holds for \( 0 \leq j \leq k \)

IS: \( f_{k+1} = f_k + f_{k-1} \) \( \text{by def’}n \)
\( \leq 2^k + 2^{k-1} \) \( \text{by IH, } P(k), P(k-1) \)
\( = 2^{k-1}(2+1) < 4 \cdot 2^{k-1} = 2^{k+1} \)
bounding the Fibonacci numbers

**Theorem:** \( \frac{n-1}{2^2} \leq f_n < 2^n \) for all \( n \geq 2 \)

\[
P(n) = \left\{ \begin{array}{ll}
\text{\( f_n \geq 2 \)} & \\
\text{\( 1 \geq \frac{n-1}{2^2} \)} & \\
\text{\( 1 \geq \frac{2^n}{2} \)} & \\
\end{array} \right.
\]

**Base case:** \( P(2) = \left\{ \begin{array}{ll}
1 \geq \frac{2}{4} = 1 & \checkmark \quad P(2)
\end{array} \right. \)

**IH:** Assume for some \( k \geq 2 \) that

- \( P(j) \) holds for all \( 2 \leq j \leq k \).
- \( f_k \geq \frac{k-1}{2^2} \) \( \forall k \leq j \leq k \) by defn
- \( k+1 \geq 2 \) and \( k+1 \geq 2 \) \( \forall k \geq 2 \)

\( f_{k+1} = f_k + f_{k-1} \)

**Case:**

- \( P(k) \) \( \forall k \geq 3 \)

\[
\begin{align*}
\frac{k}{2} - 1 & \geq \frac{k-1}{2^2} + \frac{k-1}{2^2} \\
& = \frac{1}{2} \left(\frac{k}{2} + \frac{k-1}{2} \right) 2^{k+1}
\end{align*}
\]

By str. ind., \( P(n) \) \( \forall n \geq 2 \).

\[
\frac{k}{2} - 1 \geq \frac{k}{2} + 1 \geq \frac{k}{2} \geq 2
\]
Theorem:  \( 2^{n/2-1} \leq f_n < 2^n \) for all \( n \geq 2 \)

Proof:
1. Let \( P(n) \) be "\( 2^{n/2-1} \leq f_n < 2^n \). By (strong) induction we prove \( P(n) \) for all \( n \geq 2 \).
2. **Base Case:** \( P(2) \) is true: \( f_2=1, \quad 2^{2/2-1}=2^0=1 \leq f_2, \quad 2^2=4>f_2 \)
3. **Ind.Hyp:** Assume \( 2^{j/2-1} \leq f_j < 2^j \) for all integers \( j \) with \( 2 \leq j \leq k \) for some arbitrary integer \( k \geq 2 \).
4. **Ind. Step:** *Goal: Show* \( 2^{(k+1)/2-1} \leq f_{k+1} < 2^{k+1} \)

   *Case k=2:* \( P(3) \) is true: \( f_3=f_2+f_1=1+1=2, \quad 2^{3/2-1}=2^{1/2} \leq 2 = f_3, \quad 2^3=8 > f_3 \)

   *Case k\(\geq3\):*

   \[
   f_{k+1} = f_k + f_{k-1} \geq 2^{k/2-1} + 2^{(k-1)/2-1} \quad \text{by I.H. since } k-1 \geq 2
   \]

   \[
   > 2^{(k-1)/2-1} + 2^{(k-1)/2 - 1} = 2 \cdot 2^{(k-1)/2-1} = 2^{(k+1)/2-1}
   \]

   \[
   f_{k+1} = f_k + f_{k-1} < 2^k + 2^{(k-1)} \quad \text{by I.H. since } k-1 \geq 2
   \]

   \[
   < 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}
   \]
The divisibility theorem

**Theorem:** For any integers $n$ and $d \geq 0$, there are integers $q$ and $r$ such that $n = dq + r$ and $0 \leq r \leq d - 1$.

\[ P(n) = \]

**Base case:** $P(0)$: For any $d > 0$, $0 = d \cdot 0 + 0$ so $P(0)$ holds.

**IH:** For some $k \geq 0$, $P(j)$ holds for all $0 \leq j \leq k$.

**IH:** Let $d > 0$ be arbitrary

\( \text{[Goal: } k + 1 = dq + r \text{ for some } q, r \text{ } 0 \leq r \leq d - 1] \)

- If $k + 1 < d$, $k + 1 = d \cdot 0 + k + 1 \implies P(k + 1)$
- If $k + 1 \geq d$, then $k + 1 = a + d$ for some $0 \leq a \leq k$
  By $P(a)$: $a = dq' + r \implies k + 1 = d(q' + 1) + r$
running time of Euclid’s algorithm
running time of Euclid’s algorithm

Theorem: Suppose that Euclid’s algorithm takes \( n \) steps for \( \text{gcd}(a, b) \) with \( a > b \), then \( a \geq f_{n+1} \).

Proof:

Set \( r_{n+1} = a, r_n = b \) then Euclid’s algorithm computes

\[
\begin{align*}
    r_{n+1} &= q_n r_n + r_{n-1} \\
    r_n &= q_{n-1} r_{n-1} + r_{n-2} \\
    & \vdots \\
    r_3 &= q_2 r_2 + r_1 \\
    r_2 &= q_1 r_1
\end{align*}
\]

Each quotient \( q_i \geq 1 \)

\( r_1 \geq 1 \)