



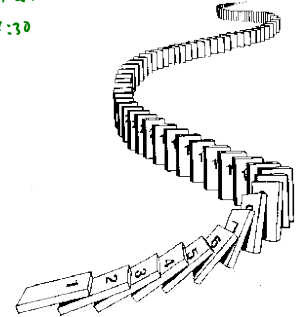
# Foundations of Computing I

\* All slides are a combined effort between previous instructors of the course

## CSE 311: Foundations of Computing

### Lecture 17: Strong Induction

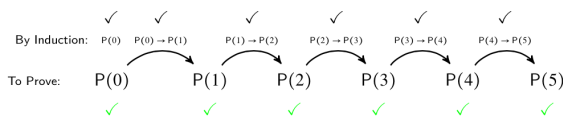
Review Session #2:  
Tomorrow 2:30 - 4:30  
in THO 101



### Induction Is A Rule of Inference

Domain: Natural Numbers  $P(0)$   
 $\forall k (P(k) \rightarrow P(k + 1))$   
 $\therefore \forall n P(n)$

How does this technique prove P(5)?



First, we prove **P(0)**.  
 Since  $P(n) \rightarrow P(n+1)$  for all  $n$ , we have  $P(0) \rightarrow P(1)$ .  
 Since **P(0)** is true and  $P(0) \rightarrow P(1)$ , by Modus Ponens, **P(1)** is true.  
 Since  $P(n) \rightarrow P(n+1)$  for all  $n$ , we have  $P(1) \rightarrow P(2)$ .  
 Since **P(1)** is true and  $P(1) \rightarrow P(2)$ , by Modus Ponens, **P(2)** is true.

Domain: Natural Numbers

### Induction Is A Rule of Inference...Again

1. **P(0)** ("Given")
2.  $\forall n (P(n) \rightarrow P(n + 1))$  ("Given")
3. **P(1)** (MP: 2, 1)
4. **P(2)** (MP: 2, 3)
5. **P(3)** (MP: 2, 4)
6. **P(4)** (MP: 2, 5)

### Induction Is A Rule of Inference

Domain: Natural Numbers

"Induction"

1. **P(0)** ("Given")
2.  $\forall n (P(n) \rightarrow P(n + 1))$  ("Given")
3. **P(1)** (MP: 2, 1)
4. **P(2)** (MP: 2, 3)
5. **P(3)** (MP: 2, 4)
6. **P(4)** (MP: 2, 5)

Notice how when we use regular induction, we're already proving the things necessary to use strong induction.

This is no extra work with a benefit!

"Strong Induction"

1. **P(0)** ("Given")
2.  $\forall n ((P(0) \wedge P(1) \wedge \dots \wedge P(n)) \rightarrow P(n + 1))$  ("Given")
3. **P(1)** (MP: 2, 1)
4. **P(2)** (MP: 2, 1, 3)
5. **P(3)** (MP: 2, 1, 3, 4)
6. **P(4)** (MP: 2, 1, 3, 4, 5)

### Strong Induction

$$\forall k ((P(0) \wedge P(1) \wedge P(2) \wedge \dots \wedge P(k)) \rightarrow P(k + 1))$$

$$\therefore \forall n P(n)$$

## Strong Induction English Proof

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 $n \geq 2$ can be expressed as a product of primes.

Let  $P(n)$  be " $n = p_0 p_1 \dots p_j$ , where  $p_0, p_1, \dots, p_j$  are prime."

We go by strong induction on  $n$ .

Base Case ( $n=2$ ): 2 is prime which is product of primes.

Induction Hypothesis: Suppose  $P(2) \wedge P(3) \wedge \dots \wedge P(k)$  for some  $k \in \mathbb{N}$  /  $k \geq 2$

Induction Step: We go by cases.

$$k+1 = p_1 p_2 \dots p_n$$

$$k = q_1 q_2 \dots q_m$$

(choose  $a, b$ , where  $a \leq k+1$ ,  $b \leq k$ ,  $a \neq 1$ ,  $b \neq 1$ .)

$$k+1 = a \cdot b$$

We know (by IH)...

All numbers smaller than  $k$  can be expressed as a product of primes.

We're trying to get...

$k$  can be expressed as a product of primes.

## Every $n \geq 2$ can be expressed as a product of primes.

Let  $P(n)$  be " $n = p_0 p_1 \dots p_j$ , where  $p_0, p_1, \dots, p_j$  are prime."

We go by induction on  $n$ .

Base Case ( $n=2$ ): Note that 2 is prime (which means it's a product of primes).

Induction Hypothesis: Suppose that  $P(2), P(3), \dots, P(k-1)$  are true for some  $k \geq 2$ .

Induction Step: We go by cases.

Case 1 ( $k$  is prime):

Then, since  $k$  is prime,  $k$  is a product of primes.

Case 2 ( $k$  is composite):

Then, by definition of composite, we have non-trivial  $1 < a, b < k$  such that  $k = ab$ . Since  $a$  and  $b$  are between 2 and  $k-1$ , we know  $P(2)$  and  $P(k-1)$  are true. So, we have:

$$a = p_0 p_1 \dots p_j \text{ and } b = p_{j+1} p_{j+2} \dots p_{j+\ell}$$

Then,  $k = ab = p_0 p_1 \dots p_j p_{j+1} p_{j+2} \dots p_{j+\ell}$

So,  $k$  can be expressed as a product of primes.

So,  $P(n)$  is true for all  $n \geq 2$  is true by induction.

We know (by IH)...

All numbers smaller than  $k$  can be expressed as a product of primes.

We're trying to get...

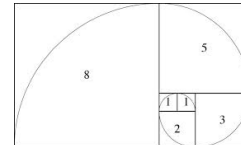
$k$  can be expressed as a product of primes.

## Fibonacci Numbers

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$



## Bounding the Fibonacci Numbers

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

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Let  $P(n)$  be " $2^{n/2-1} \leq f_n$ " for all  $n \geq 2$ .

We go by strong induction on  $n$ .

Base Case:

$$\text{note } 2^{2/2-1} = 2^0 = 1 = 1 = f_0 + f_1 = f_2$$

note

Induction Hypothesis:

Suppose  $P(2) \wedge P(3) \wedge \dots \wedge P(k)$  for some  $k \geq 3$   $k \in \mathbb{N}$

## Bounding the Fibonacci Numbers

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

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Let  $P(n)$  be " $2^{n/2-1} \leq f_n$ " for all  $n \geq 2$ .

We go by strong induction on  $n$ .

Induction Step:

$$\text{note } 2^{(k+1)/2-1} \leq$$

$$\downarrow$$

$$= 2^{k/2-1} + 2^{(k-1)/2-1}$$

$$k+1 \geq 2$$

$$\leq f_k + f_{k-1}$$

$$= f_{k+1}$$

we know  $2^{k/2-1} \leq f_k$   
 $2^{(k-1)/2-1} \leq f_{k-1}$   
 vtp  $2^{(k+1)/2-1} \leq f_{k+1}$

## Bounding the Fibonacci Numbers

Theorem:

$$f_n < 2^n \text{ for all } n \geq 2$$

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Let  $P(n)$  be " $f_n < 2^n$ " for all  $n \geq 2$ . We go by induction on  $n$ .

Base case ( $n=2$ ):

Note  $f_2 = f_0 + f_1 = 0 + 1 = 1 < 4 = 2^2$ , so,  $P(2)$  is true.

Induction Hypothesis: Suppose  $P(2) \wedge P(3) \wedge \dots \wedge P(k)$  for some  $k \in \mathbb{N}$ , where  $k \geq 3$ .

IS:  $f_{k+1} = f_k + f_{k-1}$  by def.

$$< 2^k + 2^{k-1} \text{ by IH.}$$

$$= \frac{3}{2} \cdot 2^k$$

$$< 2 \cdot 2^k$$

$$= 2^{k+1}$$

## Bounding the Fibonacci Numbers

Define  $f_n$  as:

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Theorem:

$$2^{n/2-1} \leq f_n \text{ and } f_n < 2^n$$

for all  $n \geq 2$

Proof:

Let  $P(n)$  be " $2^{n/2-1} \leq f_n$  and  $f_n < 2^n$ " for all  $n \geq 2$ .

We go by strong induction on  $n$ .

Base Case:  $2^{2/2-1} = 2^0 = 1 \leq 0 + 1 = f_2$ , and

$$f_2 = 0 + 1 = 1 < 4 = 2^2. \text{ So, } P(2) \text{ is true.}$$

Induction Hypothesis:

Suppose  $P(j)$  for all integers  $j$  s.t.  $2 \leq j \leq k$  for some  $k \geq 2$ .

Induction Step: We want to show  $2^{(k+1)/2-1} \leq f_{k+1}$  and  $f_{k+1} < 2^{k+1}$

## Bounding the Fibonacci Numbers

Define  $f_n$  as:

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Theorem:

$$2^{n/2-1} \leq f_n \text{ and } f_n < 2^n$$

for all  $n \geq 2$

Induction Step: We want to show  $2^{(k+1)/2-1} \leq f_{k+1}$  and  $f_{k+1} < 2^{k+1}$

If  $k+1=3$ ,  $2^{3/2-1} = 2^{1/2} \leq 2 = 1 + 1 = f_3$ , and

$$f_3 = 1 + 1 = 2 < 8 = 2^3. \text{ So, } P(3) \text{ is true.}$$

Otherwise, note that  $f_{k+1} = f_k + f_{k-1}$  by definition.

Taking each inequality separately:

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} < 2^k + 2^{k-1} && \text{(by IH)} \\ &< 2^k + 2^k && (2^{k-1} < 2^k) \\ &= 2^{k+1} \end{aligned}$$

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \geq 2^{k/2-1} + 2^{(k-1)/2-1} && \text{(by IH)} \\ &\geq 2^{(k-1)/2-1} + 2^{(k-1)/2-1} && \text{(Because } 2^{k/2-1} > 2^{(k-1)/2-1}\text{)} \\ &= 2 \cdot 2^{(k-1)/2-1} && \text{(Combining terms)} \\ &= 2^{2/2+(k-1)/2-1} && \text{(Multiplying)} \\ &= 2^{(k+1)/2-1} \end{aligned}$$

So, the claim is true by strong induction.

## Running time of Euclid's algorithm

**Theorem:** Suppose that Euclid's Algorithm takes  $n$  steps for  $\gcd(a,b)$  with  $a > b$ . Then,  $a \geq f_{n+1}$ .

We go by strong induction on  $n$ .

Let  $P(n)$  be " $\gcd(a,b)$  with  $a > b$  takes  $n$  steps  $\rightarrow a \geq f_{n+1}$ " for all  $n \geq 1$ .

Base Case:

If Euclid's Algorithm on  $a, b$ , with  $a > b$ , takes 1 step, then it must be the case that  $b \mid a$ .

Note that  $f_2 = 1$ .

Note that if  $a$  were 0, then  $\gcd(0, b)$ , which takes zero steps. So, the smallest possible value for  $a$  is 1, which is  $f_2$ .

Induction Hypothesis: Suppose  $P(j)$  for all integers  $j$  s.t.  $1 \leq j \leq k$  for some  $k \geq 1$ .

Induction Step: We want to show if  $\gcd(a,b)$  takes  $k+1$  steps, then  $a \geq f_{k+2}$ . If  $k = 2$ , note that  $a > 1$ , because  $\gcd(1, b)$  takes one step. Also,  $f_3 = 2$ .

## Running time of Euclid's algorithm

**Theorem:** Suppose that Euclid's Algorithm takes  $n$  steps for  $\gcd(a,b)$  with  $a > b$ . Then,  $a \geq f_{n+1}$ .

Since the algorithm took  $k+1$  steps, let's give them names:

Say  $r_{k+1} = a$  and  $r_k = b$ , and  $r_i = r_{i-1} \bmod r_{i-2}$ .

So,  $\gcd(a, b) = \gcd(r_{k+1}, r_k)$

$$= \gcd(r_k, r_k \bmod r_{k+1}) = \gcd(r_k, r_{k-1})$$

$$= \gcd(r_{k-1}, r_{k-1} \bmod r_k) = \gcd(r_{k-1}, r_{k-2})$$

= ...

Writing these as equations, we have:

$$r_{k+1} = q_k r_k + r_{k-1}$$

$$r_k = q_{k-1} r_{k-1} + r_{k-2}$$

...

$$r_3 = q_2 r_2 + r_1$$

$$r_2 = q_1 r_1$$

Note that  $q_i \geq 1$ ,  $r_i \geq 1$ .

Note that after one iteration of the algorithm, we're left with  $\gcd(r_k, r_{k-1})$  which takes  $k$  steps.

By the IH,  $r_k \geq f_{k+1}$ . So,

$$r_{k+1} = q_k r_k + r_{k-1} \text{ (by gcd algorithm)}$$

$$\geq q_k f_{k+1} + f_k \text{ (by IH)}$$

$$\geq f_{k+1} + f_k \text{ (} q_k \geq 1\text{)}$$

$$\geq f_{k+2} \text{ (definition of } f\text{)}$$