Lecture 18: Recursively Defined Functions & Sets
Last time: Fibonacci Numbers

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \\
  f_n &= f_{n-1} + f_{n-2} \quad \text{for all } n \geq 2
\end{align*}
\]
Last Time: Upper Bound \( f_n < 2^n \) for all \( n \geq 0 \)

1. Let \( P(n) \) be “\( f_n < 2^n \)”.
   We prove that \( P(n) \) is true for all integers \( n \geq 0 \) by strong induction.

2. Base Case: \( f_0 = 0 < 1 = 2^0 \) so \( P(0) \) is true.

3. Inductive Hypothesis: Assume that for some arbitrary integer \( k \geq 0 \), we have \( f_j < 2^j \) for every integer \( j \) from 0 to \( k \).

4. Inductive Step: **Goal: Show \( P(k+1) \); that is, \( f_{k+1} < 2^{k+1} \)**
   - **Case \( k+1 = 1 \):** Then \( f_1 = 1 < 2 = 2^1 \) so \( P(k+1) \) is true here.
   - **Case \( k+1 \geq 2 \):** Then \( f_{k+1} = f_k + f_{k-1} \) by definition.
     \[ f_{k+1} < 2^k + 2^{k-1} \text{ by the IH since } k-1 \geq 0 \]
     \[ < 2^k + 2^k = 2 \cdot 2^k = 2^{k+1} \]
     so \( P(k+1) \) is true in this case.
   These are the only cases so \( P(k+1) \) follows.

5. Therefore by strong induction,
   \( f_n < 2^n \) for all integers \( n \geq 0 \).

\[
\begin{align*}
f_0 &= 0 & f_1 &= 1 \\
f_n &= f_{n-1} + f_{n-2} \text{ for all } n \geq 2
\end{align*}
\]
Inductive Proofs with Multiple Base Cases

1. “Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by induction.”

2. “Base Cases:” Prove $P(b), P(b + 1), ..., P(c)$

3. “Inductive Hypothesis:”
   Assume $P(k)$ is true for an arbitrary integer $k \geq c$.

4. “Inductive Step:” Prove that $P(k + 1)$ is true:
   Use the goal to figure out what you need.
   Make sure you are using I.H. and point out where you are using it. (Don’t assume $P(k + 1)$ !!)

5. “Conclusion: $P(n)$ is true for all integers $n \geq b$”
Inductive Proofs With Multiple Base Cases

1. “Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by strong induction.”

2. “Base Cases:” Prove $P(b), P(b + 1), ..., P(c)$

3. “Inductive Hypothesis:
   Assume that for some arbitrary integer $k \geq c$
   
   $P(j)$ is true for every integer $j$ from $b$ to $k$”

4. “Inductive Step:” Prove that $P(k + 1)$ is true:
   
   Use the goal to figure out what you need. 
   
   Make sure you are using I.H. (that $P(b), ..., P(k)$ are true) and point out where you are using it.
   
   (Don’t assume $P(k + 1)$ !!)

5. “Conclusion: $P(n)$ is true for all integers $n \geq b$”
Bounding Fibonacci II: $f_n \geq 2^{n/2} - 1$ for all $n \geq 2$

1. Let $P(n)$ be “$f_n \geq 2^{n/2} - 1$.” We prove that $P(n)$ is true for all integers $n \geq 2$ by strong induction.

$$f_0 = 0 \quad f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$
Bounding Fibonacci II: \( f_n \geq 2^{n/2} - 1 \) for all \( n \geq 2 \)

1. Let \( P(n) \) be “\( f_n \geq 2^{n/2} - 1 \)”. We prove that \( P(n) \) is true for all integers \( n \geq 2 \) by strong induction.

2. Base Case: \( f_2 = f_1 + f_0 = 1 \) and \( 2^{2/2} - 1 = 2^0 = 1 \) so \( P(2) \) is true.

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \\
  f_n &= f_{n-1} + f_{n-2} & \text{for all } n \geq 2
\end{align*}
\]
Bounding Fibonacci II: \( f_n \geq 2^{n/2} - 1 \) for all \( n \geq 2 \)

1. Let \( P(n) \) be “\( f_n \geq 2^{n/2} - 1 \)”.

   We prove that \( P(n) \) is true for all integers \( n \geq 2 \) by strong induction.

2. Base Case: \( f_2 = f_1 + f_0 = 1 \) and \( 2^{2/2} - 1 = 2^0 = 1 \) so \( P(2) \) is true.

3. Inductive Hypothesis: Assume that for some arbitrary integer \( k \geq 2 \), \( P(j) \) is true for every integer \( j \) from 2 to \( k \).

\[
\begin{align*}
  f_0 &= 0 \quad f_1 = 1 \\
  f_n &= f_{n-1} + f_{n-2} \quad \text{for all } n \geq 2
\end{align*}
\]
Bounding Fibonacci II: \( f_n \geq 2^{n/2} - 1 \) for all \( n \geq 2 \)

1. Let \( P(n) \) be “\( f_n \geq 2^{n/2} - 1 \)”. We prove that \( P(n) \) is true for all integers \( n \geq 2 \) by strong induction.

2. Base Case: \( f_2 = f_1 + f_0 = 1 \) and \( 2^{2/2} - 1 = 2^0 = 1 \) so \( P(2) \) is true.

3. Inductive Hypothesis: Assume that for some arbitrary integer \( k \geq 2 \), \( P(j) \) is true for every integer \( j \) from 2 to \( k \).

4. Inductive Step: **Goal: Show** \( P(k+1) \); that is, \( f_{k+1} \geq 2^{(k+1)/2} - 1 \)

\[
\begin{align*}
f_0 &= 0 & f_1 &= 1 \\
f_n &= f_{n-1} + f_{n-2} & \text{for all} & n \geq 2
\end{align*}
\]
Bounding Fibonacci II: \( f_n \geq 2^{n/2} - 1 \) for all \( n \geq 2 \)

1. Let \( P(n) \) be “\( f_n \geq 2^{n/2} - 1 \).” We prove that \( P(n) \) is true for all integers \( n \geq 2 \) by strong induction.

2. Base Case: \( f_2 = f_1 + f_0 = 1 \) and \( 2^{2/2} - 1 = 2^0 = 1 \) so \( P(2) \) is true.

3. Inductive Hypothesis: Assume that for some arbitrary integer \( k \geq 2 \), \( P(j) \) is true for every integer \( j \) from 2 to \( k \).

4. Inductive Step: Goal: Show \( P(k+1) \); that is, \( f_{k+1} \geq 2^{(k+1)/2} - 1 \)

   No need for cases for the definition here:
   \[ f_{k+1} = f_k + f_{k-1} \text{ since } k+1 \geq 2 \]

   Now just want to apply the IH to get \( P(k) \) and \( P(k-1) \)

   Problem: Though we can get \( P(k) \) since \( k \geq 2 \), \( k-1 \) may only be 1 so we can’t conclude \( P(k-1) \)

   Solution: Separate cases for when \( k-1 = 1 \) (or \( k+1 = 3 \)).

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \\
  f_n &= f_{n-1} + f_{n-2} \text{ for all } n \geq 2
\end{align*}
\]
Bounding Fibonacci II: \( f_n \geq 2^{n/2} - 1 \) for all \( n \geq 2 \)

1. Let \( P(n) \) be “\( f_n \geq 2^{n/2} - 1 \)”. We prove that \( P(n) \) is true for all integers \( n \geq 2 \) by strong induction.

2. Base Cases: \( f_2 = f_1 + f_0 = 1 \) and \( 2^{2/2} - 1 = 2^0 = 1 \) so \( P(2) \) holds
   \( f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2} - 1 \) so \( P(3) \) holds

3. Inductive Hypothesis: Assume that for some arbitrary integer \( k \geq 3 \), \( P(j) \) is true for every integer \( j \) from 2 to \( k \).

4. Inductive Step: \( \text{Goal: Show } P(k+1); \text{ that is, } f_{k+1} \geq 2^{(k+1)/2} - 1 \)

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \\
  f_n &= f_{n-1} + f_{n-2} \quad \text{for all } n \geq 2
\end{align*}
\]
Bounding Fibonacci II: $f_n \geq 2^{n/2} - 1$ for all $n \geq 2$

1. Let $P(n)$ be “$f_n \geq 2^{n/2} - 1$.” We prove that $P(n)$ is true for all integers $n \geq 2$ by strong induction.

2. Base Cases: $f_2 = f_1 + f_0 = 1$ and $2^{2/2} - 1 = 2^0 - 1 = 0 = 1$ so $P(2)$ holds

   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2} - 1 = 2^{(k+1)/2} - 1$ so $P(3)$ holds

3. Inductive Hypothesis: Assume that for some arbitrary integer $k \geq 3$, $P(j)$ is true for every integer $j$ from 2 to $k$.

4. Inductive Step: Goal: Show $P(k+1)$; that is, $f_{k+1} \geq 2^{(k+1)/2} - 1$

   We have $f_{k+1} = f_k + f_{k-1}$ by definition since $k+1 \geq 2$

   $\geq 2^{k/2} - 1 + 2^{(k-1)/2} - 1$ by the IH since $k-1 \geq 2$

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

   so $P(k+1)$ is true.

5. Therefore by strong induction, $f_n \geq 2^{n/2} - 1$ for all integers $n \geq 2$.

   $f_0 = 0 \quad f_1 = 1$

   $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$
Running time of Euclid’s algorithm

Theorem: Suppose that Euclid’s Algorithm takes \( n \) steps for \( \gcd(a, b) \) with \( a \geq b > 0 \). Then, \( a \geq f_{n+1} \).
Running time of Euclid’s algorithm

**Theorem:** Suppose that Euclid’s Algorithm takes \( n \) steps for \( \gcd(a, b) \) with \( a \geq b > 0 \). Then, \( a \geq f_{n+1} \).

Why does this help us bound the running time of Euclid’s Algorithm?

We already proved that \( f_n \geq 2^{n/2} - 1 \) so \( f_{n+1} \geq 2^{(n-1)/2} \)

Therefore: if Euclid’s Algorithm takes \( n \) steps for \( \gcd(a, b) \) with \( a \geq b > 0 \) then \( a \geq 2^{(n-1)/2} \)

so \( (n - 1)/2 \leq \log_2 a \) or \( n \leq 1 + 2 \log_2 a \)
i.e., # of steps \( \leq 1 + \text{twice the # of bits in } a \).
Running time of Euclid’s algorithm

**Theorem:** Suppose that Euclid’s Algorithm takes \( n \) steps for \( \gcd(a, b) \) with \( a \geq b > 0 \). Then, \( a \geq f_{n+1} \).

An informal way to get the idea: Consider an \( n \) step \( \gcd \) calculation starting with \( r_{n+1} = a \) and \( r_n = b \):

\[
\begin{align*}
    r_{n+1} &= q_nr_n + r_{n-1} \\
    r_n &= q_{n-1}r_{n-1} + r_{n-2} \\
    & \quad \ldots \\
    r_3 &= q_2r_2 + r_1 \\
    r_2 &= q_1r_1 \\
    \text{For all } k \geq 2, \ r_{k-1} &= r_{k+1} \mod r_k
\end{align*}
\]

Now \( r_1 \geq 1 \) and each \( q_k \) must be \( \geq 1 \). If we replace all the \( q_k \)'s by 1 and replace \( r_1 \) by 1, we can only reduce the \( r_k \)'s. After that reduction, \( r_k = f_k \) for every \( k \).
Running time of Euclid’s algorithm

**Theorem:** Suppose that Euclid’s Algorithm takes \( n \) steps for \( \gcd(a, b) \) with \( a \geq b > 0 \). Then, \( a \geq f_{n+1} \).

We go by strong induction on \( n \).
Let \( P(n) \) be “\( \gcd(a,b) \) with \( a \geq b > 0 \) takes \( n \) steps \( \rightarrow a \geq f_{n+1} \)” for all \( n \geq 1 \).

**Base Case:** \( n=1 \) Suppose Euclid’s Algorithm with \( a \geq b > 0 \) takes 1 step.
By assumption, \( a \geq b \geq 1 = f_2 \) so \( P(1) \) holds.

**Induction Hypothesis:** Suppose that for some integer \( k \geq 1 \), \( P(j) \) is true for all integers \( j \) s.t. \( 1 \leq j \leq k \)

**Inductive Step:** We want to show: if \( \gcd(a,b) \) with \( a \geq b > 0 \) takes \( k+1 \) steps, then \( a \geq f_{k+2} \).
Running time of Euclid’s algorithm

**Induction Hypothesis:** Suppose that for some integer $k \geq 1$, $P(j)$ is true for all integers $j$ s.t. $1 \leq j \leq k$

**Inductive Step:** Goal: if \( \gcd(a, b) \) with $a \geq b > 0$ takes $k+1$ steps, then $a \geq f_{k+2}$.

Now if $k+1=2$, then Euclid’s algorithm on $a$ and $b$ can be written as

$$
\begin{align*}
    a &= q_2 b + r_1 \\
    b &= q_1 r_1 \\
    \text{and } r_1 &> 0.
\end{align*}
$$

Also, since $a \geq b > 0$, we must have $q_2 \geq 1$ and $b \geq 1$.

So $a = q_2 b + r_1 \geq b + r_1 \geq 1+1 = 2 = f_3 = f_{k+2}$ as required.
Running time of Euclid’s algorithm

Induction Hypothesis: Suppose that for some integer \( k \geq 1 \), \( P(j) \) is true for all integers \( j \) s.t. \( 1 \leq j \leq k \)

Inductive Step: Goal: if \( \gcd(a, b) \) with \( a \geq b > 0 \) takes \( k+1 \) steps, then \( a \geq f_{k+2} \).

Next suppose that \( k+1 \geq 3 \) so for the first 3 steps of Euclid’s algorithm on \( a \) and \( b \) we have

\[
\begin{align*}
    a &= q_{k+1} b + r_k \\
    b &= q_k r_k + r_{k-1} \\
    r_k &= q_{k-1} r_{k-1} + r_{k-2}
\end{align*}
\]

and there are \( k-2 \) more steps after this. Note that this means that \( \gcd(b, r_k) \) takes \( k \) steps and \( \gcd(r_k, r_{k-1}) \) takes \( k-1 \) steps.

So since \( k, k-1 \geq 1 \), by the IH we have \( b \geq f_{k+1} \) and \( r_k \geq f_k \).

Also, since \( a \geq b \), we must have \( q_{k+1} \geq 1 \).

So \( a = q_{k+1} b + r_k \geq b + r_k \geq f_{k+1} + f_k = f_{k+2} \) as required. \( \blacksquare \)
Last time: Recursive definitions of functions

• $0! = 1$; $(n + 1)! = (n + 1) \cdot n!$ for all $n \geq 0$.

• $F(0) = 0$; $F(n + 1) = F(n) + 1$ for all $n \geq 0$.

• $G(0) = 1$; $G(n + 1) = 2 \cdot G(n)$ for all $n \geq 0$.

• $H(0) = 1$; $H(n + 1) = 2^{H(n)}$ for all $n \geq 0$. 
Last time: Recursive definitions of functions

• Recursive functions allow general computation
  – saw examples not expressible with simple expressions

• So far, we have considered only simple data
  – inputs and outputs were just integers

• We need general data as well...
  – these will also be described recursively
  – will allow us to describe data of real programs
    e.g., strings, lists, trees, expressions, propositions, ...

• We’ll start simple: sets of numbers
Recursive Definitions of Sets (Data)

Natural numbers
Basis: \( 0 \in S \)
Recursive: If \( x \in S \), then \( x+1 \in S \)

Even numbers
Basis: \( 0 \in S \)
Recursive: If \( x \in S \), then \( x+2 \in S \)
Recursive Definition of Sets

Recursive definition of set $S$

- **Basis Step:** $0 \in S$
- **Recursive Step:** If $x \in S$, then $x + 2 \in S$
- **Exclusion Rule:** Every element in $S$ follows from the basis step and a finite number of recursive steps.

We need the exclusion rule because otherwise $S = \mathbb{N}$ would satisfy the other two parts. However, we won’t always write it down on these slides.
Recursive Definitions of Sets

Natural numbers
   Basis: \(0 \in S\)
   Recursive: If \(x \in S\), then \(x+1 \in S\)

Even numbers
   Basis: \(0 \in S\)
   Recursive: If \(x \in S\), then \(x+2 \in S\)

Powers of 3:
   Basis: \(1 \in S\)
   Recursive: If \(x \in S\), then \(3x \in S\).

Basis: \((0, 0) \in S, (1, 1) \in S\)
Recursive: If \((n-1, x) \in S\) and \((n, y) \in S\), then \((n+1, x + y) \in S\).
Recursive Definitions of Sets

Natural numbers
Basis: $0 \in S$
Recursive: If $x \in S$, then $x+1 \in S$

Even numbers
Basis: $0 \in S$
Recursive: If $x \in S$, then $x+2 \in S$

Powers of 3:
Basis: $1 \in S$
Recursive: If $x \in S$, then $3x \in S$.

Basis: $(0, 0) \in S$, $(1, 1) \in S$
Recursive: If $(n-1, x) \in S$ and $(n, y) \in S$, then $(n+1, x + y) \in S$. Fibonacci numbers
Last time: Recursive definitions of functions

• Before, we considered only simple data
  – inputs and outputs were just integers

• Proved facts about those functions with induction
  – $n! \leq n^n$
  – $f_n < 2^n$ and $f_n \geq 2^{n/2-1}$

• How do we prove facts about functions that work with more complex (recursively defined) data?
  – we need a more sophisticated form of induction
Structural Induction

How to prove ∀ x ∈ S, P(x) is true:

**Base Case:** Show that P(𝑢) is true for all specific elements 𝑢 of S mentioned in the Basis step

**Inductive Hypothesis:** Assume that P is true for some arbitrary values of each of the existing named elements mentioned in the Recursive step

**Inductive Step:** Prove that P(𝑤) holds for each of the new elements 𝑤 constructed in the Recursive step using the named elements mentioned in the Inductive Hypothesis

**Conclude** that ∀ x ∈ S, P(x)
Structural Induction vs. Ordinary Induction

Structural induction follows from ordinary induction:

Define $Q(n)$ to be “for all $x \in S$ that can be constructed in at most $n$ recursive steps, $P(x)$ is true.”

Ordinary induction is a special case of structural induction:

Recursive definition of $\mathbb{N}$

Basis: $0 \in \mathbb{N}$

Recursive step: If $k \in \mathbb{N}$ then $k + 1 \in \mathbb{N}$
Using Structural Induction

• Let $S$ be given by...
  – **Basis:** $6 \in S; \ 15 \in S$
  – **Recursive:** if $x, y \in S$ then $x + y \in S$.

**Claim:** Every element of $S$ is divisible by 3.
**Claim:** Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$3 \mid x$”. We prove that $P(x)$ is true for all $x \in S$ by structural induction.

   **Basis:** $6 \in S; \ 15 \in S$

   **Recursive:** if $x, y \in S$ then $x + y \in S$
Claim: Every element of $S$ is divisible by $3$.

1. Let $P(x)$ be “$3 \mid x$”. We prove that $P(x)$ is true for all $x \in S$ by structural induction.

2. Base Case: $3 \mid 6$ and $3 \mid 15$ so $P(6)$ and $P(15)$ are true

Basis: $6 \in S; \ 15 \in S$

Recursive: if $x, y \in S$ then $x + y \in S$
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$3 | x$”. We prove that $P(x)$ is true for all $x \in S$ by structural induction.

2. Base Case: $3 | 6$ and $3 | 15$ so $P(6)$ and $P(15)$ are true

3. Inductive Hypothesis: Suppose that $P(x)$ and $P(y)$ are true for some arbitrary $x, y \in S$

4. Inductive Step: Goal: Show $P(x+y)$

Basis: $6 \in S; 15 \in S$

Recursive: if $x, y \in S$ then $x + y \in S$
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$3 \mid x$”. We prove that $P(x)$ is true for all $x \in S$ by structural induction.

2. Base Case: $3 \mid 6$ and $3 \mid 15$ so $P(6)$ and $P(15)$ are true.

3. Inductive Hypothesis: Suppose that $P(x)$ and $P(y)$ are true for some arbitrary $x, y \in S$.

4. Inductive Step: **Goal: Show $P(x+y)$**

   Since $P(x)$ is true, $3 \mid x$ and so $x=3m$ for some integer $m$ and since $P(y)$ is true, $3 \mid y$ and so $y=3n$ for some integer $n$.

   Therefore $x+y=3m+3n=3(m+n)$ and thus $3 \mid (x+y)$.

   Hence $P(x+y)$ is true.

5. Therefore by induction $3 \mid x$ for all $x \in S$.

**Basis:** $6 \in S; \ 15 \in S$

**Recursive:** if $x, y \in S$ then $x + y \in S$
Using Structural Induction

• Let $T$ be given by...
  – Basis: $6 \in T$; $15 \in T$
  – Recursive: if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

• Two base cases and two recursive cases

Claim: Every element of $T$ is also in $S$. 
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$x \in S$”. We prove that $P(x)$ is true for all $x \in T$ by structural induction.

<table>
<thead>
<tr>
<th>Basis:</th>
<th>6 $\in S$; 15 $\in S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x, y \in S$, then $x + y \in S$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basis:</th>
<th>6 $\in T$; 15 $\in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>
**Claim:** Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$x \in S$”. We prove that $P(x)$ is true for all $x \in T$ by structural induction.

2. **Base Case:** $6 \in S$ and $15 \in S$ so $P(6)$ and $P(15)$ are true

<table>
<thead>
<tr>
<th>Basis:</th>
<th>$6 \in S$; $15 \in S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x, y \in S$, then $x + y \in S$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basis:</th>
<th>$6 \in T$; $15 \in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be "$x \in S$". We prove that $P(x)$ is true for all $x \in T$ by structural induction.
2. Base Case: $6 \in S$ and $15 \in S$ so $P(6)$ and $P(15)$ are true
3. Inductive Hypothesis: Suppose that $P(x)$ is true for some arbitrary $x \in T$

<table>
<thead>
<tr>
<th>Basis:</th>
<th>6 $\in S$; 15 $\in S$</th>
<th>Basis:</th>
<th>6 $\in T$; 15 $\in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x, y \in S$, then $x + y \in S$</td>
<td>Recursive:</td>
<td>if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$x \in S$”. We prove that $P(x)$ is true for all $x \in T$ by structural induction.
2. Base Case: $6 \in S$ and $15 \in S$ so $P(6)$ and $P(15)$ are true
3. Inductive Hypothesis: Suppose that $P(x)$ is true for some arbitrary $x \in T$
4. Inductive Step: Goal: Show $P(x+6)$ and $P(x+15)$

<table>
<thead>
<tr>
<th>Basis: $6 \in S$; $15 \in S$</th>
<th>Basis: $6 \in T$; $15 \in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive: if $x, y \in S$, then $x + y \in S$</td>
<td>Recursive: if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>
Claim: Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$x \in S$”. We prove that $P(x)$ is true for all $x \in T$ by structural induction.

2. Base Case: $6 \in S$ and $15 \in S$ so $P(6)$ and $P(15)$ are true.

3. Inductive Hypothesis: Suppose that $P(x)$ is true for some arbitrary $x \in T$.

4. Inductive Step: Goal: Show $P(x+6)$ and $P(x+15)$.
   Since $P(x)$ holds, we have $x \in S$. From the recursive step of $S$, we can see that $x + 6 \in S$, so $P(x+6)$ is true, and we can see that $x + 15 \in S$, so $P(x+15)$ is true.

<table>
<thead>
<tr>
<th>Basis: $6 \in S$; $15 \in S$</th>
<th>Basis: $6 \in T$; $15 \in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive: if $x, y \in S$, then $x + y \in S$</td>
<td>Recursive: if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>
**Claim:** Every element of $S$ is divisible by 3.

1. Let $P(x)$ be “$x \in S$”. We prove that $P(x)$ is true for all $x \in T$ by structural induction.
2. Base Case: $6 \in S$ and $15 \in S$ so $P(6)$ and $P(15)$ are true.
3. Inductive Hypothesis: Suppose that $P(x)$ is true for some arbitrary $x \in T$.
4. Inductive Step: **Goal:** Show $P(x+6)$ and $P(x+15)$.

   Since $P(x)$ holds, we have $x \in S$. From the recursive step of $S$, we can see that $x + 6 \in S$, so $P(x+6)$ is true, and we can see that $x + 15 \in S$, so $P(x+15)$ is true.
5. Therefore $P(x)$ for all $x \in T$ by induction.

<table>
<thead>
<tr>
<th>Basis:</th>
<th>6 $\in S$; 15 $\in S$</th>
<th>Basis:</th>
<th>6 $\in T$; 15 $\in T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive:</td>
<td>if $x, y \in S$, then $x + y \in S$</td>
<td>Recursive:</td>
<td>if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$</td>
</tr>
</tbody>
</table>