Structural Induction
Announcements

• HW4 due tonight at 11:59 pm. Turn it in with no late days to receive feedback by tomorrow for induction
Find the Bug
Find the Bug

Claim: For every odd integer $n$, $n^2 \equiv_4 1$.

Proof: Let $n$ be an arbitrary odd integer. Then by definition of odd, $n = 2k + 1$ for some integer $k$. Then consider $n^2 \equiv_4 1$. Plugging in $n = 2k + 1$ for $n^2$:

\[
\begin{align*}
    n^2 &\equiv_4 1 \\
    (2k + 1)^2 &\equiv_4 1 \\
    4k^2 + 4k + 1 &\equiv_4 1
\end{align*}
\]

Then by definition of congruence, $4 \mid 4k^2 + 4k + 1 - 1$, so $4 \mid 4k^2 + 4k$. Since this is true, the claim holds.
Find the Bug

Claim: For every odd integer $n$, $n^2 \equiv_4 1$.

Proof: Let $n$ be an arbitrary odd integer. Then by definition of odd, $n = 2k + 1$ for some integer $k$. Then consider $n^2 \equiv_4 1$. Plugging in $n = 2k + 1$ for $n^2$:

\[
\begin{align*}
    n^2 &\equiv_4 1 \\
    (2k + 1)^2 &\equiv_4 1 \\
    4k^2 + 4k + 1 &\equiv_4 1
\end{align*}
\]

Then by definition of congruence, $4 \mid 4k^2 + 4k + 1 - 1$, so $4 \mid 4k^2 + 4k$. Since this is true, the claim holds.
Fixed Proof

Claim: For every odd integer \( n \), \( n^2 \equiv_4 1 \).

Proof: Let \( n \) be an arbitrary odd integer. Then by definition of odd, \( n = 2k + 1 \) for some integer \( k \). Then consider \( n^2 \):

\[
\begin{align*}
n^2 &= (2k + 1)^2 \\
&= 4k^2 + 4k + 1 \\
&= 4(k^2 + k) + 1
\end{align*}
\]

Since \( k \) is an integer, \( k^2 + k \) is an integer. So by definition of divides, \( 4 \mid n^2 - 1 \). So by definition of congruence, \( n^2 \equiv_4 1 \). Since \( n \) was arbitrary, the claim holds.
Fixed Proof

Claim: For every odd integer $n$, $n^2 \equiv_4 1$.

Proof: Let $n$ be an arbitrary odd integer. Then by definition of odd, $n = 2k + 1$ for some integer $k$. Then consider $n^2$:

$$n^2 = (2k + 1)^2 = 4k^2 + 4k + 1$$
$$n^2 - 1 = 4k^2 + 4k$$
$$n^2 - 1 = 4(k^2 + k)$$

Since $k$ is an integer, $k^2 + k$ is an integer. So by definition of divides, $4 \mid n^2 - 1$. So by definition of congruence, $n^2 \equiv_4 1$. Since $n$ was arbitrary, the claim holds.
Backwards Reasoning

**Backwards reasoning** is the incorrect proof technique of *assuming* the goal is true, and then deriving some other true statement.

This reasoning can be used to incorrectly prove false statements.

**Claim:** For all integer $x$, if $x^2 = 25$, then $x = 5$.

**Backwards Proof:** Let $x$ be an arbitrary integer. Suppose $x^2 = 25$. Plugging in $x = 5$, we have $5^2 = 25$. Since this is true, the claim holds.

**False!** What if $x = -5$?
Find the 4 Bugs

Claim: For all integers \( n \geq 1 \), \( 1 + \cdots + n = \frac{n(n+1)}{2} \).

Proof: Let \( P(n) \) be "\( 1 + \cdots + n = \frac{n(n+1)}{2} \) for all integers \( n \geq 1 \)". We prove by induction.

Base Case: Plugging in \( n = 1 \), we have \( 1 = \frac{1(1+1)}{2} \). So \( 1 = \frac{2}{2} \). So \( 1 = 1 \). Since this is true, the base case holds.

IH: Suppose \( 1 + \cdots + k = \frac{k(k+1)}{2} \) for an arbitrary integer \( k \).

IS: We aim to show \( P(k+1) \). Observe that:

\[
1 + \cdots + (k+1) = 1 + \cdots + k + (k + 1) = \frac{k(k+1)}{2} + (k + 1) = \frac{k(k+1)+2(k+1)}{2} = \frac{(k+1)(k+2)}{2}
\]

So \( P(k + 1) \) holds.

Conclusion: Thus \( P(n) \) holds for all integers \( n \geq 1 \) by induction.
Claim: For all integers $n \geq 1$, $1 + \cdots + n = \frac{n(n+1)}{2}$.

Proof: Let $P(n)$ be "$1 + \cdots + n = \frac{n(n+1)}{2}$ for all integers $n \geq 1"$. We prove by induction.

Base Case: Plugging in $n = 1$, we have $1 = \frac{1(1+1)}{2}$. So $1 = \frac{2}{2}$. So $1 = 1$. Since this is true, the base case holds.

IH: Suppose $1 + \cdots + k = \frac{k(k+1)}{2}$ for an arbitrary integer $k$.

IS: We aim to show $P(k + 1)$. Observe that:

$$1 + \cdots + (k + 1) = 1 + \cdots + k + (k + 1) = \frac{k(k+1)}{2} + (k + 1) = \frac{k(k+1)+2(k+1)}{2} = \frac{(k+1)(k+2)}{2}.$$ 

So $P(k + 1)$ holds.

Conclusion: Thus $P(n)$ holds for all integers $n \geq 1$ by induction.
Avoiding Backwards Reasoning in the Base Case

Incorrect Technique: Backwards reasoning
Plugging in $n = 1$, we have $1 = \frac{1(1+1)}{2}$. So $1 = \frac{2}{2}$. So $1 = 1$. Since this is true, the base case holds.

Valid Technique 1: Separating LHS and RHS
The LHS evaluates to 1. The RHS evaluates to $\frac{1(1+1)}{2} = \frac{2}{2} = 1$. Since 1 = 1, the base case holds.

Valid Technique 2: Start from Left, convert to the Right
Observe that $1 = \frac{2}{2} = \frac{1\cdot2}{2} = \frac{1(1+1)}{2}$. So the base case holds.
Induction Big Picture

Weak and Strong Induction: Prove statements over the natural numbers.

“Prove that $P(n)$ holds for all natural numbers $n$."

Structural Induction: In CS, we deal with Strings, Lists, Trees, and other objects. Now we prove statements about these objects.

“Prove that $P(T)$ holds for all trees $T$.”

“Prove that $P(x)$ holds for all strings $x$. “
Recursively Defined Sets
Recursively Defined Sets

- In order to prove a fact about all trees or all lists, we need rigorous mathematical definitions for these sets.

- We will define these sets *recursively*. A *recursively defined set* has 3 components:
  - Basis Step
  - Recursive Step
  - Exclusion Rule
Recursively Defined Sets

For example, define a set $S$ as follows:

Basis Step: $0 \in S$

Recursive Step: If $x \in S$ then $x + 2 \in S$.

Exclusion Rule: Every element of $S$ follows from the basis step or a finite number of recursive steps.

What is $S$? The set of all non-negative even integers. $\{0, 2, 4,\ldots\}$

Why do we need the exclusion rule? To clarify that there aren’t any other elements in the set. In practice this isn’t usually written.
Recursively Defined Sets

Natural Numbers ($\mathbb{N}$)

Integers ($\mathbb{Z}$)

Integer coordinates in the line $y = x$
Recursively Defined Sets

Natural Numbers ($\mathbb{N}$)
Basis Step: $0 \in S$
Recursive Step: If $x \in S$ then $x + 1 \in S$.

Integers ($\mathbb{Z}$)
Basis Step: $0 \in S$
Recursive Step: If $x \in S$ then $x + 1 \in S$ and $x - 1 \in S$.

Integer coordinates in the line $y = x$
Basis Step: $(0,0) \in S$
Recursive Step: If $(x, y) \in S$ then $(x + 1, y + 1) \in S$ and $(x - 1, y - 1) \in S$. 
Recursively Defined Sets

Q1: Write a recursive definition for the set of positive even integers
Basis Step:
Recursive Step:

Q2: Write a recursive definition for the set of powers of 3 \( \{1,3,9,27, \ldots \} \)
Basis Step:
Recursive Step:
Recursively Defined Sets

Q1: Write a recursive definition for the set of positive even integers
Basis Step: $2 \in S$
Recursive Step: If $x \in S$ then $x + 2 \in S$

Q2: Write a recursive definition for the set of powers of 3 \{1,3,9,27, ... \}
Basis Step: $1 \in S$
Recursive Step: If $n \in S$, then $3n \in S$
Structural Induction
On Sets of Numbers
Claim about a Recursively Defined Set

Let $S$ be the set defined:

Basis Step: $6 \in S, 15 \in S$

Recursive Step: if $x, y \in S$ then $x + y \in S$.

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

How would we prove this?
Structural Induction Idea

To show $P(s)$ for all $s \in S$...

Base Case: Show $P(b)$ for all elements $b$ in the basis step.

Inductive Hypothesis: Assume $P()$ holds for arbitrary element(s) that we’ve already constructed.

Inductive Step: Prove that $P()$ holds for a new element constructed using the recursive step.

Basis: $6 \in S, 15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$. 
Structural Induction Idea

To show $P(s)$ for all $s \in S$...

- Here, $P(s)$ is "$3 \mid s$".

Base Case: Show $P(b)$ for all elements $b$ in the basis step.
- Show $P(6)$ and $P(15)$ hold.

Inductive Hypothesis: Assume $P()$ holds for arbitrary element(s) that we’ve already constructed.
- Assume $P(x)$ and $P(y)$ for arbitrary $x, y \in S$.

Inductive Step: Prove that $P()$ holds for a new element constructed using the recursive step.
- Show $P(x + y)$ holds.
Structural Induction

1. Let $P(s)$ be “$s$ is divisible by 3”. We show $P(s)$ holds for all $s \in S$ by structural induction.

2. Base Case(s): $6 = 2 \cdot 3$ so $3|6$, and $P(6)$ holds. $15 = 5 \cdot 3$, so $3|15$ and $P(15)$ holds.

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

4. Inductive Step: \[ \text{Goal: } P(x + y) \text{ holds} \]
   By IH $3 | x$ and $3 | y$. So by definition of divides, $x = 3n$ and $y = 3m$ for integers $m, n$.

   Adding the equations: $x + y = 3(n + m)$. Since $n, m$ are integers $n + m$ is an integer. Thus by definition of divides, $3 | (x + y)$. So $P(x + y)$ holds.

5. Conclusion: Thus $P(s)$ for all $s \in S$ by structural induction.
How does this work?

**S**

6  15

12  21  30

18  27

24...

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

Recursive: if $x, y \in S$ then $x + y \in S$.

**We proved:**

Base Case: $P(6)$ and $P(15)$

$IH \rightarrow IS$: If $P(x)$ and $P(y)$, then $P(x+y)$
Weak Induction is a special case of Structural

\[ \begin{align*} 
\mathbb{N} & \quad 1 \quad 2 \\
0 & \quad 1 \\
3 & \quad 4 \quad 5 \\
6 & \quad 7 \\
8 & \ldots 
\end{align*} \]

- **Basis:** \( 0 \in \mathbb{N} \)
- **Recursive:** if \( k \in \mathbb{N} \) then \( k + 1 \in \mathbb{N} \).

**We proved:**
- **Base Case:** \( P(0) \)
- **IH \rightarrow IS:** If \( P(k) \), then \( P(k+1) \)
Wait a minute! Why can we do this?

Think of each element of $S$ as requiring $k$ “applications of a rule” to get in

$P(\text{base cases})$ is true

$P(\text{base cases}) \rightarrow P(\text{one application}) \circ P(\text{one application})$

$P(\text{one application}) \rightarrow P(\text{two applications}) \circ P(\text{two applications})$

...

It’s the same principle as regular induction. You’re just inducting on “how many steps did we need to get this element?”

You’re still only assuming the IH about a domino you’ve knocked over.
Wait a minute! Why can we do this?

Imagine building $S$ “step-by-step”

\[
S_0 = \{6, 15\} \\
S_1 = \{12, 21, 30\} \\
S_2 = \{18, 24, 27, 36, 42, 45, 60\}
\]

IS can always of the form “suppose $P(x) \forall x \in (S_0 \cup \cdots \cup S_k)$” and show $P(y)$ for some $y \in S_{k+1}$

We use the structural induction phrasing assuming our reader knows how induction works and so don’t phrase it explicitly in this form.
1. Define $P()$. Claim that $P(s)$ holds for all $s \in S$. State your proof is by structural induction.

2. Base Case: Show $P(b_1), ..., P(b_n)$ holds for each basis step $b_1, ..., b_n$ in $S$.

3. Inductive Hypothesis: Suppose $P(x_1), ..., P(x_m)$ for all values listed in the recursive rules.

4. Inductive Step: Show $P()$ holds for the “new element” given by the recursive step. You will need a separate step for every rule.

5. Conclusion: Conclude that $P(s)$ holds for all $s \in S$ by structural induction.
Structural Induction

On Strings
String Terminology

\(\Sigma\) is the **alphabet**, i.e. the set of all letters you can use in strings.

For example: \(\Sigma = \{0,1\}\) or \(\Sigma = \{a, b, c, \ldots, z, _\}\)

\(\Sigma^*\) is the set of **all strings** you can build from the letters in the alphabet.

For example: If \(\Sigma = \{0,1\}\) then \(01001 \in \Sigma^*\). If \(\Sigma = \{a, b, c, \ldots, z, _\}\), then \(i\_love\_induction \in \Sigma^*\)

- \(\varepsilon\) is the **empty string**

  Analogous to "" in Java
Recursive definition of Strings

The set of all strings $\Sigma^*$ can be defined recursively (using $\Sigma$, $\varepsilon$):

Basis Step: $\varepsilon \in \Sigma^*$

Recursive Step: If $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

$wa$ here means the string $w$ with the character $a$ appended on to it.

$\Sigma$ is the alphabet
$\Sigma^*$ is the set of all strings
$\varepsilon$ is the empty string
Functions on Strings

To prove interesting facts about strings, we need functions on strings.

Length:
\[ \text{len}(\varepsilon) = 0 \]
\[ \text{len}(wa) = \text{len}(w) + 1 \quad \text{for } w \in \Sigma^*, a \in \Sigma \]

Reversal:
\[ \varepsilon^R = \varepsilon \]
\[ (wa)^R = aw^R \quad \text{for } w \in \Sigma^*, a \in \Sigma \]
Claim about Strings

Claim: For any string $s \in \Sigma^*$, $\text{len}(s^R) = \text{len}(s)$
Proof

1. Let $P(s)$ be induction.

2. Base Case(s):

3. Inductive Hypothesis:

4. Inductive Step:

5. Conclusion:

We prove $P(s)$ for all strings $s$ by structural
**Proof**

1. Let $P(s)$ be $\text{len}(s^R) = \text{len}(s)$. We prove $P(s)$ for all strings $s$ by structural induction.

2. Base Case(s):

3. Inductive Hypothesis:

4. Inductive Step:

5. Conclusion:
Proof

1. Let $P(s)$ be $\text{len}(s^R) = \text{len}(s)$. We prove $P(s)$ for all strings $s$ by structural induction.

2. Base Case(s): ($s = \varepsilon$). LHS: Since $\varepsilon^R = \varepsilon$, $\text{len}(\varepsilon^R) = \text{len}(\varepsilon) = 0$. RHS: $\text{len}(\varepsilon) = 0$. Since $0 = 0$, the base case holds.

3. Inductive Hypothesis:

4. Inductive Step:

5. Conclusion:
1. Let $P(s)$ be $\text{len}(s^R) = \text{len}(s)$. We prove $P(s)$ for all strings $s$ by structural induction.

2. Base Case(s): ($s = \varepsilon$). LHS: Since $\varepsilon^R = \varepsilon$, $\text{len}(\varepsilon^R) = \text{len}(\varepsilon) = 0$. RHS: $\text{len}(\varepsilon) = 0$. Since $0 = 0$, the base case holds.

3. Inductive Hypothesis: Suppose $P(w)$ for some arbitrary string $w$. Then $\text{len}(w^R) = \text{len}(w)$

4. Inductive Step: $\text{Goal: } \text{len}((wa)^R) = \text{len}(wa)$

5. Conclusion:
Proof

1. Let $P(s)$ be $\text{len}(s^R) = \text{len}(s)$. We prove $P(s)$ for all strings $s$ by structural induction.

2. Base Case(s): $(s = \varepsilon)$. LHS: Since $\varepsilon^R = \varepsilon$, $\text{len}(\varepsilon^R) = \text{len}(\varepsilon) = 0$. RHS: $\text{len}(\varepsilon) = 0$. Since $0 = 0$, the base case holds.

3. Inductive Hypothesis: Suppose $P(w)$ for some arbitrary string $w$. Then $\text{len}(w^R) = \text{len}(w)$

4. Inductive Step: \( \text{Goal: len}((wa)^R) = \text{len}(wa) \)
   
   Let $a$ be an arbitrary character. Observe:

   $\text{len}((wa)^R) = \text{len}(aw^R)$
   
   By definition of reverse

5. Conclusion:
1. Let \( P(s) \) be \( \text{len}(s^R) = \text{len}(s) \). We prove \( P(s) \) for all strings \( s \) by structural induction.

2. Base Case(s): \((s = \varepsilon)\). LHS: Since \( \varepsilon^R = \varepsilon \), \( \text{len}(\varepsilon^R) = \text{len}(\varepsilon) = 0 \). RHS: \( \text{len}(\varepsilon) = 0 \). Since \( 0 = 0 \), the base case holds.

3. Inductive Hypothesis: Suppose \( P(w) \) for some arbitrary string \( w \). Then \( \text{len}(w^R) = \text{len}(w) \)

4. Inductive Step: \textbf{Goal: } \text{len}((wa)^R) = \text{len}(wa) \)

   Let \( a \) be an arbitrary character. Observe:

   \[
   \begin{align*}
   \text{len}((wa)^R) &= \text{len}(aw^R) & \text{By definition of reverse} \\
   &= \text{len}(w^R) + 1 & \text{By definition of length} \\
   &= \text{len}(w) + 1 & \text{By IH} \\
   &= \text{len}(wa) & \text{By definition of length}
   \end{align*}
   \]

5. Conclusion: Thus \( P(s) \) holds for all strings \( s \) by structural induction.
Trees!
More Structural Sets

Binary Trees are another common source of structural induction.

Basis: A single node is a rooted binary tree.

Recursive Step: If \( T_1 \) and \( T_2 \) are rooted binary trees with roots \( r_1 \) and \( r_2 \), then a tree rooted at a new node, with children \( r_1, r_2 \) is a binary tree.

\[
\begin{array}{c}
\text{Root} \\
T_1 \\
T_2
\end{array}
\]
Functions on Binary Trees

size(•) = 1

size( ) = size($T_1$) + size($T_2$) + 1

height( ) = 0

height( ) = 1 + \max(\text{height}(T_1), \text{height}(T_2))
Binary Trees

Basis: A single node is a rooted binary tree.

Recursive Step: If $T_1$ and $T_2$ are rooted binary trees with roots $r_1$ and $r_2$, then a tree rooted at a new node, with children $r_1, r_2$ is a binary tree.

- $size(\bullet) = 1$
- $size(T) = size(T_1) + size(T_2) + 1$
- $height(\bullet) = 0$
- $height(T) = 1 + \max(height(T_1), height(T_2))$
Claim

We want to show that trees of a certain height can’t have too many nodes. Specifically our claim is this:

For all trees $T$, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$

Take a moment to absorb this formula, then we’ll do induction!
Structural Induction on Binary Trees

Let $P(T)$ be true for all binary trees $T$ by structural induction.

Base Case:

Inductive Hypothesis:
Structural Induction on Binary Trees

Let $P(T)$ be “$\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$“. We show $P(T)$ for all binary trees $T$ by structural induction.

Base Case: Let $T = \bullet$. $\text{size}(T) = 1$ and $\text{height}(T) = 0$, so $\text{size}(T) = 1 \leq 2 - 1 = 2^{0+1} - 1 = 2^{\text{height}(T)+1} - 1$.

Inductive Hypothesis:
Structural Induction on Binary Trees

Let $P(T)$ be "$\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$". We show $P(T)$ for all binary trees $T$ by structural induction.

Base Case: Let $T = \bullet$. $\text{size}(T)=1$ and $\text{height}(T) = 0$, so $\text{size}(T)=1 \leq 2 - 1 = 2^{0+1} - 1 = 2^{\text{height}(T)+1} - 1$.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ hold for arbitrary trees $L, R$. Let $T$ be the tree

Inductive step: Figure out, (1) what we must show (2) a formula for height and a formula for size of $T$. 

![Tree Diagram]
Let $P(T)$ be \("size(T) \leq 2^{\text{height}(T)} + 1 - 1\". We show $P(T)$ for all binary trees $T$ by structural induction.

Let $T$ be a binary tree with children $L$ and $R$.

- \(\text{height}(T) = 1 + \max\{\text{height}(L), \text{height}(R)\}\)
- \(\text{size}(T) = 1 + \text{size}(L) + \text{size}(R)\)

So $P(T)$ holds, and we have $P(T)$ for all binary trees $T$ by the principle of induction.
Let \( P(T) \) be "size(\( T \)) \leq 2^{\text{height}(\( T \)) + 1} - 1". We show \( P(T) \) for all binary trees \( T \) by structural induction.

\[
T = \begin{array}{c}
\text{L} \\
\text{R}
\end{array}
\]

height(\( T \)) = 1 + \max\{\text{height}(L), \text{height}(R)\}

size(\( T \)) = 1 + \text{size}(L) + \text{size}(R)

size(\( T \)) = 1 + \text{size}(L) + \text{size}(R) \leq 1 + 2^{\text{height}(L) + 1} - 1 + 2^{\text{height}(R) + 1} - 1 \quad \text{(by IH)}

\leq 2^{\text{height}(L) + 1} + 2^{\text{height}(R) + 1} - 1 \quad \text{(cancel 1's)}

\leq 2^{\text{height}(T)} + 2^{\text{height}(T)} - 1 = 2^{\text{height}(T) + 1} - 1 \quad \text{\( T \) taller than subtrees)

So \( P(T) \) holds, and we have \( P(T) \) for all binary trees \( T \) by the principle of induction.