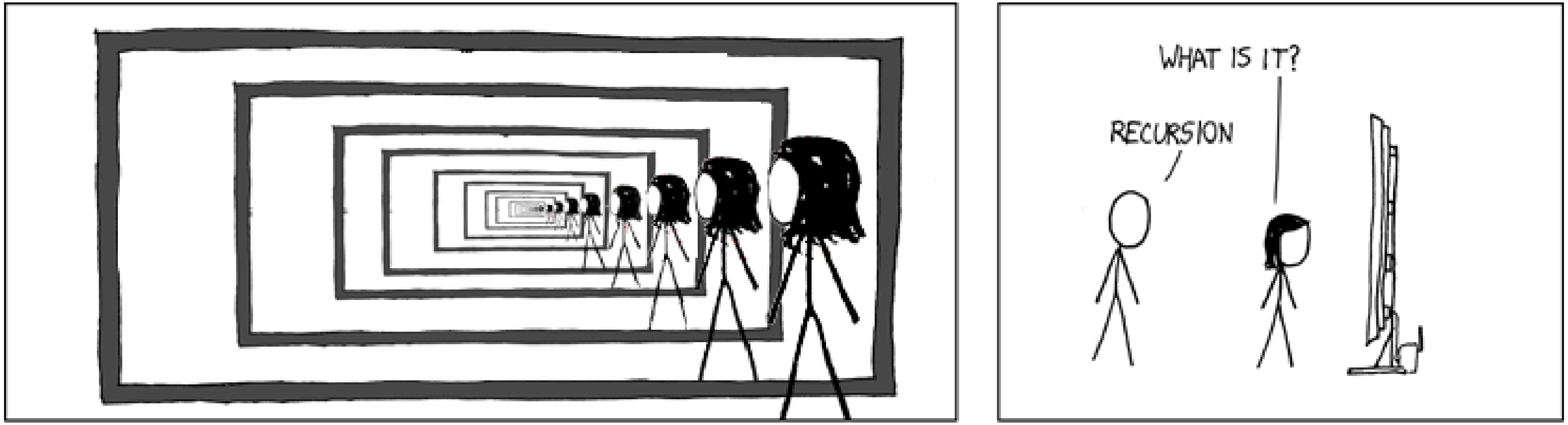


Structural Induction

CSE 311: Foundations of Computing I
Lecture 17

Announcements

- Congrats on completing the Midterm! Grades will be released later this week.
- HW5 grades published tonight for those who used late days. Solutions will be distributed in class on Wednesday.
- HW6 will be released this Wednesday.

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,…}

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 ($\mathbb{N}$)

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: What is this set? Th set $\{6, 12, 15, 18, 21, 24\}$

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

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

Q2: Write a recursive definition for the set of powers of 3 $\{1,3,9,27, \dots\}$

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

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

To show $\mathrm{P}(s)$ for all $s \in S$...
Here, $\mathrm{P}(s)$ is "$3 \mid s$".

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

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

Inductive Step: Prove that $\mathrm{P}()$ holds for a new element constructed using the recursive step.
Show $\mathrm{P}(x + y)$ holds.

Structural Induction

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

1. Let $\mathrm{P}(s)$ 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 $\mathrm{P}(6)$ holds. $15 = 5 \cdot 3$, so $3|15$ and $\mathrm{P}(15)$ holds.

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

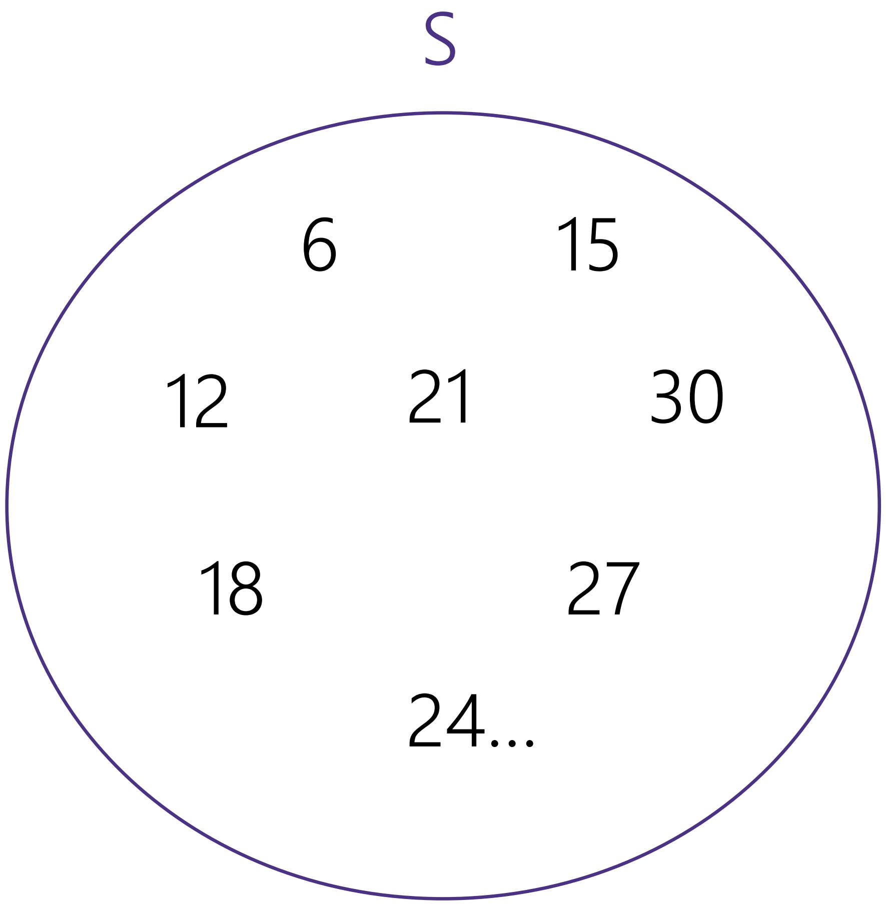
4. Inductive Step: **Goal: $\mathrm{P}(x + y)$ holds**

By IH $3 \mid x$ and $3 \mid y$. 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, $3 \mid (x + y)$. So $\mathrm{P}(x + y)$ holds.

5. Conclusion: Thus $\mathrm{P}(s)$ for all $s \in S$ by structural induction.

How does this work?



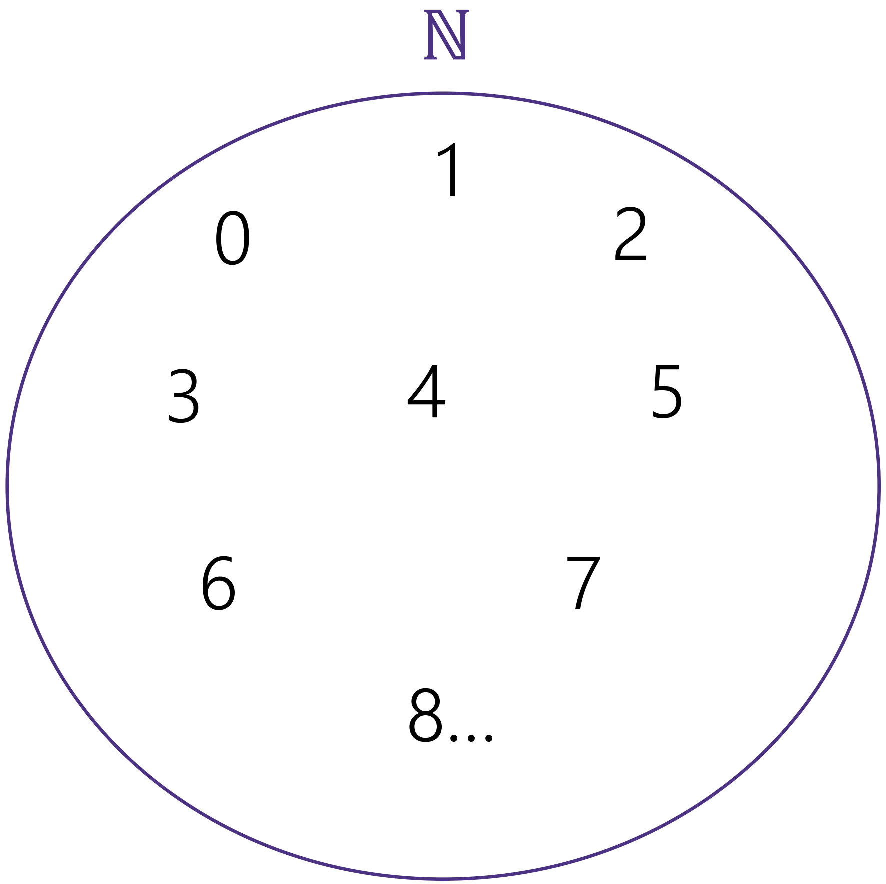
Basis: $6 \in S, 15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$.

We proved:

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

IH → IS: If P(x) and P(y), then P(x+y)

Weak Induction is a special case of Structural



Basis: $\mathbf{0} \in \mathbb{N}$
Recursive: if $k \in \mathbb{N}$ then $k + 1 \in \mathbb{N}$.

We proved:

Base Case: P(0)

IH → IS: If P(k), then P(k+1)

Structural Induction Template

1. Define $\mathrm{P}()$. Claim that $\mathrm{P}(s)$ holds for all $s \in S$. State your proof is by structural induction.

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

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

4. Inductive Step: Show $\mathrm{P}()$ holds for the "new element" given by the recursive step. **You will need a separate step for every rule.**

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

Structural Induction

On Strings

String Terminology

Σ is the alphabet, i.e. the set of all letters you can use in strings.

For example: $\Sigma = \{0,1\}$ or $\Sigma = \{a, b, c, \dots, z, \_\}$

Σ^* 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 $\Sigma = \{a, b, c, \dots, z, \_\}$, then $\text{i\_love\_induction} \in \Sigma^*$

ε is the empty string

Analogous to “” in Java

Recursive definition of Strings

Σ is the alphabet
Σ^* is the set of all strings
ε is the empty string

The set of all strings Σ^* can be defined recursively (using Σ, ε):

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

Functions on Strings

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

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

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

Length:

$\text{len}(\varepsilon) = 0$

$\text{len}(wa) = \text{len}(w) + 1$ for $w \in \Sigma^*, a \in \Sigma$

Reversal:

$\varepsilon^R = \varepsilon$

$(wa)^R = aw^R$ for $w \in \Sigma^*$, $a \in \Sigma$

Claim about Strings

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

Proof

$\text{len}(\varepsilon) = 0$
$\text{len}(wa) = \text{len}(w) + 1$

$\varepsilon^R = \varepsilon$
$(wa)^R = aw^R$

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

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: **Goal:** $\text{len}((wa)^R) = \text{len}(wa)$

Let a be an arbitrary character. Observe:

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

5. Conclusion: Thus $\text{P}(s)$ holds for all strings s by structural induction.

Structural Induction

On Lists and Trees

Recursive Definition for Lists of Integers

Basis Step: $[] \in \mathbf{List}$

Recursive Step: If $\mathbf{L} \in \mathbf{List}$ and $a \in \mathbb{Z}$ then $a :: \mathbf{L} \in \mathbf{List}$

Recursive Definition for Binary Trees of Integers

Basis Step: $\text{null} \in \text{Tree}$

Recursive Step: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$