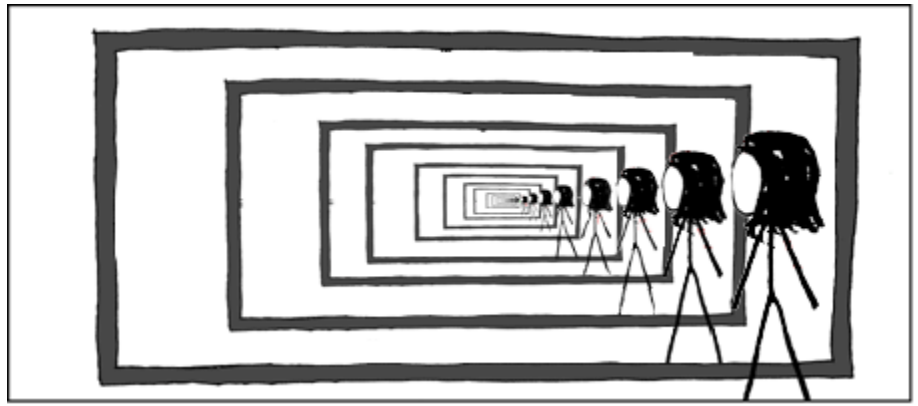
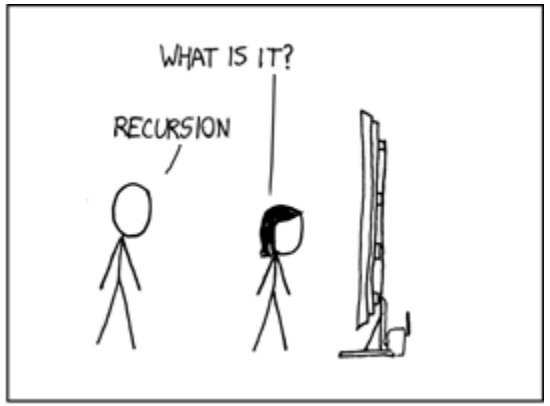
Warm up:

What's the first sentence of the base case and IS for a proof of:

$P(n)$: "Every set of integers of size $n$ has a largest element"






# Structural Induction

CSE 311 Autumn 23
Lecture 18

# Announcements

Don't forget HW5 is due this **Wednesday** at 10 PM.

No new HW on Wednesday; we want you to study for the midterm!

Friday is a university holiday! (Observance of Veteran's Day)

No lecture on Friday

Office Hours are more limited/on zoom (see the calendar)

We'll have a few midterm review opportunities in class

This Thursday's section will be mostly midterm review.

The day of the midterm (Wed Nov 15), TAs will come to lecture to answer questions, but no "official" lecture (and no concept check released that day).

# Induction Big Picture

So far: We used induction to prove a statement over the natural numbers.

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

Next goal: In CS, we deal with Strings, Lists, Trees, and other recursively defined sets. Would like to prove statements over these sets.

> "Prove that P(T) holds for all trees T."
>
> "Prove that P(x) holds for all strings x."

# Recursive Definition of Sets

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 $S$ is in $S$ from the basis step (alone) or a finite number of recursive steps starting from a basis step.

What is $S$?

# Recursive Definitions of Sets

We'll always list the Basis and Recursive parts of the definition.

Starting…now…we're going to be lazy and skip writing the "exclusion" rule. It's still part of the definition.

# Recursive Definitions of Sets

All Natural Numbers

**Basis Step**: $0 \in S$

**Recursive Step**: If $x \in S$ then $x + 1 \in S$.

All Integers

**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 $(x - 1, y - 1) \in S$.

# Recursive Definitions of Sets

Q1: What is this set?

**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 $\{1,3,9,27, \ldots\}$

**Basis Step**:

**Recursive Step**:

# Structural Induction

Every element is built up recursively…

So to show $P(s)$ for all $s \in S$…

Show $P(b)$ for all base case elements $b$.

Show for an arbitrary element not in the base case, if $P()$ holds for every named element in the recursive rule, then $P()$ holds for the new element (each recursive rule will be a case of this proof).

# Structural Induction Example

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

Show that every element of $S$ is divisible by $3$.

# Structural Induction

Let $P(x)$ be "$x$ is divisible by 3"

We show $P(x)$ holds for all $x \in S$ by structural induction.

Base Cases:
Inductive Hypothesis:

Inductive Step:

We conclude $P(x) \forall x \in S$ by the principle of induction.

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

# Structural Induction

Let $P(x)$ be "$x$ is divisible by 3"

We show $P(x)$ holds for all $x \in S$ by structural induction.

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

Let $s$ be an arbitrary element of $S$ not covered by the base cases. By the exclusion rule, $s = x + y$ for $x, y \in S$.

Inductive Hypothesis: Suppose $P(x)$ and $P(y)$.

Inductive Step:

We conclude $P(x) \forall x \in S$ by the principle of induction.

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

# Structural Induction

Let $P(x)$ be $x$ is divisible by 3

We show $P(x)$ holds for all $x \in S$ by structural induction.

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

Let $s$ be an arbitrary element of $S$ not covered by the base cases. By the exclusion rule, $s = x + y$ for $x, y \in S$.

Inductive Hypothesis: Suppose $P(x)$ and $P(y)$.

Inductive Step: By IH $3|\mathrm{x}$ and $3|\mathrm{y}$. So $x = 3n$ and $y = 3m$ for integers $m, n$.

Adding the equations, $x + y = 3(n + m)$. Since $n, m$ are integers, we have $3|(x + y)$ by definition of divides. This gives $P(x + y)$.

We conclude $P(x) \forall x \in \mathrm{S}$ by the principle of induction.

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

# Structural Induction Template

1. Define $P()$ Show that $P(x)$ holds for all $x \in S$. State your proof is by structural induction.

2. Base Case: Show $P(x)$
[Do that for every base cases $x$ in $S$.]

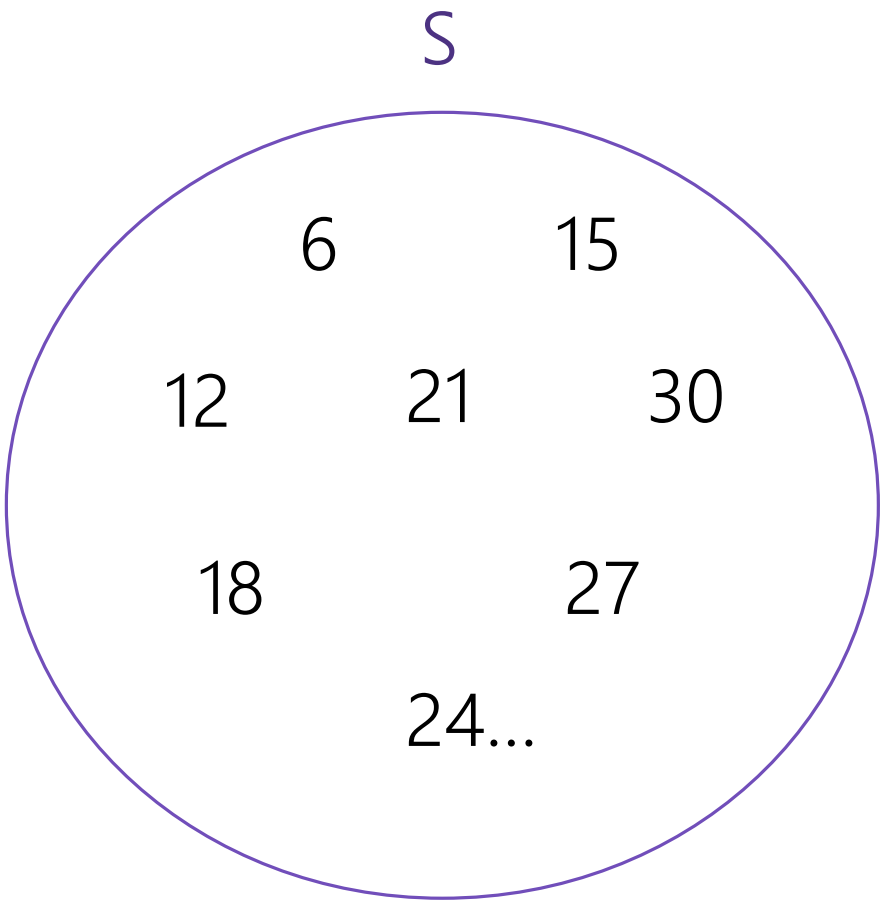
Let $y$ be an arbitrary element of $S$ not covered by the base cases. By the exclusion rule, $y =$<recursive rules>

3. Inductive Hypothesis: Suppose $P(x)$
[Do that for every $x$ listed as in $S$ in the recursive rules.]

4. Inductive Step: Show $P()$ holds for $y$.
[You will need a separate case/step for every recursive rule.]

5. Therefore $P(x)$ holds for all $x \in S$ by the principle of induction.

# Wait a minute! Why can we do this?



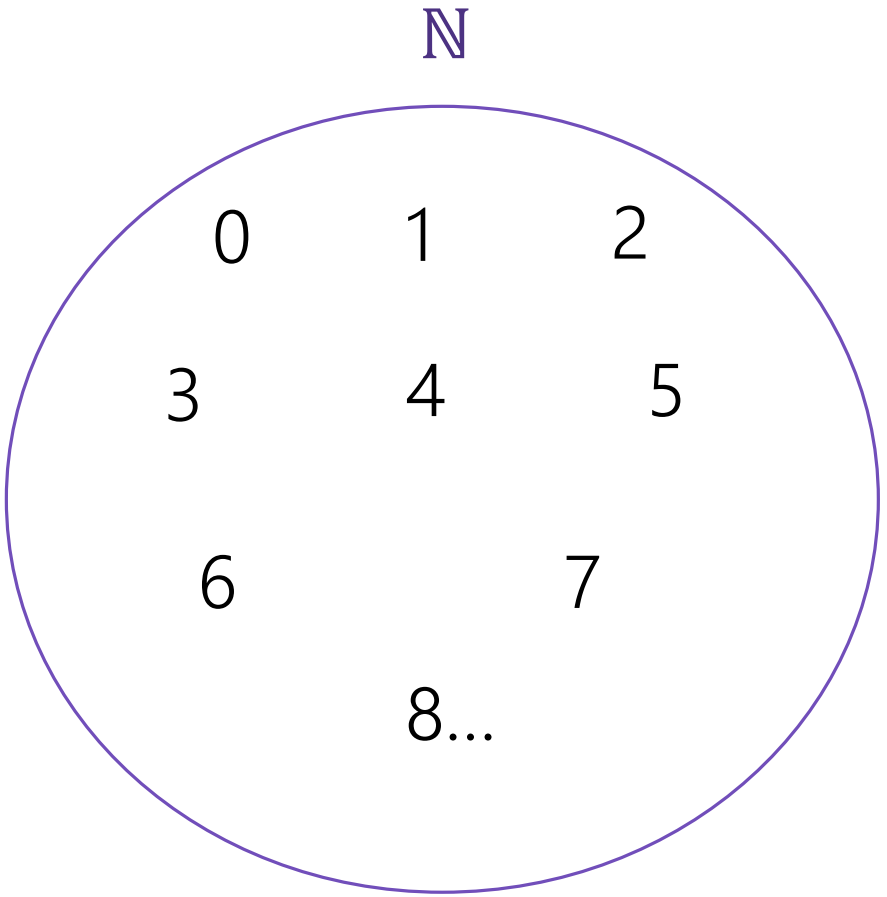

Basis: $6 \in S, 15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$.

**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: $0 \in \mathbb{N}$
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)

# Wait a minute! Why can we do this?

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

$P(base\ cases)$ is true

$P(base\ cases) \rightarrow P(one\ application)$ so $P(one\ application)$

$P(one\ application) \rightarrow P(two\ applications)$ so $P(two\ applications)$

…

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.

# Strings

Why these recursive definitions?

They're the basis for regular expressions, which we'll introduce next week. Answer questions like "how do you search for anything that looks like an email address"

First, we need to talk about strings.

$\Sigma$ will be an **alphabet** the set of all the letters you can use in words.

$\Sigma^*$ is the set of all **words** all the strings you can build off of the letters.

# Strings

$\varepsilon$ is "the empty string"

The string with $\mathbf{0}$ characters – `""` in Java (not `null`!)

$\Sigma^*$:

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

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

$wa$ means the string of $w$ with the character $a$ appended.

You'll also see $w \cdot a$ (a $\cdot$ to mean "concatenate" i.e. + in Java)

# Functions on Strings

Since strings are defined recursively, most functions on strings are as well.

Length:

len($\varepsilon$)=0;

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

Reversal:

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

Concatenation

$x \cdot \varepsilon = x$ for all $x \in \Sigma^*$;
$x \cdot (wa) = (x \cdot w)a$ for $w \in \Sigma^*, a \in \Sigma$

Number of $c$'s in a string

$\#_c(\varepsilon) = 0$
$\#_c(wc) = \#_c(w) + 1$ for $w \in \Sigma^*$;
$\#_c(wa) = \#_c(w)$ for $w \in \Sigma^*, a \in \Sigma \setminus \{c\}$.

# Functions on Strings

Since strings are defined recursively, most functions on strings are as well.

Length:

len($\varepsilon$)=0;

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

Reversal:

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

Concatenation

$x \cdot \varepsilon = x$ for all $x \in \Sigma^*$;
$x \cdot (wa) = (x \cdot w)a$ for $w \in \Sigma^*, a \in \Sigma$

Number of $c$'s in a string

$\#_c(\varepsilon) = 0$
$\#_c(wc) = \#_c(w) + 1$ for $w \in \Sigma^*$;
$\#_c(wa) = \#_c(w)$ for $w \in \Sigma^*, a \in \Sigma \setminus \{c\}$.

# Claim for all $x, y \in \Sigma^*$ len(x·y)=len(x) + len(y).

Let $P(y)$ be "for all $x \in \Sigma^*$ len(x·y)=len(x) + len(y)."

Notice the strangeness of this $P()$ there is a "for all $x$" inside the definition of $P(y)$.

That means we'll have to introduce an arbitrary $x$ as part of the base case and the inductive step!

# Claim for all $x, y \in \Sigma^*$ len(x·y)=len(x) + len(y).

Define Let $P(y)$ be "len(x·y)=len(x) + len(y) for all $x \in \Sigma^*$."

We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

Base Case:

Inductive Hypothesis:

Inductive Step:

len($\varepsilon$)=0;
len($wa$)=len($w$)+1 for $w \in \Sigma^*$, $a \in \Sigma$
$\Sigma^*$:Basis: $\varepsilon \in \Sigma^*$.
Recursive: If $w \in \Sigma^*$ and $a \in \Sigma$ then $wa \in \Sigma^*$

# Claim for all $x, y \in \Sigma^*$ len(x·y)=len(x) + len(y).

Define Let $P(y)$ be "len(x·y)=len(x) + len(y) for all $x \in \Sigma^*$."

We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

Base Case: Let $x$ be an arbitrary string, len($x \cdot \varepsilon$)=len(x)
=len(x)+0=len(x)+len($\varepsilon$)

Let $y$ be an arbitrary string not covered by the base case. By the exclusion rule, $y = wa$ for a string $w$ and character $a$.

Inductive Hypothesis: Suppose $P(w)$

Inductive Step: Let $x$ be an arbitrary string.

Therefore, len(xwa)=len(x) + len(wa)

len($\varepsilon$)=0;
len($wa$)=len($w$)+1 for $w \in \Sigma^*$, $a \in \Sigma$
$\Sigma^*$:Basis: $\varepsilon \in \Sigma^*$.
Recursive: If $w \in \Sigma^*$ and $a \in \Sigma$ then $wa \in \Sigma^*$

# Claim for all $x, y \in \Sigma^*$ len(x·y)=len(x) + len(y).

Define Let $P(y)$ be "len(x·y)=len(x) + len(y) for all $x \in \Sigma^*$. "

We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

Base Case: Let $x$ be an arbitrary string, len($x \cdot \epsilon$)=len(x)
=len(x)+0=len(x)+len($\varepsilon$)

Let $y$ be an arbitrary string not covered by the base case. By the exclusion rule, $y = wa$ for a string $w$ and character $a$.

Inductive Hypothesis: Suppose $P(w)$

Inductive Step: Let $x$ be an arbitrary string.

len(xy)=len(xwa) =len(xw)+1 (by definition of len)

=len(x) + len(w) + 1 (by IH)

=len(x) + len(wa) (by definition of len)

Therefore, len(xy)=len(x) + len(y), as required.

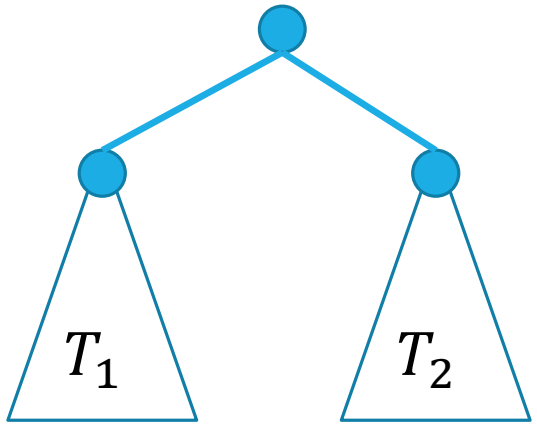
We conclude that $P(y)$ holds for all string $y$ by the principle of induction. Unwrapping the definition of $y$, we get $\forall x \forall y \in \Sigma^*$ len(xy)=len(x)+len(y), as required.

# More Structural Sets

Binary Trees are another common source of structural induction.

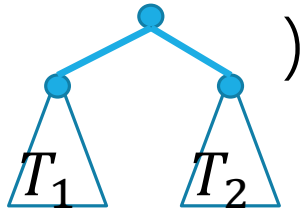
Basis: A single node is a rooted binary tree.

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.



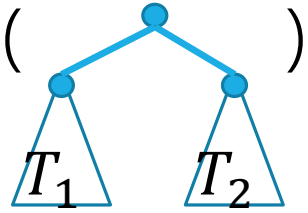
# Functions on Binary Trees

size( ● )=1

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



height(●) = 0

height(  ) = 1+$\mathbf{max}$(height($T_1$),height($T_2$))

# Structural Induction on Binary Trees

For every rooted binary tree $T$, size($T$) $\leq 2^{height(T)+1} - 1$

We'll show this next time.