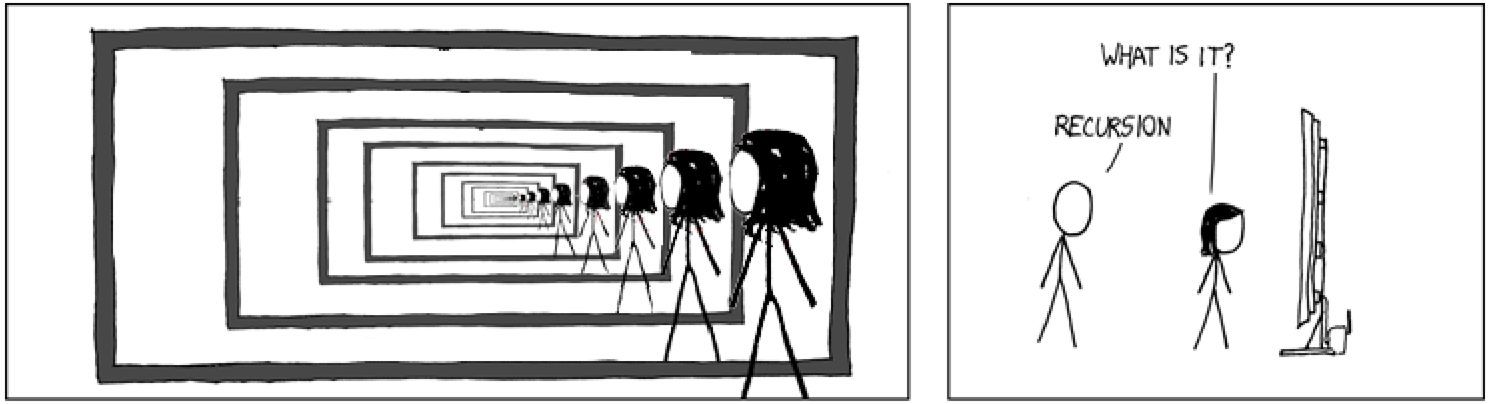


Structural Induction

CSE 311 Winter 23
Lecture 17

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."

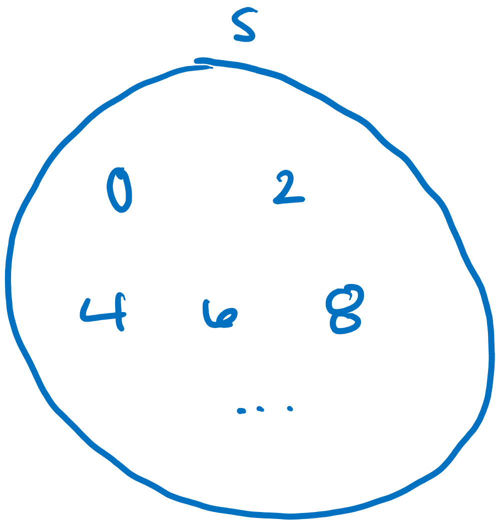
E.g. $P(n)$ is "$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$"

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 Definitions 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 follows from the basis step or a finite number of recursive steps.

↳ Don't write in practice

Q1: What is S? All nonnegative even integers $\{0, 2, 4, 6 \ldots\}$

Q2: Why do we need the exclusion rule?

Recursive Definitions of Sets

All Natural Numbers

Basis Step: $0 \in S$

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

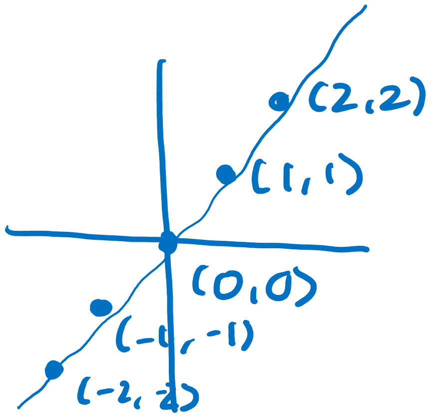
All Integers

Basis Step: $0 \in S$

two recursive steps

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

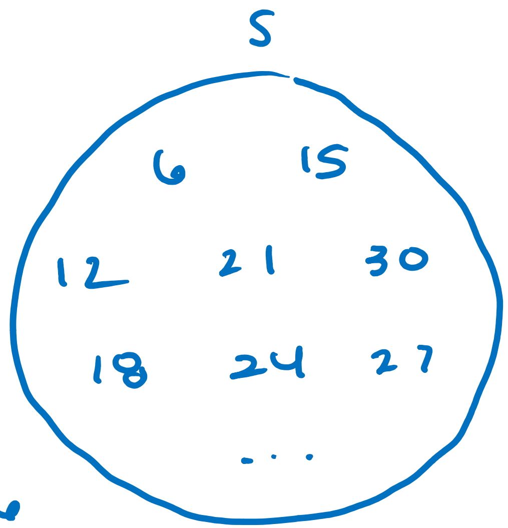
Q1: What is this set?

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

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

$\{6, \underbrace{12, 15, 18, 21, 24, 27 \ldots}\}$

all integers ≥ 12 divisible by 3, and 6



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

Basis Step: $1 \in S$

Recursive Step: If $n \in S$, then $3n \in S$

Structural Induction

Goal is to prove $P(s)$ for all $s \in S$...

recursively defined set

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

Conclusion: Conclude $P()$ holds for all $s \in S$.

Structural Induction Example

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

Show by structural induction that every element of S is divisible by 3.

Structural Induction

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

1. Intro: Let $P(x)$ be "$3 \mid x$". We show $P(x)$ holds for all $x \in S$ by structural induction.
2. Base Case(s): $\boxed{P(6) \text{ and } P(15)}$
   $6 = 3 \cdot 2$. So $3 \mid 6$. So $P(6)$ holds. $15 = 3 \cdot 5$. So $3 \mid 15$. So $P(15)$ holds.
3. Inductive Hypothesis: $\boxed{\text{Suppose } P(x) \text{ and } P(y)}$
   Suppose $P(x)$ and $P(y)$ hold for arbitrary $x, y \in S$.
4. Inductive Step: $\boxed{\text{Goal: show } P(x+y)}$
   By IH, $3 \mid x$. So $x = 3a$ for $a \in \mathbb{Z}$. By IH, $3 \mid y$. So $y = 3b$ for $b \in \mathbb{Z}$.
   Then:
   $x + y = 3a + 3b = 3(a+b)$. So $3 \mid x+y$. Thus $P(x+y)$ holds.

5. Conclusion: Thus $P(x)$ holds for all $x \in S$ by structural induction.

Structural Induction

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

1. Intro: Let $P(x)$ be x is divisible by 3. We show $P(x)$ holds for all $x \in S$ 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:

4. Inductive Step:

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

Structural Induction

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

1. Intro: Let $P(x)$ be x is divisible by 3. We show $P(x)$ holds for all $x \in S$ 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:

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

Structural Induction

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

1. Intro: Let $P(x)$ be x is divisible by 3. We show $P(x)$ holds for all $x \in S$ 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: $\boxed{\text{Goal } P(x + y) \text{ holds}}$

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 $P(x + y)$.

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

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 $P(x)$ for all base cases x in S.

3. Inductive Hypothesis: Suppose $P(x)$ for all x listed as in S in the recursive rules.

4. Inductive Step: Show $P()$ holds for the "new element" given.

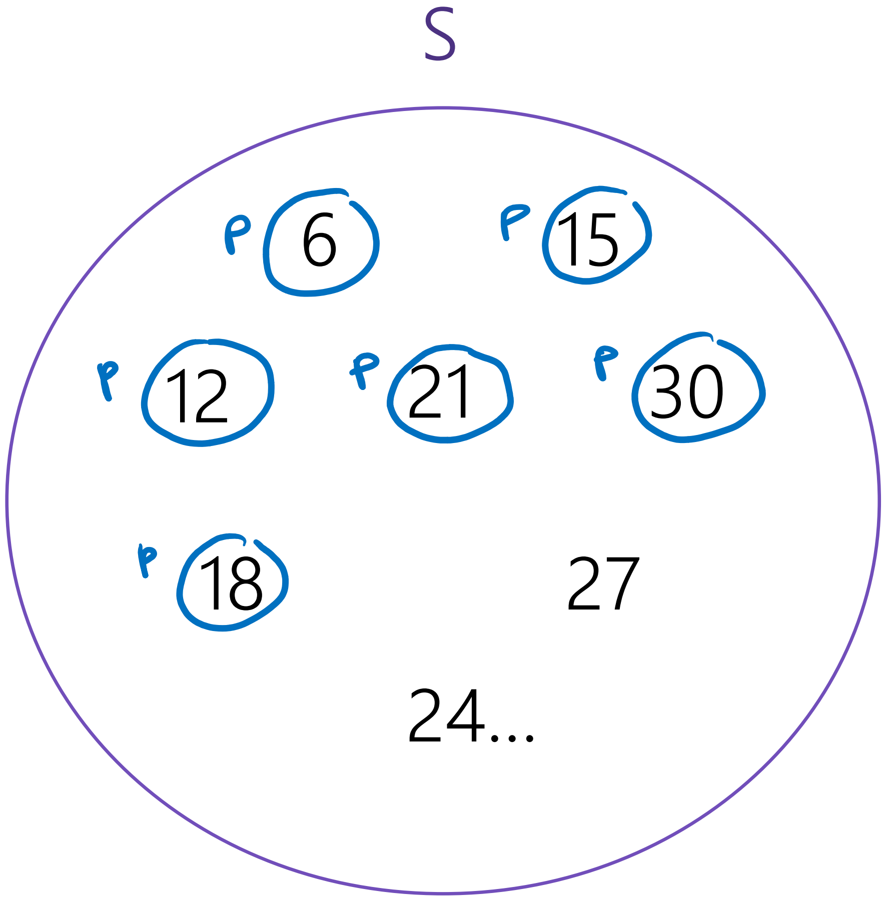
You will need a separate step for every rule.

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

If $x \in S$, $x+1 \in S$ show $P(x+1)$

If $x \in S$, $5x \in S$ show $P(5x)$

Wait a minute! Why can we do this?



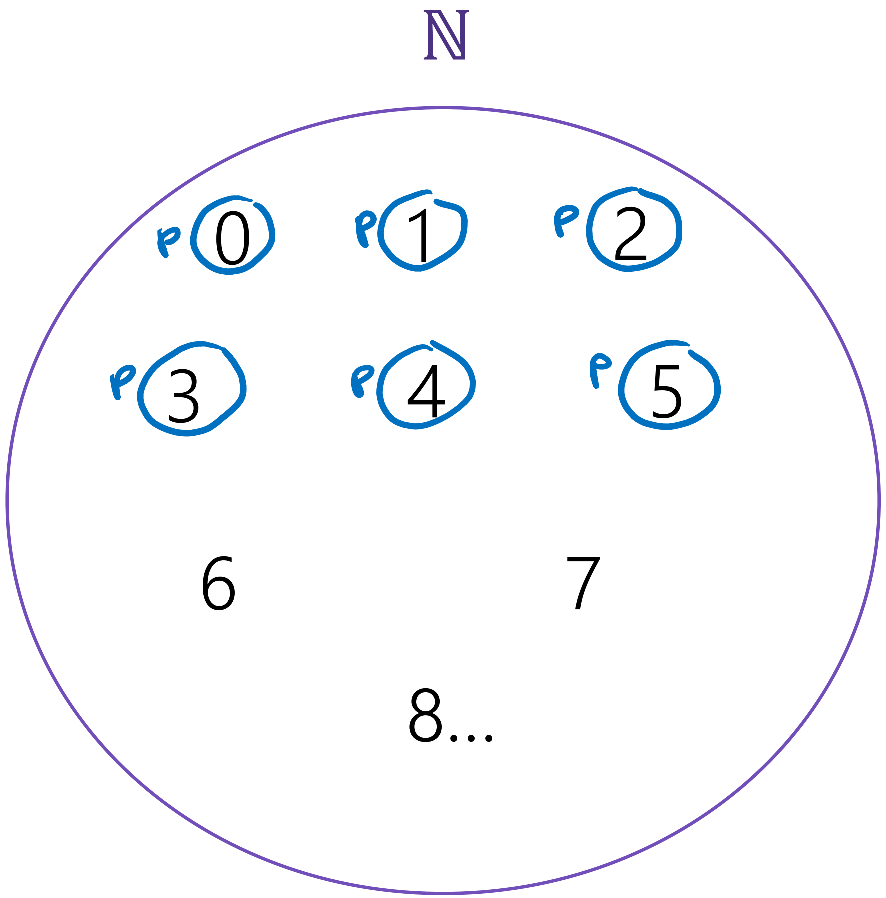
Basis: $6 \in S, 15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$.

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

Strings

Σ^*:

Basis step:

Recursive step:

Σ will be an **alphabet** the set of all the letters you can use in strings.

E.g. $\Sigma = \{0,1\}$

E.g. $\Sigma = \{a, b, c, \dots, z, \_\}$

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

E.g. if $\Sigma = \{0,1\}$, then $01001 \in \Sigma^*$ $\quad 0 \in \Sigma^* \quad 11111 \in \Sigma^* \quad \varepsilon \in \Sigma^*$

E.g. if $\Sigma = \{a, b, c, \dots, z, \_\}$, then $i\_love\_recursive\_sets \in \Sigma^*$

ε is the **empty string**

Analogous to “” in Java (not null)

Strings

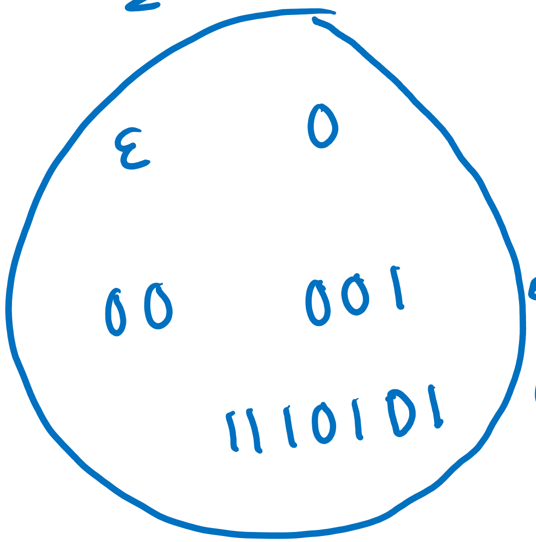
The set of all strings Σ^* can be defined as:

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

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

wa means the string of w with the character a appended

Σ^*



all binary strings

$\Sigma = \{0, 1\}$

If $w =$ "hello" and $a =$ 'c',

$wa =$ "helloc"

↳ same as $w + a$ in Java

Functions on Strings

$P(n)$: "$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$"

↳ Functions on $\mathbb{N}$: sum, product, etc.

$\boxed{a = \varepsilon a}$

Length:

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

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

$\text{len}(abc) = \text{len}(ab) + 1$
$= \text{len}(a) + 1 + 1$
$= \text{len}(\varepsilon a) + 1 + 1$
$= \text{len}(\varepsilon) + 1 + 1 + 1$
$= 0 + 1 + 1 + 1$
$= 3$ (shocking!)

Reversal:

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

$(abc)^R = c(ab)^R$
$= cb(a)^R$
$= cb(\varepsilon a)^R$
$= cba(\varepsilon)^R$
$= cba\varepsilon = cba$

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: len(x·y)=len(x) + len(y) $\underbrace{\text{for all } x, y \in \Sigma^*}_{\text{nested forall}}$.

Let $P(y)$ be "$\mathrm{len}(x \cdot y) = \mathrm{len}(x) + \mathrm{len}(y)$ for all $x \in \Sigma^*$." Prove $P(y)$ for all $y \in \Sigma^*$.

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!

e.g. $P(\varepsilon)$ is "$\mathrm{len}(x \cdot \varepsilon) = \mathrm{len}(x) + \mathrm{len}(\varepsilon)$ for all $x \in \Sigma^*$".

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

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

1. Introduction: 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.
2. Base Case: $\boxed{\text{Goal: } P(\varepsilon)}$
   Let $x \in \Sigma^*$ be arbitrary. Then len(x·ε) = len(x) = len(x) + 0 = len(x) + len(ε).
   So P(ε) holds.
3. Inductive Hypothesis: $\boxed{\text{Suppose } P(w)}$
   Suppose P(w) holds for arbitrary $w \in \Sigma^*$. Then for all $x \in \Sigma^*$,
   len(x·w) = len(x) + len(w).
4. Inductive Step: $\boxed{\text{Goal: } P(wa) \text{ for } a \in \Sigma}$
   Let $x \in \Sigma^*$ be arbitrary. Let $a \in \Sigma$ be arbitrary. Then:

$$
\begin{aligned}
\text{len}(x \cdot wa) &= \text{len}(x \cdot w) + 1 && \text{by def of len} \\
&= \text{len}(x) + \text{len}(w) + 1 && \text{by IH} \\
&= \text{len}(x) + \text{len}(wa) && \text{by def of len}
\end{aligned}
$$

So P(wa) holds.

<include conclusion too>

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

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

1. Introduction: Let $P(y)$ be "$\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x \in \Sigma^*$." We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

2. Base Case:

3. Inductive Hypothesis:

4. Inductive Step:

5. Therefore $P(y)$ holds for all $\mathrm{y} \in \Sigma^*$by the principle of induction.

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

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

1. Introduction: Let $P(y)$ be "$\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x \in \Sigma^*$." We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

2. Base Case: Let x be an arbitrary string. Then $\text{len}(x \cdot \epsilon) = \text{len}(x) = \text{len}(x) + 0 = \text{len}(x) + \text{len}(\varepsilon)$

3. Inductive Hypothesis:

4. Inductive Step:

5. Therefore $P(y)$ holds for all $\mathrm{y} \in \Sigma^*$by the principle of induction.

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

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

1. Introduction: Let $P(y)$ be "$\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x \in \Sigma^*$." We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

2. Base Case: Let x be an arbitrary string. Then $\text{len}(x \cdot \epsilon) = \text{len}(x) = \text{len}(x) + 0 = \text{len}(x) + \text{len}(\varepsilon)$

3. Inductive Hypothesis: Suppose $P(w)$ for an arbitrary $w \in \Sigma^*$.

4. Inductive Step:

5. Therefore $P(y)$ holds for all $\mathrm{y} \in \Sigma^*$by the principle of induction.

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

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

1. Introduction: Let $P(y)$ be "$\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x \in \Sigma^*$." We prove $P(y)$ for all $y \in \Sigma^*$ by structural induction.

2. Base Case: Let x be an arbitrary string. Then $\text{len}(x \cdot \epsilon) = \text{len}(x) = \text{len}(x) + 0 = \text{len}(x) + \text{len}(\varepsilon)$ $P(\varepsilon)$

3. Inductive Hypothesis: Suppose $P(w)$ for an arbitrary $w \in \Sigma^*$.

4. Inductive Step: Let $x \in \Sigma^*$ be an arbitrary string, and $a \in \Sigma$ be arbitrary.

$P(wa)$

$$\begin{aligned}
\text{len}(xwa) &= \text{len}(xw) + 1 && \text{(by definition of len)} \\
&= \text{len}(x) + \text{len}(w) + 1 && \text{(by IH)} \\
&= \text{len}(x) + \text{len}(wa) && \text{(by definition of len)}
\end{aligned}$$

Therefore, $\text{len}(xwa) = \text{len}(x) + \text{len}(wa)$.

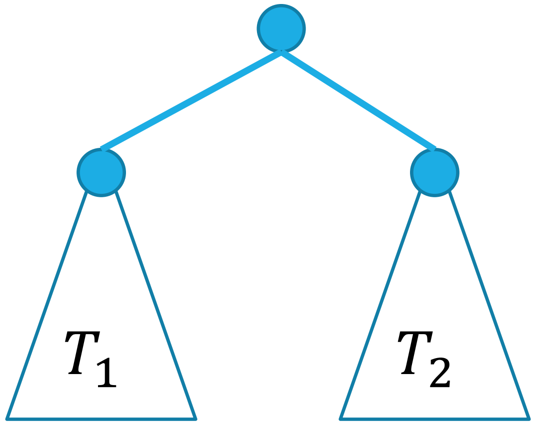
5. Therefore $P(y)$ holds for all $y \in \Sigma^*$ by the principle of induction.

More Structural Sets

Binary Trees are another common source of structural induction.

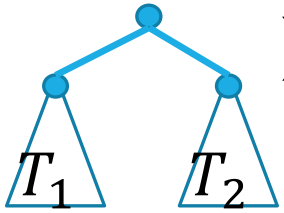
Basis Step: A single node is a rooted binary tree. 

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.

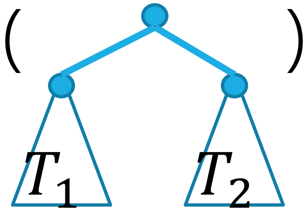


Functions on Binary Trees

size()=1

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

height() = 0

height() = 1+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.