Discrete Mathematics

Recursive Definitions and Structural Induction

Recursively Defined Functions

Giving a recursive definition for a function

To define a recursive function $\mathbf{f}$ over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ $\quad$ $n+1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ $\quad$ $n+1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$ $\quad$ 2^n for $n \in \mathbb{N}$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ $\quad$ $n+1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$ $\quad$ 2^n for $n \in \mathbb{N}$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$ $\quad$ $n!$ for $n \in \mathbb{N}$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ — $n + 1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$ — 2^n for $n \in \mathbb{N}$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$ — $n!$ for $n \in \mathbb{N}$

When the recursive case refers only to $f(n)$, as in these examples, we can prove properties of $f(n)$ easily using ordinary induction.

Example: prove $\mathbf{n}! \leq \mathbf{n}^{\mathbf{n}}$ for all $\mathbf{n} \geq 1$

Example: prove $\mathbf{n! \leq n^n}$ for all $\mathbf{n \geq 1}$

① **Let $\mathbf{P(n)}$ be $\mathbf{n! \leq n^n}$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

Example: prove $\mathbf{n! \le n^n}$ for all $\mathbf{n \ge 1}$

① **Let $\mathbf{P(n)}$ be $\mathbf{n! \le n^n}$.**

We will show that $P(n)$ is true for every integer $n \ge 1$ by induction.

② **Base case ($\mathbf{n = 1}$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

Example: prove $\mathbf{n! \leq n^n}$ for all $\mathbf{n \geq 1}$

① **Let $\mathbf{P(n)}$ be $\mathbf{n! \leq n^n}$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($\mathbf{n = 1}$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 1$.

Example: prove $\mathbf{n! \le n^n}$ for all $\mathbf{n \ge 1}$

① **Let $\mathbf{P(n)}$ be $\mathbf{n! \le n^n}$.**

We will show that $P(n)$ is true for every integer $n \ge 1$ by induction.

② **Base case ($\mathbf{n = 1}$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \ge 1$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $(k+1)! \le (k+1)^{(k+1)}$.

$$\begin{aligned}
(k+1)! &= (k+1) \cdot k! && \text{by definition of !} \\
&\le (k+1) \cdot k^k && \text{by the inductive hypothesis} \\
&\le (k+1) \cdot (k+1)^k && \text{since } k \ge 0 \\
&= (k+1)^{(k+1)} && \text{which is exactly } P(k+1).
\end{aligned}$$

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($n = 1$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 1$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $(k+1)! \leq (k+1)^{(k+1)}$.

$$\begin{aligned}
(k+1)! &= (k+1) \cdot k! && \text{by definition of !} \\
&\leq (k+1) \cdot k^k && \text{by the inductive hypothesis} \\
&\leq (k+1) \cdot (k+1)^k && \text{since } k \geq 0 \\
&= (k+1)^{(k+1)} && \text{which is exactly } P(k+1).
\end{aligned}$$

⑤ **The result follows for all $n \geq 1$ by induction.** ■

Recursively Defined Fibonacci Number

Recursively Defined Functions

Define the Fibonacci numbers as follows:

$f(0) = 1$

$f(1) = 1$

$f(n) = f(n-1) + f(n-2)$ for all $n \in \mathbb{N}, n \geq 2$.

*This is a somewhat unusual definition, $f(0) = 0, f(1) = 1$ is more common.

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Fibonacci Inequality

Show that $f(n) \leq 2^n$ for all $n \geq 0$ by induction.

Fibonacci Inequality

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \leq 2^n$ for all $n \geq 0$ by induction.

Define $P(n)$ to be "$f(n) \leq 2^n$" We show $P(n)$ is true for all $n \geq 0$ by induction on n.

Base Cases: $(n = 0)$: $f(0) = 1 \leq 1 = 2^0$.

$(n = 1)$: $f(1) = 1 \leq 2 = 2^1$.

Inductive Hypothesis: Suppose $P(0) \land P(1) \land \cdots \land P(k)$ for an arbitrary $k \geq 1$.

Inductive step:

Target: $P(k+1)$. i.e. $f(k+1) \leq 2^{k+1}$

Fibonacci Inequality

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \leq 2^n$ for all $n \geq 0$ by induction.

Define $P(n)$ to be "$f(n) \leq 2^n$" We show $P(n)$ is true for all $n \geq 0$ by induction on n.

Base Cases: $(n = 0)$: $f(0) = 1 \leq 1 = 2^0$.

$(n = 1)$: $f(1) = 1 \leq 2 = 2^1$.

Inductive Hypothesis: Suppose $P(0) \land P(1) \land \cdots \land P(k)$ for an arbitrary $k \geq 1$.

Inductive step: $f(k+1) = f(k) + f(k-1)$ by the definition of the Fibonacci numbers. Applying IH twice, we have $f(k+1) \leq 2^k + 2^{k-1} < 2^k + 2^k = 2^{k+1}$.

Therefore, we have $P(n)$ for all $n \geq 0$ by the principle of induction. ■

Fibonacci Inequality Two

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \geq 2^{n/2}$ for all $n \geq 2$ by induction.

[Define $P(n)$]

Base Cases:

Inductive Hypothesis:

Inductive step:

Therefore, we have $P(n)$ for all $n \geq 0$ by the principle of induction.

Fibonacci Inequality Two

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \geq 2^{n/2}$ for all $n \geq 2$ by induction.

Define $P(n)$ to be "$f(n) \geq 2^{n/2}$" We show $P(n)$ is true for all $n \geq 2$ by induction on n.

Base Cases: $f(2) = f(1) + f(0) = 2 \geq 2 = 2^1 = 2^{2/2}$

$f(3) = f(2) + f(1) = 2 + 1 = 3 = 2 \cdot \frac{3}{2} \geq 2\sqrt{2} = 2^{1.5} = 2^{3/2}$

Inductive Hypothesis: Suppose $P(2) \wedge P(3) \wedge \cdots \wedge P(k)$ for an arbitrary $k \geq 3$.

Inductive step: $f(k+1) = f(k) + f(k-1)$ by the definition of the Fibonacci numbers. Applying IH twice, we have

Target: $f(k+1) \geq 2^{(k+1)/2}$

Therefore, we have $P(n)$ for all $n \geq 0$ by the principle of induction.

Fibonacci Inequality Two

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \geq 2^{n/2}$ for all $n \geq 2$ by induction.

Define $P(n)$ to be "$f(n) \geq 2^{n/2}$" We show $P(n)$ is true for all $n \geq 2$ by induction on n.

Base Cases: $f(2) = f(1) + f(0) = 2 \geq 2 = 2^1 = 2^{2/2}$

$f(3) = f(2) + f(1) = 2 + 1 = 3 = 2 \cdot \frac{3}{2} \geq 2\sqrt{2} = 2^{1.5} = 2^{3/2}$

Inductive Hypothesis: Suppose $P(2) \wedge P(3) \wedge \cdots \wedge P(k)$ for an arbitrary $k \geq 3$.

Inductive step: $f(k+1) = f(k) + f(k-1)$ by the definition of the Fibonacci numbers. Applying IH twice, we have

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

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

Therefore, we have $P(n)$ for all $n \geq 0$ by the principle of induction.

Fibonacci Inequality Two

$$f(0) = 1; \qquad f(1) = 1$$
$$f(n) = f(n-1) + f(n-2) \text{ for all } n \in \mathbb{N}, n \geq 2.$$

Show that $f(n) \geq 2^{n/2}$ for all $n \geq 2$ by induction.

Define $P(n)$ to be "$f(n) \geq 2^{n/2}$" We show $P(n)$ is true for all $n \geq 2$ by induction on n.

Base Cases: $f(2) = f(1) + f(0) = 2 \geq 2 = 2^1 = 2^{2/2}$

$f(3) = f(2) + f(1) = 2 + 1 = 3 = 2 \cdot \frac{3}{2} \geq 2\sqrt{2} = 2^{1.5} = 2^{3/2}$

Inductive Hypothesis: Suppose $P(2) \wedge P(3) \wedge \cdots \wedge P(k)$ for an arbitrary $k \geq 3$.

Inductive step: $f(k+1) = f(k) + f(k-1)$ by the definition of the Fibonacci numbers. Applying IH twice, we have

$$\begin{aligned} f(k+1) &\geq 2^{k/2} + 2^{(k-1)/2} \\ &= 2^{(k-1)/2}\left(\sqrt{2} + 1\right) \\ &\geq 2^{(k-1)/2} \cdot 2 \\ &\geq 2^{(k+1)/2} \end{aligned}$$

Therefore, we have $P(n)$ for all $n \geq 0$ by the principle of induction. $\blacksquare$

Recursive Definition of Sets

Giving a recursive definition of a set

A recursive definition of a set S has the following parts:

Basis step specifies one or more initial members of S.

Recursive step specifies the rule(s) for constructing new elements of S from the existing elements.

Exclusion (or closure) rule states that every element in S follows from the basis step and a finite number of recursive steps.

Giving a recursive definition of a set

A recursive definition of a set S has the following parts:

Basis step specifies one or more initial members of S.

Recursive step specifies the rule(s) for constructing new elements of S from the existing elements.

Exclusion (or closure) rule states that every element in S follows from the basis step and a finite number of recursive steps.

The exclusion rule is assumed, so no need to state it explicitly.

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Even natural numbers

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Even natural numbers

Basis: $0 \in S$

Examples of recursively defined sets

Natural numbers

- **Basis:** $0 \in S$
- **Recursive:** if $n \in S$, then $n + 1 \in S$

Even natural numbers

- **Basis:** $0 \in S$
- **Recursive:** if $x \in S$, then $x + 2 \in S$

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Even natural numbers

Basis: $0 \in S$

Recursive: if $x \in S$, then $x + 2 \in S$

Powers of 3

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Even natural numbers

Basis: $0 \in S$

Recursive: if $x \in S$, then $x + 2 \in S$

Powers of 3

Basis: $1 \in S$

Examples of recursively defined sets

Natural numbers

Basis: $0 \in S$

Recursive: if $n \in S$, then $n + 1 \in S$

Even natural 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 $3x \in S$

Structural Induction

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 of the set, if $P()$ holds for that element then $P()$ holds for everything you can make out of it.

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

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

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.

Structural Induction

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

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.

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

Inductive Step:

This gives $P(x + y)$.

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

Structural Induction

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

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.

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

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

Structural Induction Template

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

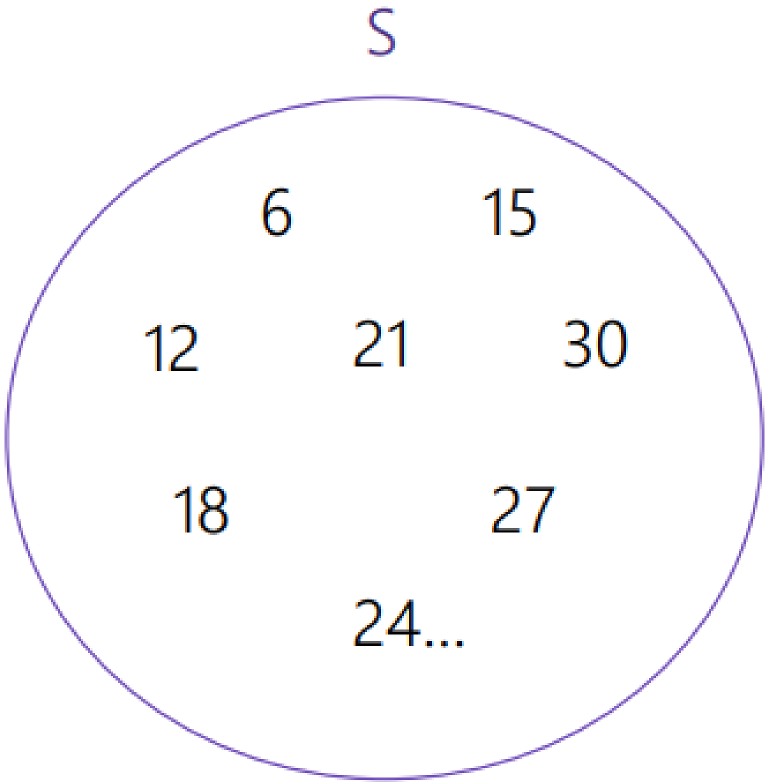
2. Base Case: Show $P(b)$
[Do that for every b in the basis step of defining S]

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

4. Inductive Step: Show $P()$ holds for the “new elements.”
[You will need a separate step for every element created by the recursive rules].

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

Wait a minute! Why can we do this?



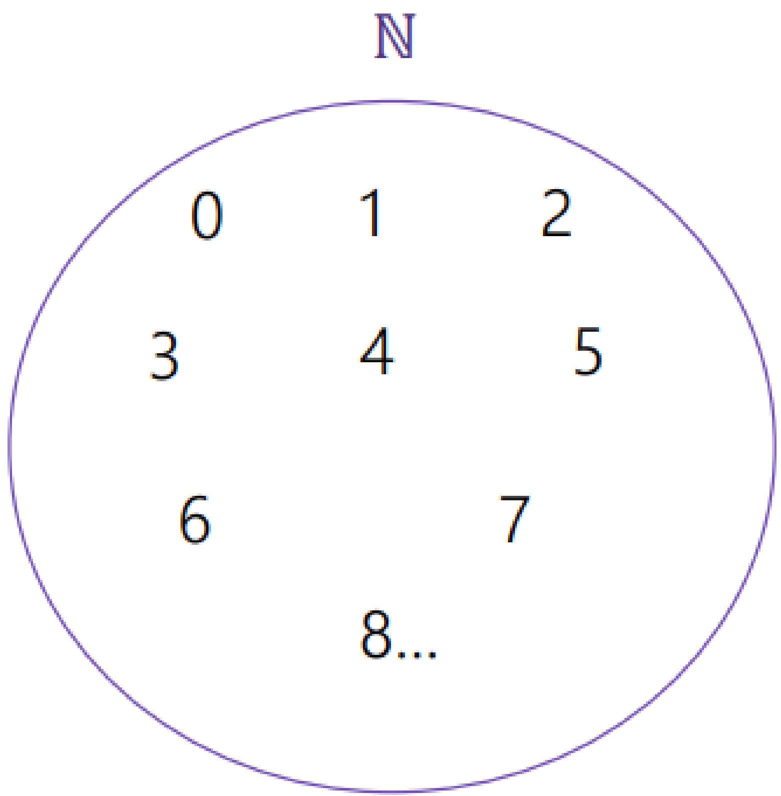
Basis: $6 \in S, 15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$.

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

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

$P(one\ application) \to P(two\ applications)$ 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.

Recursive Definition of Strings

Strings

An *alphabet* Σ is any finite set of characters.
The set Σ^* of *strings* over the alphabet Σ is defined as follows.
Basis: $\varepsilon \in \Sigma^*$, where ε is the empty string.
Recursive: if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Functions on recursively defined sets

Length

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

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

Define Σ^* by

Basis: $\varepsilon \in \Sigma^*$, where ε is the empty string.

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

Functions on recursively defined sets

Length

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

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

Concatenation

$x \bullet \varepsilon = x$ for $x \in \Sigma^*$

$x \bullet (wa) = (x \bullet w)a$ for $x, w \in \Sigma^*, a \in \Sigma$

Define Σ^* by

Basis: $\varepsilon \in \Sigma^*$, where ε is the empty string.

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

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

What object (x or y) to do structural induction on?

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*. \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and $a \in \Sigma$,

then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\operatorname{len}(x \cdot y) = \operatorname{len}(x) + \operatorname{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*. \operatorname{len}(x \cdot y) = \operatorname{len}(x) + \operatorname{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\operatorname{len}(x \cdot \varepsilon) = \operatorname{len}(x) = \operatorname{len}(x) + \operatorname{len}(\varepsilon)$ since $\operatorname{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and
$a \in \Sigma$,
then $wa \in \Sigma^*$

Length

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

$\operatorname{len}(wa) = \operatorname{len}(w) + 1$

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*.\ \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*. \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

④ **Inductive step:**

We want to prove that $P(wa)$ is true for every $a \in \Sigma$.

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and
$a \in \Sigma$,
then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*.\ \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

④ **Inductive step:**

We want to prove that $P(wa)$ is true for every $a \in \Sigma$.

Let $a \in \Sigma$ and $x \in \Sigma^*$ be arbitrary. Then

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*.\ \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

④ **Inductive step:**

We want to prove that $P(wa)$ is true for every $a \in \Sigma$.

Let $a \in \Sigma$ and $x \in \Sigma^*$ be arbitrary. Then

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

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and

$a \in \Sigma$,

then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^* .\ \text{len}(x \cdot y) = \text{len}(x) + \text{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

④ **Inductive step:**

We want to prove that $P(wa)$ is true for every $a \in \Sigma$.

Let $a \in \Sigma$ and $x \in \Sigma^*$ be arbitrary. Then

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

So $\text{len}(x \cdot wa) = \text{len}(x) + \text{len}(wa)$ for all $x \in \Sigma^*$, and $P(wa)$ is true.

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Length

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

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

Concatenation

$x \cdot \varepsilon = x$

$x \cdot (wa) = (x \cdot w)a$

Prove $\mathrm{len}(x \cdot y) = \mathrm{len}(x) + \mathrm{len}(y)$ for all $x, y \in \Sigma^*$

① **Let $P(y)$ be $\forall x \in \Sigma^*.\ \mathrm{len}(x \cdot y) = \mathrm{len}(x) + \mathrm{len}(y)$.**

We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($y = \varepsilon$):**

For every $x \in \Sigma^*$, $\mathrm{len}(x \cdot \varepsilon) = \mathrm{len}(x) = \mathrm{len}(x) + \mathrm{len}(\varepsilon)$ since $\mathrm{len}(\varepsilon) = 0$.
So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

Assume that $P(w)$ is true for some arbitrary $w \in \Sigma^*$.

④ **Inductive step:**

We want to prove that $P(wa)$ is true for every $a \in \Sigma$.

Let $a \in \Sigma$ and $x \in \Sigma^*$ be arbitrary. Then

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

So $\mathrm{len}(x \cdot wa) = \mathrm{len}(x) + \mathrm{len}(wa)$ for all $x \in \Sigma^*$, and $P(wa)$ is true.

⑤ **The result follows for all $y \in \Sigma^*$ by structural induction.** ■

Define Σ^* by

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

Recursive:

if $w \in \Sigma^*$ and

$a \in \Sigma$,

then $wa \in \Sigma^*$

Length

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

$\mathrm{len}(wa) = \mathrm{len}(w) + 1$

Concatenation

$x \cdot \varepsilon = x$

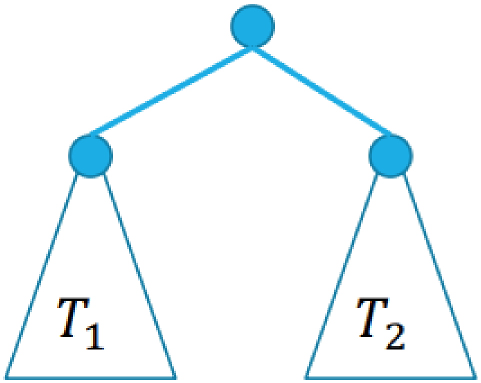
$x \cdot (wa) = (x \cdot w)a$

Recursive Definition of Rooted Binary Trees

Rooted Binary Trees

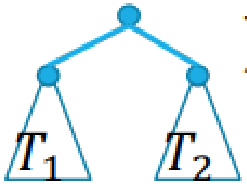
Basis: A single node is a rooted binary tree. 

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.

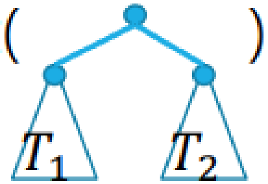


Functions on Binary Trees

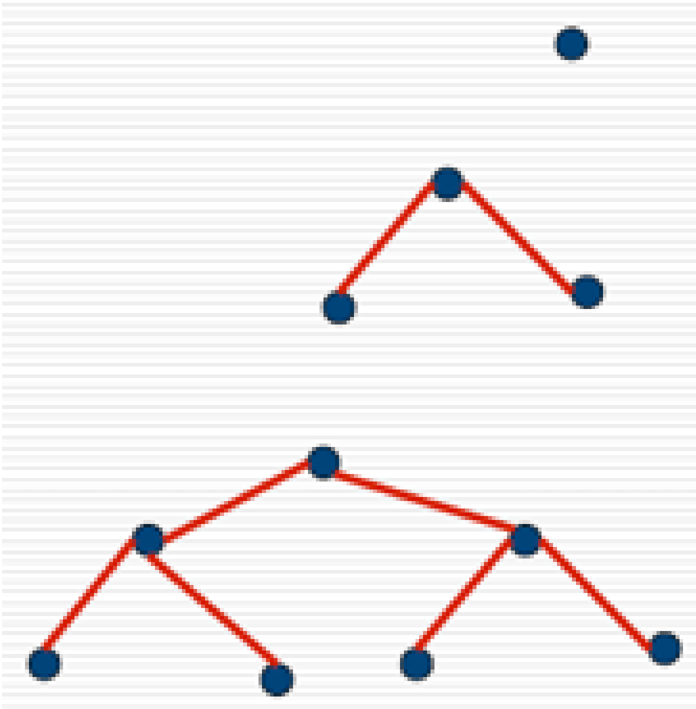
size(●)=1

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

height(●) = 0

height() = 1+max(height(T_1),height(T_2))

Binary tree size

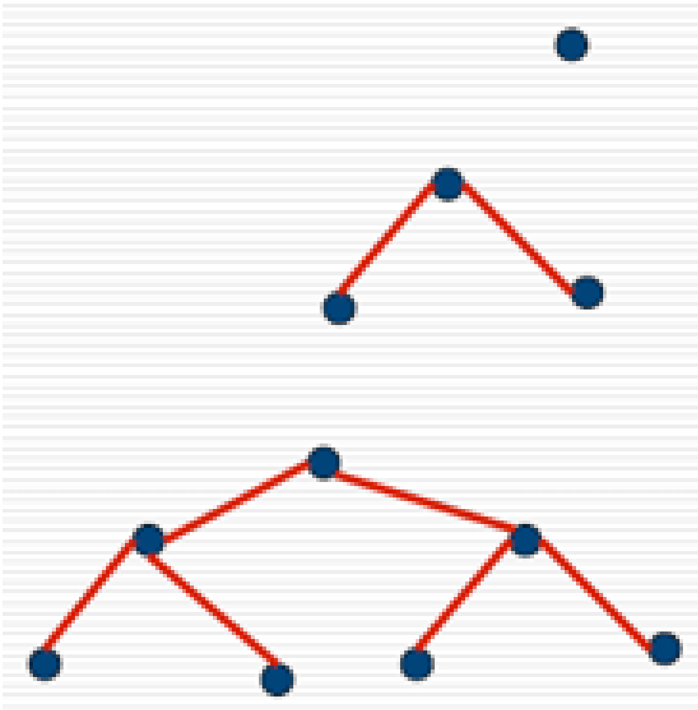


size(T) = 1

size(T) = 1 +1 +1

size(T) = 1 +3 +3 = 7

Binary tree height



height(T) = 0

height(T) = 1 + max(0, 0) = 1

height(T) = 1 + max(1, 1) = 2

Claim

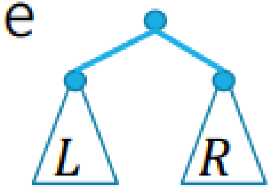
For all trees T, size(T) $\leq 2^{height(T)+1} - 1$

Structural Induction on Binary Trees

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

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

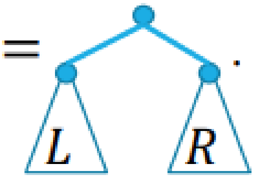
Inductive Hypothesis: Suppose $\mathrm{P}(L)$ and $\mathrm{P}(R)$ hold for arbitrary trees L, R. Let T be the tree



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

Structural Induction on Binary Trees (cont.)

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

$T =$ 

.

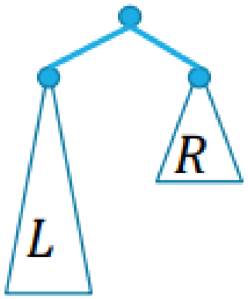
height(T)=$1 + \max\{height(L), height(R)\}$

size(T)= 1 +size(L)+size(R)

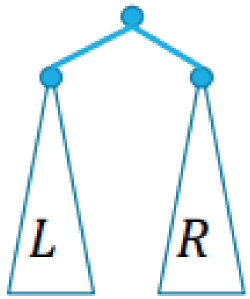
So $P(T)$ holds, and we have $P(T)$ for all binary trees T by the principle of induction.

How do heights compare?

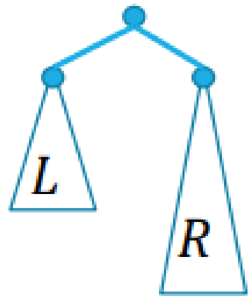
If L is taller than R?



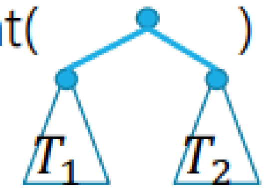
If L, R same height?



If R is taller than L?



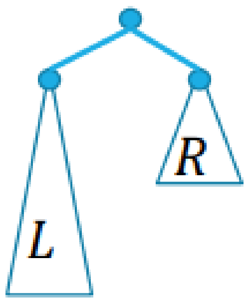
height(●) = 0

height() =

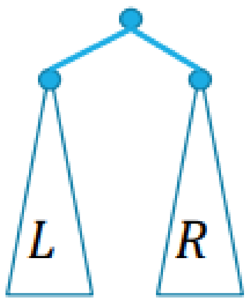
1+max(height(T_1),height(T_2))

How do heights compare?

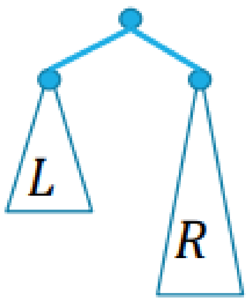
If L is taller than R?



If L, R same height?



If R is taller than L?



height(T) =height$(L) + 1$

height(T) >height$(R) + 1$

height(T) =height$(L) + 1$

height(T) =height$(R) + 1$

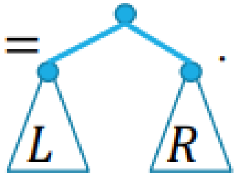
height(T) >height$(L) + 1$

height(T) =height$(R) + 1$

In all cases: height$(T)\geq$height(L)+1, height$(T)\geq$height(R)+1

Structural Induction on Binary Trees (cont.)

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

$T =$ 

.

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

size(T)= 1 +size(L)+size(R)

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

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

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

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