CSE 311 Lecture 19: Structural Induction

Emina Torlak and Kevin Zatloukal

Topics

Homework 6 advice

Start early!

Recursively defined sets

Recursive definitions of sets.

Structural induction

A method for proving properties of recursive structures.

Using structural induction

Example proofs about recursively defined numbers, strings, and trees.

Homework 6 advice

Start early!

Homework 6 isn’t necessarily harder …

But you may find it to be more work than most other assignments.

So please start early :)

Pay special attention to Problem 6.4.

Requires keeping careful track of

(1) what you know and

(2) what you need to prove.

Recursively defined sets

Recursive definitions of sets.

Giving a recursive definition of a set

A recursive definition of a set $\mathbf{S}$ 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 $\mathbf{S}$ 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 $3x \in S$

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 $3x \in S$

Fibonacci 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: $0 \in S$

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

Powers of 3

Basis: $1 \in S$

Recursive: if $x \in S$, then $3x \in S$

Fibonacci numbers

Basis: $(0, 0) \in S$, $(1, 1) \in S$

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 $3x \in S$

Fibonacci numbers

- **Basis:** $(0, 0) \in S$, $(1, 1) \in S$
- **Recursive:** if $(n - 1, x) \in S$ and $(n - 2, y) \in S$, then $(n, x + y) \in S$

More examples of recursively defined sets

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 $wa \in \Sigma^*$

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

Basis: $\varepsilon \in S$ and $a \in S$ for every $a \in \Sigma$

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

Basis: $\varepsilon \in S$ and $a \in S$ for every $a \in \Sigma$

Recursive: if $p \in S$, then $apa \in S$ for every $a \in \Sigma$

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

Basis: $\varepsilon \in S$ and $a \in S$ for every $a \in \Sigma$

Recursive: if $p \in S$, then $apa \in S$ for every $a \in \Sigma$

All binary strings with no 1's before 0's

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

Basis: $\varepsilon \in S$ and $a \in S$ for every $a \in \Sigma$

Recursive: if $p \in S$, then $apa \in S$ for every $a \in \Sigma$

All binary strings with no 1's before 0's

Basis: $\varepsilon \in S$

More examples of recursively defined sets

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 $wa \in \Sigma^*$

Palindromes (strings that are the same forwards and backwards)

Basis: $\varepsilon \in S$ and $a \in S$ for every $a \in \Sigma$

Recursive: if $p \in S$, then $apa \in S$ for every $a \in \Sigma$

All binary strings with no 1's before 0's

Basis: $\varepsilon \in S$

Recursive: if $x \in S$, then $0x \in S$ and $x1 \in S$

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 $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*$, $a \in \Sigma$

Reversal

Define $\boldsymbol{\Sigma^*}$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

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

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

$(wa)^{\mathsf{R}} = aw^{\mathsf{R}}$ 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 $\mathbf{\Sigma^*}$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

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

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

Number of $\mathbf{c}$'s in a string

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

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

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

Number of $\mathbf{c}$'s in a string

$\#_c(\varepsilon) = 0$

Define $\boldsymbol{\Sigma}^*$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

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

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

Number of $\mathbf{c}$'s in a string

$\#_c(\varepsilon) = 0$

$\#_c(wc) = \#_c(w) + 1$ for $w \in \Sigma^*$

Define $\boldsymbol{\Sigma^*}$ 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 $w \in \Sigma^*, a \in \Sigma$

Reversal

$\varepsilon^{\mathsf{R}} = \varepsilon$

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

Concatenation

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

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

Number of $\mathbf{c}$'s in a string

$\#_c(\varepsilon) = 0$

$\#_c(wc) = \#_c(w) + 1$ for $w \in \Sigma^*$

$\#_c(wa) = \#_c(w)$ for $w \in \Sigma^*, a \in \Sigma, a \neq c$

Define $\boldsymbol{\Sigma^*}$ by

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

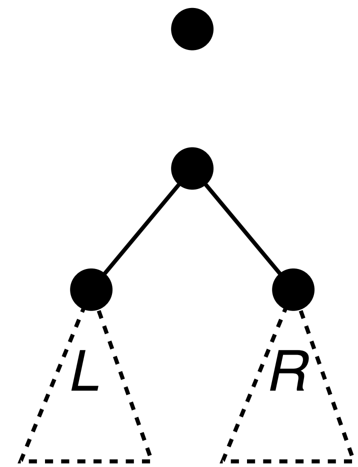
Recursive: if $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$

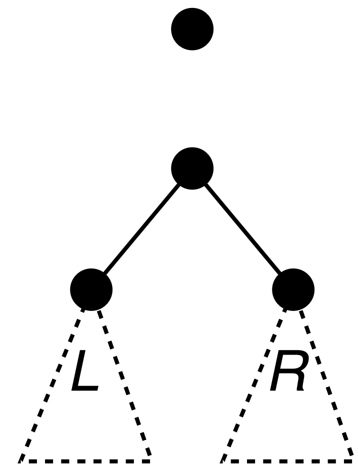


Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$



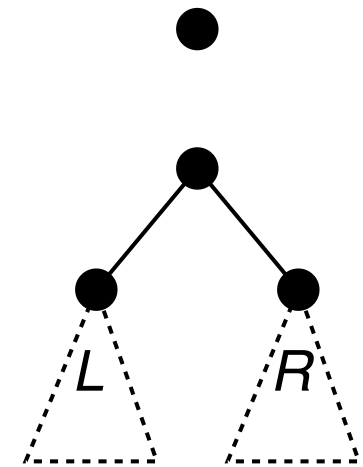
Size of a rooted binary tree

Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\mathsf{Tree}(\bullet, L, R) \in S$



Size of a rooted binary tree

$|\bullet| = 1$

Rooted binary trees and functions on them

Rooted binary trees

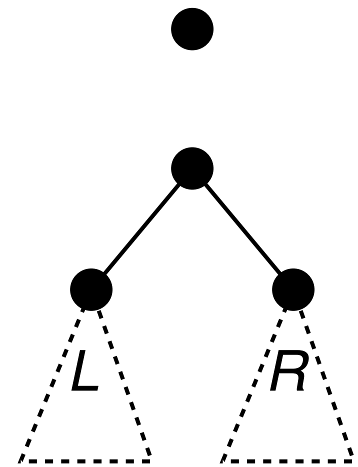
Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$

Size of a rooted binary tree

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| = 1 + |L| + |R|$

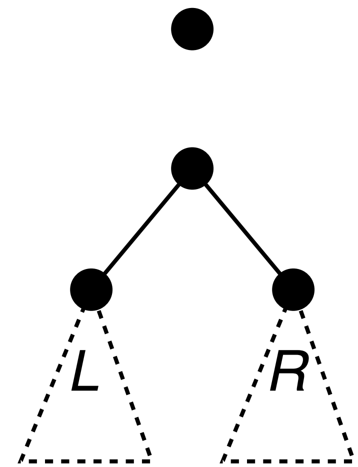


Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$



Size of a rooted binary tree

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| = 1 + |L| + |R|$

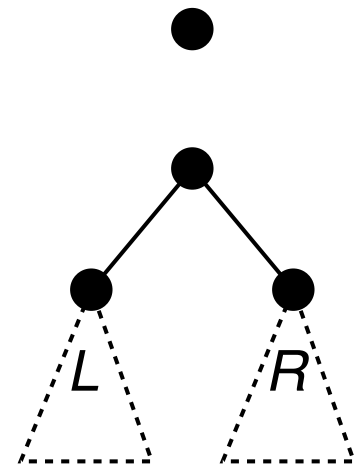
Height of a rooted binary tree

Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$



Size of a rooted binary tree

$| \bullet | = 1$

$|\text{Tree}(\bullet, L, R)| = 1 + |L| + |R|$

Height of a rooted binary tree

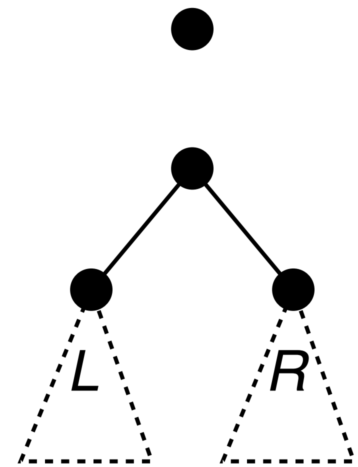
$\lceil \bullet \rceil = 0$

Rooted binary trees and functions on them

Rooted binary trees

Basis: $\bullet \in S$

Recursive: if $L \in S$ and $R \in S$, then $\text{Tree}(\bullet, L, R) \in S$



Size of a rooted binary tree

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| = 1 + |L| + |R|$

Height of a rooted binary tree

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R) \rceil = 1 + \max(\lceil L \rceil, \lceil R \rceil)$

Structural induction

A method for proving properties of recursive structures.

How can we prove properties of recursive structures?

Suppose that $\mathbf{S}$ is a recursively defined set.

And we want to prove that every element of S satisfies a predicate P.

Can we use ordinary induction to prove $\forall \mathbf{x} \in \mathbf{S}. \mathbf{P}(\mathbf{x})$?

How can we prove properties of recursive structures?

Suppose that $\mathbf{S}$ is a recursively defined set.

And we want to prove that every element of S satisfies a predicate P.

Can we use ordinary induction to prove $\mathbf{\forall x \in S. P(x)}$?

Yes! Define $Q(n)$ to be "for all $x \in S$ that can be constructed in at most n recursive steps, $P(x)$ is true."

How can we prove properties of recursive structures?

Suppose that $\mathbf{S}$ is a recursively defined set.

And we want to prove that every element of S satisfies a predicate P.

Can we use ordinary induction to prove $\forall \mathbf{x} \in \mathbf{S}. \mathbf{P(x)}$?

Yes! Define $Q(n)$ to be "for all $x \in S$ that can be constructed in at most n recursive steps, $P(x)$ is true."

But this proof would be long and cumbersome to do!

So we use **structural induction** instead.

- Follows from ordinary induction (on Q), while providing a more convenient proof template for reasoning about recursive structures.
- As powerful as ordinary induction, which is just structural induction applied to the recursively defined set of natural numbers.

Proving $\forall x \in S.\ P(x)$ by structural induction

① **Let $P(x)$ be** *[definition of $P(x)$]*.

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases:**

[Proof of $P(s_0), \ldots, P(s_m)$.]

③ **Inductive hypothesis:**

Assume that $P(y_0), \ldots, P(y_k)$ are true for some arbitrary $y_0, \ldots, y_k \in S$.

④ **Inductive step:**

We want to prove that $P(y)$ is true.

[Proof of $P(y)$. The proof ***must*** *invoke the structural inductive hypothesis.]*

⑤ **The result follows for all $x \in S$ by structural induction.**

Recursive definition of S

Basis step:

$s_0 \in S, \ldots, s_m \in S$.

Recursive step:

if $y_0, \ldots, y_k \in S$, then $y \in S$.

Proving $\forall \mathbf{x} \in \mathbf{S}.\ \mathbf{P(x)}$ by structural induction

① **Let $\mathbf{P(x)}$ be** *[definition of $\boldsymbol{P(x)}$]*.

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases:**

[Proof of $P(s_0), \ldots, P(s_m)$.]

③ **Inductive hypothesis:**

Assume that $P(y_0), \ldots, P(y_k)$ are true for some arbitrary $y_0, \ldots, y_k \in S$.

④ **Inductive step:**

We want to prove that $P(y)$ is true.

*[Proof of $P(y)$. The proof **must** invoke the structural inductive hypothesis.]*

⑤ **The result follows for all $\mathbf{x} \in \mathbf{S}$ by structural induction.**

Recursive definition of $\mathbf{S}$

Basis step:

$s_0 \in S, \ldots, s_m \in S$.

Recursive step:

if $y_0, \ldots, y_k \in S$, then $y \in S$.

If the **recursive step** of S includes multiple rules for constructing new elements from existing elements, then ③ **assume** P for the existing elements in every rule, and ④ **prove** P for the new element in every rule.

Using structural induction

Example proofs about recursively defined numbers, strings, and trees.

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

Define $\mathbf{S}$ by

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

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

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

Define $\mathbf{S}$ by

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

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

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

Define $\mathbf{S}$ by

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

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

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x), P(y)$ are true for some arbitrary $x, y \in S$.

Define $\mathbf{S}$ by

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

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

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

Define $\mathbf{S}$ by

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

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

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x), P(y)$ are true for some arbitrary $x, y \in S$.

④ **Inductive step:**

We want to prove that $P(x + y)$ is true.

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x), P(y)$ are true for some arbitrary $x, y \in S$.

④ **Inductive step:**

We want to prove that $P(x + y)$ is true.
By the inductive hypothesis, $3|x$ and $3|y$, so $x = 3i$ and $y = 3j$ for some $i, j \in \mathbb{Z}$.

Define $\mathbf{S}$ by

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

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

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

Define $\mathbf{S}$ by

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

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

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x), P(y)$ are true for some arbitrary $x, y \in S$.

④ **Inductive step:**

We want to prove that $P(x + y)$ is true.

By the inductive hypothesis, $3|x$ and $3|y$, so $x = 3i$ and $y = 3j$ for some $i, j \in \mathbb{Z}$. Therefore, $x + y = 3i + 3j = 3(i + j)$ so $3|(x + y)$.

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

Define $\mathbf{S}$ by

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

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

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x)$, $P(y)$ are true for some arbitrary $x, y \in S$.

④ **Inductive step:**

We want to prove that $P(x + y)$ is true.
By the inductive hypothesis, $3|x$ and $3|y$, so $x = 3i$ and $y = 3j$ for some $i, j \in \mathbb{Z}$. Therefore, $x + y = 3i + 3j = 3(i + j)$ so $3|(x + y)$. Hence, $P(x + y)$ is true.

Prove that every $\mathbf{x} \in \mathbf{S}$ is divisible by 3

Define $\mathbf{S}$ by

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

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

① **Let $\mathbf{P(x)}$ be $\mathbf{3|x}$.**

We will show that $P(x)$ is true for every $x \in S$ by structural induction.

② **Base cases ($\mathbf{x = 6, x = 15}$):**

$3|6$ so $P(6)$ holds, and $3|15$ so $P(15)$ holds.

③ **Inductive hypothesis:**

Assume that $P(x)$, $P(y)$ are true for some arbitrary $x, y \in S$.

④ **Inductive step:**

We want to prove that $P(x + y)$ is true.
By the inductive hypothesis, $3|x$ and $3|y$, so $x = 3i$ and $y = 3j$ for some $i, j \in \mathbb{Z}$. Therefore, $x + y = 3i + 3j = 3(i + j)$ so $3|(x + y)$. Hence, $P(x + y)$ is true.

⑤ **The result follows for all $\mathbf{x} \in \mathbf{S}$ by structural induction.**

Prove len$(\mathbf{x} \bullet \mathbf{y})$ = len$(\mathbf{x})$ + len$(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \boldsymbol{\Sigma}^*$

Define $\boldsymbol{\Sigma}^*$ by

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

Recursive:

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

then $wa \in \Sigma^*$

Length

len$(\varepsilon) = 0$

len$(wa) =$ len$(w) + 1$

Concatenation

$x \bullet \varepsilon = x$

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

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

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

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

$x \bullet \varepsilon = x$

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

Prove len$(\mathbf{x} \bullet \mathbf{y})$ = len$(\mathbf{x})$ + len$(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \boldsymbol{\Sigma}^*$

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

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

Define $\boldsymbol{\Sigma}^*$ by

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

Recursive:

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

Length

len$(\varepsilon) = 0$

len(wa) = len$(w) + 1$

Concatenation

$x \bullet \varepsilon = x$

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

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

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ since $\text{len}(\varepsilon) = 0$. 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

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

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

Concatenation

$x \bullet \varepsilon = x$

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

Prove len($\mathbf{x} \bullet \mathbf{y}$) = len($\mathbf{x}$) + len($\mathbf{y}$) for all $\mathbf{x}, \mathbf{y} \in \boldsymbol{\Sigma}^*$

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, len$(x \bullet \varepsilon) =$ len$(x) =$ len$(x) +$ len(ε) since len$(\varepsilon) = 0$. So $P(\varepsilon)$ is true.

③ **Inductive hypothesis:**

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

Define $\boldsymbol{\Sigma}^*$ by

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

Recursive:

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

Length

len$(\varepsilon) = 0$

len$(wa) =$ len$(w) + 1$

Concatenation

$x \bullet \varepsilon = x$

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

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

① **Let $\mathbf{P}(\mathbf{y})$ be $\forall \mathbf{x} \in \Sigma^*. \text{len}(\mathbf{x} \bullet \mathbf{y}) = \text{len}(\mathbf{x}) + \text{len}(\mathbf{y})$.**
We will show that $P(y)$ is true for every $y \in \Sigma^*$ by structural induction.

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**
For every $x \in \Sigma^*$, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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
$x \bullet \varepsilon = x$
$x \bullet (wa) = (x \bullet w)a$

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

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, $\mathrm{len}(x \bullet \varepsilon) = \mathrm{len}(x) = \mathrm{len}(x) + \mathrm{len}(\varepsilon)$ 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

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

$x \bullet \varepsilon = x$

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

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

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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

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

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

$x \bullet \varepsilon = x$

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

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

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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

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

So $\text{len}(x \bullet wa) = \text{len}(x) + \text{len}(wa)$ 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

$x \bullet \varepsilon = x$

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

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

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

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

② **Base case ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

For every $x \in \Sigma^*$, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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

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

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

⑤ **The result follows for all $\mathbf{y} \in \Sigma^*$ 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

$x \bullet \varepsilon = x$

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

Prove $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$ for every rooted binary tree $\mathbf{t}$

Define $\mathbf{S}$ by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

Tree$(\bullet, L, R) \in S$

Size

$| \bullet | = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$ for every rooted binary tree $\mathbf{t}$

① **Let $\mathbf{P(t)}$ be $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

Define $\mathbf{S}$ by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

Tree$(\bullet, L, R) \in S$

Size

$| \bullet | = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|t| \leq 2^{\lceil t \rceil + 1} - 1$ for every rooted binary tree t

① **Let $\mathbf{P(t)}$ be $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

② **Base case ($\mathbf{t} = \bullet$):**

$|\bullet| = 1 = 2^1 - 1 = 2^{0+1} - 1 = 2^{\lceil \bullet \rceil + 1} - 1$ so $P(\bullet)$ is true.

Define S by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

Tree$(\bullet, L, R) \in S$

Size

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$ for every rooted binary tree $\mathbf{t}$

① **Let $\mathbf{P(t)}$ be $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

② **Base case ($\mathbf{t} = \bullet$):**

$|\bullet| = 1 = 2^1 - 1 = 2^{0+1} - 1 = 2^{\lceil \bullet \rceil + 1} - 1$ so $P(\bullet)$ is true.

③ **Inductive hypothesis:**

Assume that $P(L)$ and $P(R)$ are true for some arbitrary $L, R \in S$.

Define $\mathbf{S}$ by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

Tree$(\bullet, L, R) \in S$

Size

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|t| \leq 2^{\lceil t \rceil + 1} - 1$ for every rooted binary tree t

① **Let $\mathbf{P(t)}$ be $|\mathbf{t}| \leq \mathbf{2^{\lceil t \rceil + 1} - 1}$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

② **Base case ($\mathbf{t} = \bullet$):**

$|\bullet| = 1 = 2^1 - 1 = 2^{0+1} - 1 = 2^{\lceil \bullet \rceil + 1} - 1$ so $P(\bullet)$ is true.

③ **Inductive hypothesis:**

Assume that $P(L)$ and $P(R)$ are true for some arbitrary $L, R \in S$.

④ **Inductive step:**

We want to prove that $P(\text{Tree}(\bullet, L, R))$ is true.

Define $\mathbf{S}$ by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

$\text{Tree}(\bullet, L, R) \in S$

Size

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|t| \leq 2^{\lceil t \rceil+1} - 1$ for every rooted binary tree t

① **Let $P(t)$ be $|t| \leq 2^{\lceil t \rceil+1} - 1$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

② **Base case ($t = \bullet$):**

$|\bullet| = 1 = 2^1 - 1 = 2^{0+1} - 1 = 2^{\lceil \bullet \rceil+1} - 1$ so $P(\bullet)$ is true.

③ **Inductive hypothesis:**

Assume that $P(L)$ and $P(R)$ are true for some arbitrary $L, R \in S$.

④ **Inductive step:**

We want to prove that $P(\text{Tree}(\bullet, L, R))$ is true.

$$\begin{aligned}
|\text{Tree}(\bullet, L, R)| &= 1 + |L| + |R| && \text{by defn of } || \\
&\leq 1 + (2^{\lceil L \rceil+1} - 1) + (2^{\lceil R \rceil+1} - 1) && \text{by IH} \\
&\leq 2^{\lceil L \rceil+1} + 2^{\lceil R \rceil+1} - 1 && \text{algebra} \\
&\leq 2(2^{\max(\lceil L \rceil, \lceil R \rceil)+1}) - 1 && \text{by defn of max} \\
&= 2(2^{\lceil \text{Tree}(\bullet, L, R) \rceil}) - 1 && \text{by defn of } \lceil\rceil \\
&= 2^{\lceil \text{Tree}(\bullet, L, R) \rceil+1} - 1 && \text{which is the desired result.}
\end{aligned}$$

Define S by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

$\text{Tree}(\bullet, L, R) \in S$

Size

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Prove $|t| \leq 2^{\lceil t \rceil + 1} - 1$ for every rooted binary tree t

① **Let $P(t)$ be $|t| \leq 2^{\lceil t \rceil + 1} - 1$.**

We will show that $P(t)$ is true for every $t \in S$ by structural induction.

② **Base case ($t = \bullet$):**

$|\bullet| = 1 = 2^1 - 1 = 2^{0+1} - 1 = 2^{\lceil \bullet \rceil + 1} - 1$ so $P(\bullet)$ is true.

③ **Inductive hypothesis:**

Assume that $P(L)$ and $P(R)$ are true for some arbitrary $L, R \in S$.

④ **Inductive step:**

We want to prove that $P(\text{Tree}(\bullet, L, R))$ is true.

$$
\begin{aligned}
|\text{Tree}(\bullet, L, R)| &= 1 + |L| + |R| && \text{by defn of } |\,| \\
&\leq 1 + (2^{\lceil L \rceil + 1} - 1) + (2^{\lceil R \rceil + 1} - 1) && \text{by IH} \\
&\leq 2^{\lceil L \rceil + 1} + 2^{\lceil R \rceil + 1} - 1 && \text{algebra} \\
&\leq 2(2^{\max(\lceil L \rceil, \lceil R \rceil) + 1}) - 1 && \text{by defn of max} \\
&= 2(2^{\lceil \text{Tree}(\bullet, L, R) \rceil}) - 1 && \text{by defn of } \lceil\,\rceil \\
&= 2^{\lceil \text{Tree}(\bullet, L, R) \rceil + 1} - 1 && \text{which is the desired result.}
\end{aligned}
$$

⑤ **The result follows for all $t \in S$ by structural induction.**

Define S by

Basis: $\bullet \in S$.

Recursive:

if $L, R \in S$, then

$\text{Tree}(\bullet, L, R) \in S$

Size

$|\bullet| = 1$

$|\text{Tree}(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil \text{Tree}(\bullet, L, R)) \rceil =$

$1 + \max(\lceil L \rceil, \lceil R \rceil$

Summary

To define a set recursively, specify its basis and recursive step.

Recursive set definitions assume the *exclusion rule*.

We use recursive functions to operate on elements of recursive sets.

Use structural induction to prove properties of recursive structures.

Structural induction follows from ordinary induction but is easier to use.

To prove $\forall \mathbf{x} \in \mathbf{S}.\, \mathbf{P}(\mathbf{x})$ using structural induction:

Show that P holds for the elements in the basis step of S.

Assume P for every existing element of S named in the recursive step.

Prove P for every new element of S created in the recursive step.