CSE 311 Lecture 20: Regular Expressions

Emina Torlak and Kevin Zatloukal

Topics

Structural induction

A brief review of Lecture 19.

Using structural induction

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

Regular expressions

Definition, examples, applications.

Structural induction

A brief review of Lecture 19.

Structural induction proof template

① **Let $\mathbf{P(x)}$ 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 $\mathbf{x} \in \mathbf{S}$ 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 $y \in S$.

Structural induction proof template

① **Let $\mathbf{P(x)}$ 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 $\mathbf{x \in S}$ 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.

Structural induction works just like ordinary induction

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

We will show that $P(x)$ is true for every $x \in \mathbb{N}$ by structural induction.

② **Base cases:**

[Proof of $P(0)$.]

③ **Inductive hypothesis:**

Assume that $P(n)$ is true for some arbitrary $n \in \mathbb{N}$.

④ **Inductive step:**

We want to prove that $P(n+1)$ is true.

[Proof of $P(n+1)$. The proof ***must*** *invoke the structural inductive hypothesis.]*

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

Recursive definition of $\mathbb{N}$

Basis step: $0 \in \mathbb{N}$.

Recursive step:

if $n \in \mathbb{N}$, then

$n + 1 \in \mathbb{N}$.

Ordinary induction is just structural induction applied to the recursively defined set of natural numbers!

Understanding structural induction

$$\frac{P(\bullet);\ \forall L, R \in S.\,(P(L) \land P(R)) \to P(\mathsf{Tree}(\bullet, L, R))}{\therefore\ \forall x \in S.\,P(x)}$$

How do we get $P(\mathsf{Tree}(\bullet, \mathsf{Tree}(\bullet, \bullet, \bullet)))$ from $P(\bullet)$ and $\forall L, R \in S.\,(P(L) \land P(R)) \to P(\mathsf{Tree}(\bullet, L, R))$?

Define S by

Basis: $\bullet \in S$.

Recursive:

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

1. First, we have $\forall L, R \in S.\,(P(L) \land P(R)) \to P(\mathsf{Tree}(\bullet, L, R))$
2. Next, we have $P(\bullet)$.
3. Intro $\land$ on 2 gives us $P(\bullet) \land P(\bullet)$.
4. Elim $\forall$ on 1 gives us $(P(\bullet) \land P(\bullet)) \to P(\mathsf{Tree}(\bullet, \bullet, \bullet))$.
5. Modus Ponens on 3 and 4 gives us $P(\mathsf{Tree}(\bullet, \bullet, \bullet))$.
6. Intro $\land$ on 2 and 5 gives us $P(\bullet) \land P(\mathsf{Tree}(\bullet, \bullet, \bullet))$.
7. Elim $\forall$ on 1 gives us
   $(P(\bullet) \land P(\mathsf{Tree}(\bullet, \bullet, \bullet))) \to P(\mathsf{Tree}(\bullet, \mathsf{Tree}(\bullet, \bullet, \bullet)))$.
8. Modus Ponens on 6 and 7 gives us $P(\mathsf{Tree}(\bullet, \mathsf{Tree}(\bullet, \bullet, \bullet)))$.

$\mathbf{P(\bullet)}$

$\mathbf{P(\bullet) \land P(\bullet)}$

$\Downarrow$ $(P(\bullet) \land P(\bullet)) \to P(\mathsf{Tree}(\bullet,\bullet,\bullet))$

$\mathbf{P(\mathsf{Tree}(\bullet, \bullet, \bullet))}$

$\mathbf{P(\bullet) \land P(\mathsf{Tree}(\bullet, \bullet, \bullet))}$

$\Downarrow$ $(P(\bullet) \land P(\mathsf{Tree}(\bullet,\bullet,\bullet))) \to P(\mathsf{Tree}(\bullet,\mathsf{Tree}(\bullet,\bullet,\bullet)$

$\mathbf{P(\mathsf{Tree}(\bullet, \mathsf{Tree}(\bullet, \bullet, \bullet)))}$

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

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.

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 $x, y \in S$.

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 $\mathbf{x}, \mathbf{y} \in \boldsymbol{\Sigma}^*$

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

Define $\boldsymbol{\Sigma}^*$ 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 $\mathbf{x}, \mathbf{y} \in \mathbf{\Sigma}^*$

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

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

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

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

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

② **Base case ($y = \varepsilon$):**

Let x in Σ^* be arbitrary.

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

Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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 $\mathbf{x}, \mathbf{y} \in \Sigma^*$

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

Let x in Σ^* be arbitrary. Then, 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 Σ^* 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 $\mathbf{P(y)}$ 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 ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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 $\text{len}(\mathbf{x} \bullet \mathbf{y}) = \text{len}(\mathbf{x}) + \text{len}(\mathbf{y})$ 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 ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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

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

Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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 ($\mathbf{y} = \boldsymbol{\varepsilon}$):**

Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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 ($\mathbf{y} = \varepsilon$):**
Let x in Σ^* be arbitrary. Then, $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ 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

$| \bullet | = 1$

$|$Tree$(\bullet, L, R)| =$

$1 + |L| + |R|$

Height

$\lceil \bullet \rceil = 0$

$\lceil$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.

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.

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 $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$ 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 $L, R \in S$.

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 $|\mathbf{t}| \leq 2^{\lceil \mathbf{t} \rceil + 1} - 1$ 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.

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 $\mathbf{S}$ 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 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 $L, R \in S$.

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 $\mathbf{t} \in \mathbf{S}$ by structural induction.**

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

$\lceil \bullet \rceil = 0$

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

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

Regular expressions

Definition, examples, applications.

Sets of strings as languages

A *language* is a sets of strings with specific syntax, e.g.:

Syntactically correct Java/C/C++ programs.

The set Σ^* of all strings over the alphabet Σ.

Palindromes over Σ.

Binary strings with no 1's before 0's.

Sets of strings as languages

A *language* is a sets of strings with specific syntax, e.g.:

Syntactically correct Java/C/C++ programs.

The set Σ^* of all strings over the alphabet Σ.

Palindromes over Σ.

Binary strings with no 1's before 0's.

Regular expressions let us specify *regular languages*, e.g.:

All binary strings.

The strings $\{0000, 0010, 1000, 1010\}$.

All strings that contain the string "CSE311".

Regular expressions over Σ: syntax

Basis step:

$\emptyset, \varepsilon$ are regular expressions.

a is a regular expression for any $a \in \Sigma$.

Recursive step:

If A and B are regular expressions, then so are AB, $A \cup B$, and A^*.

Regular expressions over Σ: syntax

Basis step:

$\emptyset, \varepsilon$ are regular expressions.

a is a regular expression for any $a \in \Sigma$.

Recursive step:

If A and B are regular expressions, then so are AB, $A \cup B$, and A^*.

Examples: regular expressions of $\Sigma = \{0, 1\}$

Basis: $\emptyset, \varepsilon, 0, 1$.

Recursive: $01011, 0^*1^*, (0 \cup 1)0(0 \cup 1)0$, etc.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

$\emptyset$ represents the set with no strings.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

$\emptyset$ represents the set with no strings.

ε represents the set $\{\varepsilon\}$.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

- $\emptyset$ represents the set with no strings.
- ε represents the set $\{\varepsilon\}$.
- a represents the set $\{a\}$.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

$\emptyset$ represents the set with no strings.

ε represents the set $\{\varepsilon\}$.

a represents the set $\{a\}$.

AB represents the concatenation of the sets represented by A and B: $\{a \bullet b \mid a \in A, b \in B\}$.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

$\emptyset$ represents the set with no strings.

ε represents the set $\{\varepsilon\}$.

a represents the set $\{a\}$.

AB represents the concatenation of the sets represented by A and B: $\{a \bullet b \mid a \in A, b \in B\}$.

$A \cup B$ represents the union of the sets represented by A and B: $A \cup B$.

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

- $\emptyset$ represents the set with no strings.
- ε represents the set $\{\varepsilon\}$.
- a represents the set $\{a\}$.
- AB represents the concatenation of the sets represented by A and B: $\{a \bullet b \mid a \in A, b \in B\}$.
- $A \cup B$ represents the union of the sets represented by A and B: $A \cup B$.
- A^* represents the concatenation of the set represented by A with itself zero or more times: $A^* = \{\varepsilon\} \cup A \cup AA \cup AAA \cup AAAA \cup \ldots$

Regular expressions over Σ: semantics

A regular expression over Σ represents a set of strings over Σ.

$\emptyset$ represents the set with no strings.

ε represents the set $\{\varepsilon\}$.

a represents the set $\{a\}$.

AB represents the concatenation of the sets represented by A and B: $\{a \bullet b \mid a \in A, b \in B\}$.

$A \cup B$ represents the union of the sets represented by A and B: $A \cup B$.

A^* represents the concatenation of the set represented by A with itself zero or more times: $A^* = \{\varepsilon\} \cup A \cup AA \cup AAA \cup AAAA \cup \ldots$

This just defines a recursive function definition for computing the meaning of a regular expression:

$$
\begin{aligned}
\text{language}(\emptyset) &= \{\,\} \\
\text{language}(\varepsilon) &= \{\varepsilon\} \\
\text{language}(AB) &= \{a \bullet b \mid a \in \text{language}(A), b \in \text{language}(B)\} \\
\text{language}(A \cup B) &= \text{language}(A) \cup \text{language}(B) \\
\text{language}(A^*) &= \{\varepsilon\} \cup \text{language}(A) \cup \text{language}(AA) \cup \ldots
\end{aligned}
$$

Examples of regular expressions

$\mathbf{001^*}$

$\mathbf{0^*1^*}$

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\mathbf{(0^*1^*)^*}$

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Examples of regular expressions

$\mathbf{001^*}$

Binary strings with “00” followed by any number of 1s.

$\mathbf{0^*1^*}$

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\mathbf{(0^*1^*)^*}$

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Examples of regular expressions

$\mathbf{001^*}$

Binary strings with “00” followed by any number of 1s.

$\mathbf{0^*1^*}$

Binary strings with any number of 0s followed by any number of 1s.

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\mathbf{(0^*1^*)^*}$

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Examples of regular expressions

$\mathbf{001^*}$

Binary strings with “00” followed by any number of 1s.

$\mathbf{0^*1^*}$

Binary strings with any number of 0s followed by any number of 1s.

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\{0000, 0010, 1000, 1010\}$

$\mathbf{(0^*1^*)^*}$

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Examples of regular expressions

$\mathbf{001^*}$

Binary strings with “00” followed by any number of 1s.

$\mathbf{0^*1^*}$

Binary strings with any number of 0s followed by any number of 1s.

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\{0000, 0010, 1000, 1010\}$

$\mathbf{(0^*1^*)^*}$

All binary strings.

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Examples of regular expressions

$\mathbf{001^*}$

Binary strings with "00" followed by any number of 1s.

$\mathbf{0^*1^*}$

Binary strings with any number of 0s followed by any number of 1s.

$\mathbf{(0 \cup 1)0(0 \cup 1)0}$

$\{0000, 0010, 1000, 1010\}$

$\mathbf{(0^*1^*)^*}$

All binary strings.

$\mathbf{(0 \cup 1)^*0110(0 \cup 1)^*}$

Binary strings that contain "0110".

Regular expressions in practice

Used to define the *tokens* in a programming language.

Legal variable names, keywords, etc.

Used in `grep`, a Unix program that searches for patterns in a set of files.

For example, `grep "311" *.md` searches for the string “311” in all Markdown files in the current directory.

Used in programs to process strings.

These slides are generated with the help of regular expressions :)

Summary

Use structural induction to prove properties of recursive structures.

Follows from ordinary induction but is easier to use.

As powerful as ordinary induction.

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.

A regular expression defines a set of strings over an alphabet $\boldsymbol{\Sigma}$.

$\emptyset$, ε, and $a \in \Sigma$ are regular expressions.

If A and B are regular expressions, then so are (AB), $(A \cup B)$, A^*.

Many practical applications, from `grep` to everyday programming.