Spring 2015

Lecture 19: Structural induction and regular expressions

- An *alphabet* **Σ** is any finite set of characters.

  e.g. $\Sigma = \{0,1\}$ or $\Sigma = \{A, B, C, \ldots X, Y, Z\}$ or

  $\Sigma =$

1		28	∟	95	_	153	Ö	186	║	219	█
2	☻	29	↔	96	`	154	Ü	187	╗	220	▄
3	♥	30	▲	97-122	a-z	155	¢	188	╝	221	▌
4	♦	31	▼	123	{	156	£	189	╜	222	▐
5	♣	32	(space)	124	|	157	¥	190	╛	223	▀
6	♠	33	!	125	}	158	₧	191	┐	224	α
7	●	34	"	126	~	159	ƒ	192	└	225	ß
8	◘	35	#	127	⌂	160	á	193	┴	226	Γ
9	○	36	$	128	Ç	161	í	194	┬	227	π
10	◙	37	%	129	ü	162	ó	195	├	228	Σ
11	♂	38	&	130	é	163	ú	196	─	229	σ

- The set **Σ*** of *strings* over the alphabet **Σ** is defined by
  - **Basis:** $\varepsilon \in \Sigma^*$ (ε is the empty string)
  - **Recursive:** if $w \in \Sigma^*$, $a \in \Sigma$, then $wa \in \Sigma^*$

Length:

len (ε) = 0;

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

Reversal:

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

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

$x \bullet (a_1 a_2 \cdots a_k)$

$= (x \bullet (a_1 \cdots a_{k-1}))\, a_k$

$= \cdots$

Concatenation:

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

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

$= x\, a_1 a_2 \cdots a_k$

How to prove $\forall\, x \in S,\, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the *Basis step*

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S,\, P(x)$

structural induction for strings

Let S be a set of strings over $\Sigma = \{a, b\}$ defined by

Basis: $a \in S$

Recursive:

If $w \in S$ then $wa \in S$ and $wba \in S$

If $u, v \in S$ then $uv \in S$

Claim: If $w \in S$ then w has more a's than b's.

$P(w)$ = "w has more a's than b's"

Base case: a has more a's than b's so $P(a)$ holds.

IH: Assume $P(w), P(u), P(v)$ for some $u, v, w \in S$

If $\#_a(w) > \#_b(w)$ then $\#_a(wa) > \#_b(wa) \Rightarrow P(wa)$

then $\#_a(wba) > \#_b(wba) \Rightarrow P(wba)$

cont $\rightarrow$

By IH $\rightarrow$ $\#_a(uv) = \#_a(u) + \#_a(v)$

$\#_b(uv) = \#_b(u) + \#_b(v)$

$\Rightarrow$

$\#_a(uv) > \#_b(uv)$

$\Rightarrow P(uv)$.

$\square$

proof continued?

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

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

Goal: $P(y)\ \forall y \in \Sigma^*$

Base case: $\text{len}(x \cdot \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon) \Rightarrow P(\varepsilon)$

$(\forall x \in \Sigma^*)$ — def of $\cdot$ — def of len

IH: Assume $P(w)$ for some $w \in \Sigma^*$.

IS: Fix $x \in \Sigma^*$.

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

def of $\cdot$ — def of len

$= 1 + \text{len}(x) + \text{len}(w)$ — $P(w)$

$= \text{len}(x) + \text{len}(wa)$ — def of len

$\Rightarrow P(wa)$

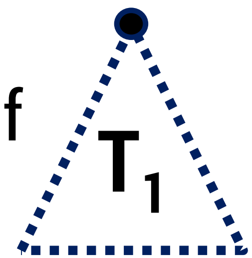
By struc. Ind $\forall y \in \Sigma^*\ P(y)$.

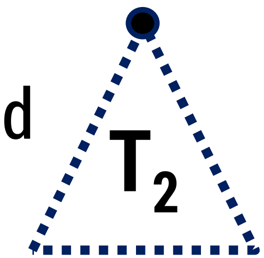
Length:

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

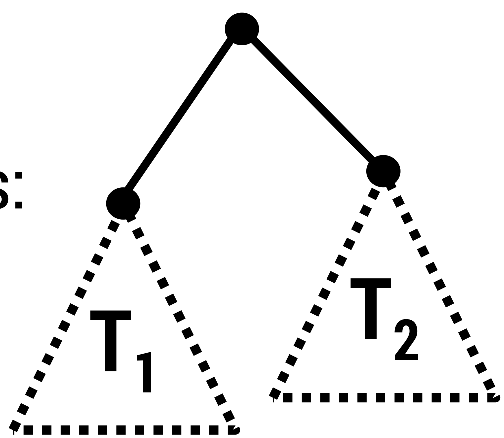
$\text{len}(wa) = 1 + \text{len}(w)$; for $w \in \Sigma^*, a \in \Sigma$

- Basis: • is a rooted binary tree
- Recursive step:

If 

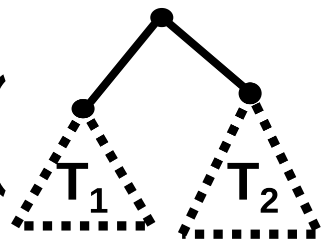
and 

are rooted binary trees,

then so is: 

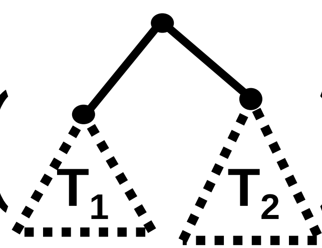
defining a function on rooted binary trees

- size(•) = 1

- size (

) $= 1 + \text{size}(T_1) + \text{size}(T_2)$

- height(•) = 0

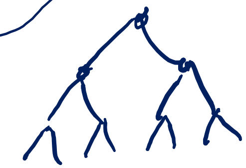
- height (

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

size vs. height

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

$P(T) =$ "$\text{size}(T) \le 2^{\text{height}(T)+1} - 1$"



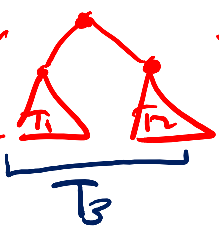
$15 \le 2^{3+1} - 1$

Base case:

$$\text{size}(\cdot) = 1 = 2^{0+1} - 1$$
$$= 2^{\text{height}(\cdot)+1} - 1 \Rightarrow P(\cdot)$$

$\Rightarrow P(T_3)$

IH: $P(T_1)$ and $P(T_2)$ for some RBTs T_1, T_2

Goal: $P($ 

$)$

$$\text{size}(T_3) = 1 + \text{size}(T_1) + \text{size}(T_2)$$
$$\overset{IH}{\le} 1 + (2^{h(T_1)+1} - 1) + (2^{h(T_2)+1} - 1)$$
$$= 2(2^{h(T_1)} + 2^{h(T_2)}) - 1 \le 2 \cdot 2 \cdot 2^{\max(h(T_1), h(T_2))} - 1$$

$$2 \cdot 2^{\max(h(T_1), h(T_2)) + 1} - 1$$
$$= 2 \cdot 2^{h(T_3)} - 1$$
$$= 2^{h(T_3)+1} - 1$$

Sets of strings that satisfy special properties are called languages.

$L \subseteq \Sigma^*$

Examples:

- English sentences
- Syntactically correct Java/C/C++ programs
- $\Sigma^* =$ All strings over alphabet Σ
- Palindromes over Σ
- Binary strings that don't have a 0 after a 1
- Legal variable names, keywords in Java/C/C++
- Binary strings with an equal # of 0's and 1's