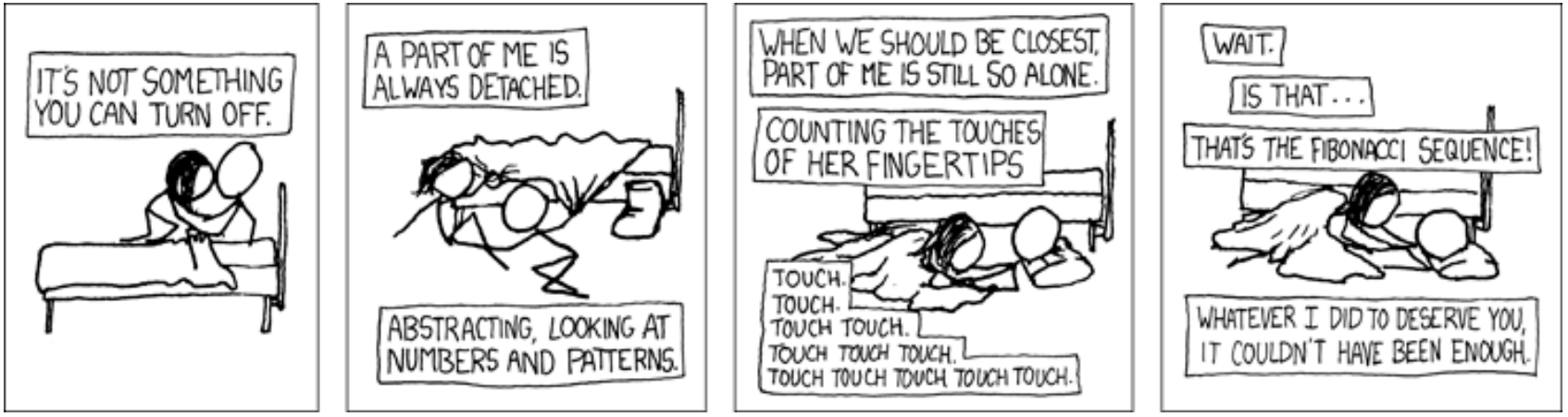
CSE 311: Foundations of Computing

Fall 2014

Lecture 17: Recursive Definitions and Structural Induction



Administrivia

Midterm

Monday, November 3rd, IN CLASS

Topics: Everything up to ordinary induction.

Review sessions:

~~Thursday, Oct 30: 4:30pm – 6:00pm (THO 101)~~

Saturday, Nov 1: 4:30pm – 7:00pm (EEB 125)

Sunday, Nov 2: 4:00pm – 6:00pm (EEB 125)

Strings

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

- 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 $wa \in \Sigma^*$

Palindromes

Palindromes are strings that are the same backwards and forwards (e.g. “abba”, “tht”, “neveroddoreven”).

Basis

- ε is a palindrome
- $\mathbf{a} \in \Sigma$ is a palindrome

Recursive Step

- If **p** is a palindrome and $\mathbf{a} \in \Sigma$, then **apa** is a palindrome.

All Binary Strings with no 1's before 0's…

Write a recursive definition for the set of binary strings in which all 0's appear before any 1's in the entire string.

Basis:

$\varepsilon \in S$

Recursive:

If $x \in S$, **then** $0x \in S$

If $x \in S$, **then** $x1 \in S$

Function Definitions on Recursively Defined Sets

Length:

$len(\varepsilon) = 0$

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

Reversal:

$\varepsilon^R = \varepsilon$

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

Concatenation:

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

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

Number of c's in a string:

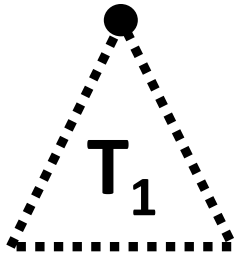
$\#_c(\varepsilon) = 0$

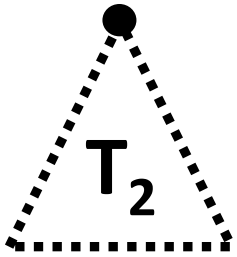
$\#_c(wc) = \#_c(w) + 1$ for $w \in \Sigma^*$

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

Rooted Binary Trees

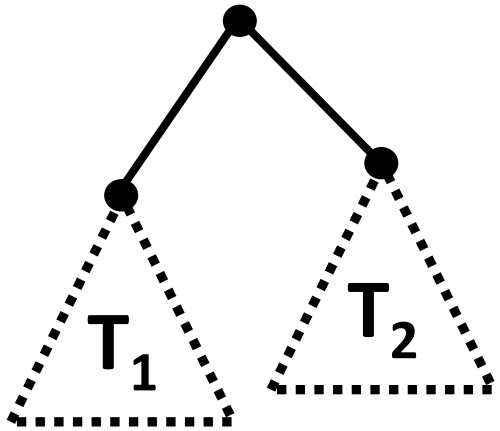
- **Basis:** • is a rooted binary tree
- **Recursive step:**

If 

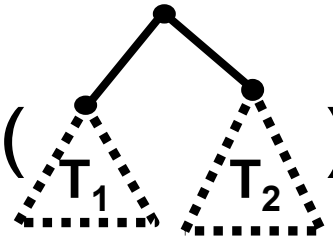
and 

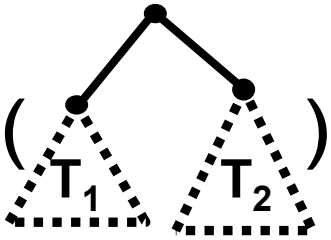
are rooted binary trees,

then so is:



Functions Defined on Rooted Binary Trees

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

- height(•) = 0
- height()=1 + max{height(T_1), height(T_2)}

Structural Induction

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

- **Base Case: Show that P(u) is true for all specific elements of $u \in S$ 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 constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis**
- **Conclude that $\forall x \in S, P(x)$**

Structural Induction vs. Ordinary Induction

Ordinary induction is a special case of structural induction:

Recursive definition of $\mathbb{N}$

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

Recursive Step: If $k \in \mathbb{N}$ then $k+1 \in \mathbb{N}$

Structural induction follows from ordinary induction:

Let Q(n) be true iff for all $x \in S$ that take n recursive steps to be constructed, P(x) is true.

Using Structural Induction

- **Let S be given by…**

  Basis: 6 ∈ S, 15 ∈ S

  Recursive: If x, y ∈ S, then x + y ∈ S.

Claim: Every element of S is divisible by 3.

Let P(x) be "3 | x". We go by structural induction on S.

Base Case:

6 = 2•3; So, 3 | 6.

15 = 5•3; So, 3 | 15.

Induction Hypothesis:

Suppose P(x), P(y) for some x, y ∈ S.

Induction Step:

We know, by the IH, that x = k•3 and y = j•3 for some k, j.

So, x + y = k•3 + j•3 = 3(k + j). So, 3 | x + y.

It follows that P(k) is true for all k ∈ S by structural induction.

Function Definitions on Recursively Defined Sets

Length:

$len(\varepsilon) = 0$

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

Reversal:

$\varepsilon^R = \varepsilon$

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

Concatenation:

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

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

Number of c's in a string:

$\#_c(\varepsilon) = 0$

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

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

Structural Induction for Strings

- **Let S be a set of strings over {a,b}, defined as follows…**

  Basis: $a \in S$

  Recursive:

  - If $w \in S$, then $wa \in S$ and $wba \in S$
  - If $u \in S$ and $v \in S$, then $uv \in S$

Claim: **If $w \in S$, then $\#_a(w) > \#_b(w)$**

Claim: If $w \in S$, then $\#_a(w) > \#_b(w)$

Basis: $a \in S$

Number of c's in a string:

$\#_c(\varepsilon) = 0$

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

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

Recursive:

- If $w \in S$, then $wa \in S$ and $wab \in S$
- If $u \in S$ and $v \in S$, then $uv \in S$

Proof:

Let P(w) be "$\#_a(w) > \#_b(w)$".

We go by structural induction on S.

Base Case:

$\#_a(a) = \#_a(\varepsilon a) = \#_a(\varepsilon) + 1 = 0 + 1 = 1$

$\#_b(a) = \#_b(\varepsilon a) = \#_b(\varepsilon) = 0$

Since $1 > 0$, it follows that $\#_a(a) > \#_b(a)$.

Claim: If $w \in S$, then $\#_a(w) > \#_b(w)$

Recursive:

If $w \in S$, then $wa \in S$ and $wab \in S$

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

Number of c's in a string:

$\#_c(\varepsilon) = 0$

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

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

Recursive Case 1 ($w \in S \rightarrow wa \in S$ and $wab \in S$):

Suppose $\#_a(w) > \#_b(w)$ for some $w \in S$.

We want to show $P(wa)$ and $P(wba)$.

Note that $\#_a(wa) = \#_a(w) + 1$ (def of $\#_a$)

$> \#_b(w) + 1$ (IH)

$> \#_b(w)$ (inequalities)

$= \#_b(wa)$ (def of $\#_b$)

Also, $\#_a(wab) = \#_a(wa)$ (def of $\#_a$)

$> \#_b(w) + 1$ (proof above)

$= \#_b(wa) + 1$

$= \#_b(wab)$ (def of $\#_c$)

Claim: If $w \in S$, then $\#_a(w) > \#_b(w)$

Recursive:

If $w \in S$, then $wa \in S$ and $wab \in S$

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

Number of c's in a string:

$\#_c(\varepsilon) = 0$

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

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

Recursive Case 2 ($u \in S$ and $v \in S \rightarrow uv \in S$):

Suppose P(u) and P(v) for some $u,v \in S$.

We want to show P(uv).

Note that $\#_a(uv) = \#_a(u) + \#_a(v)$ ($\#_a$ is additive; we're pretending we've proven this)

$> \#_b(u) + \#_b(v)$ (IH)

$= \#_b(uv)$ ($\#_b$ is additive)

So, the P(x) is true for all $x \in S$ by structural induction.

Claim: len(x • y) = len(x) + len(y) for all x,y ∈ Σ^*

Concatenation:

x • Ɛ = x for x ∈ Σ^*

x • wa = (x • w)a for x ∈ Σ^*, a ∈ Σ

Length:

len(Ɛ) = 0

len(wa) = 1 + len(w) for w ∈ Σ^*, a ∈ Σ

Let P(y) be "len(x • y) = len(x) + len(y) for all x ∈ Σ^*"

We prove P(y) for all y ∈ Σ* by structural induction.

Base Case:

len(x • Ɛ)	= len(x)	(def of concat.)
	= len(x) + 0	(adding 0)
	= len(x) + len(Ɛ)	(def of len(Ɛ))

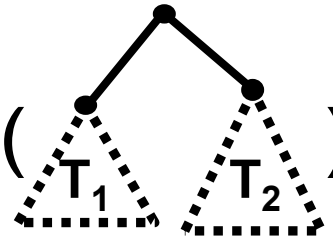
Induction Hypothesis:

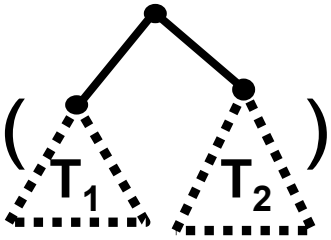
Suppose P(w) for some w ∈ Σ^*.

Induction Step:

len(x • wa)	= len((x • w)a)	(def of concat.)
	= 1 + len(x • w)	(def of len)
	= 1 + len(w) + len(x)	(IH)
	= len(wa) + len(x)	(def of len)

Functions Defined on Rooted Binary Trees

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

- height(•) = 0
- height()=1 + max{height(T_1), height(T_2)}