Mathematics for Computer Science

MIT 6.042J/18.062J

Recursive Definitions & Structural Induction

Recursive Definitions

Define something in terms of a simpler version of the same thing:

- Base case(s) that don't depend on anything else.
- Constructor case(s) that depend on simpler cases.

Matched Paren Strings, M

set of strings, $M \subseteq \{], [\}^*$

- Base: $\lambda \in M$, (the empty string)
- Constructor:

If s, t $\in$ M, then

[s]t $\in$ M

Matched Paren Strings M

strings in **M**

[]	s = λ	t = λ
[[]]	s = []	t = λ
[][]	s = λ	t = []
[[]][]	s = []	t = []
[[[]]]	s = [[]]	t = λ
⋮	⋮	⋮

not in M

strings starting with]

are not in M because

- λ does not start with]
- [s]t does not start with]

and everything in M arises in one of these two ways

Matched Paren Strings, M

set of strings, $M \subseteq \{], [\}^*$

- Base: $\lambda \in M$,
- Constructor: If s,t $\in$ M, then [s]t $\in$ M
- That's all

Extremal Clause

(Implicit part of definition)

Structural Induction

To prove P(x) holds for all x in recursively defined set R, prove

- P(b) for each base case $b \in R$
- P(c(x)) for each constructor, c, assuming ind. hyp. P(x)

Matched Paren Strings M

Lemma: Every s in M has the same number of]'s and ['s.

Proof by structural induction on the definition of M

Matched Paren Strings M

Lemma: Every s in M has the same number of]'s and ['s.

Let EQ ::= {strings with same number of] and [}

Lemma (restated): $M \subseteq EQ$

Structural Induction on M

Proof:

Ind. Hyp. P(s) ::= ($s \in EQ$)

Base case ($s = \lambda$):

λ has 0]'s and 0 ['s,

so P(λ) is true.

base case is OK

Structural Induction on M

Constructor step: s = [r]t

can assume P(r) and P(t)

#] in s	=	#] in r	+	#] in t	+ 1
#[in s	=	#[in r	+	#[in t	+ 1
so =		= by P(r)		= by P(t)	

so P(s) is true constrct case is OK

Structural Induction on M

so by struct. induct.

$$M \subseteq EQ$$

QED

The 18.01 Functions, F18

The set F18 of functions on $\mathbb{R}$:

$Id_{\mathbb{R}}$, constant functions, and sin x are in F18.

if f, g ∈ F18, then

- f + g, f · g, e^f, (the constant e)
- the inverse, $f^{(-1)}$, of f, and
- f ∘ g (the composition of f and g)

are in F18.

The 18.01 Functions, F18

Some functions in F18:

$-x = (-1) \cdot x$

$\sqrt{x} = (x^2)^{(-1)}$ ---inverse

$\cos x = (1 - (\sin x \cdot \sin x))^{1/2}$

$\ln x = (e^x)^{(-1)}$

The 18.01 Functions, F18

Lemma.

F18 is closed under taking derivatives:

if f ∈ F18, then f′ ∈ F18

Class Problem

Recursive function on M

Def. depth(s) for s ∈ M

depth(λ) ::= 0

depth([s]t) ::= max{1+d(s), d(t)}

k^n — recursive function on $\mathbb{N}$

expt(k, 0) ::= 1

expt(k, n+1) ::= k · expt(k,n)

--uses recursive def of $\mathbb{N}$:

- $0 \in \mathbb{N}$
- if $n \in \mathbb{N}$ then $n+1 \in \mathbb{N}$

Recursive Functions

summary:

f: Data → Values

f(b) def'd directly for base b

f(cnstr(x)) def'd using f(x), x

positive powers of two

$2 \in PP2$

if $x,y \in PP2$, then $x \cdot y \in PP2$

$2,\ 2 \cdot 2,\ 4 \cdot 2,\ 4 \cdot 4,\ 4 \cdot 8, \ldots$

$2 \quad 4 \quad 8 \quad 16 \quad 32 \ldots \in PP2$

loggy function on PP2

loggy(2) ::= 1

loggy($x \cdot y$) ::= x + loggy(y)
for $x,y \in PP2$

loggy(4) = loggy($2 \cdot 2$) = 2 + 1 = 3

loggy(8) = loggy($2 \cdot 4$) = 2 + loggy(4)
= 2 + 3 = 5

loggy(16) = loggy($8 \cdot 2$) = 8 + loggy(2)
= 8 + 1 = 9

loggy function on PP2

loggy(16) = loggy($8 \cdot 2$) = 9

WAIT A SEC!:

loggy(16) = loggy($2 \cdot 8$)
= 2 + loggy(8) = 2 + 5
= 7

ambiguous constructors

The Problem: more than one way to construct elements of PP2 from

cnstrct(x,y) = $x \cdot y$

16 = cnstrct(8,2) but also

16 = cnstrct(2,8)

ambiguous

Team Problems

Problems

1–3