Mathematics for Computer Science

MIT 6.042J/18.062J

Recursive Data

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.

Example Definition: set E

Define set $E \subseteq \mathbb{Z}$, recursively:

- Base case: $0 \in E$
- Constructor cases:

  If $n \in E$, then

  1. $n + 2 \in E$, if $n \geq 0$;
  2. $-n \in E$, if $n > 0$.

Example Definition: set E

1. $n \in E$ and $n \geq 0$, then $n + 2 \in E$:

   0, 0+2, (0+2)+2, ((0+2)+2) +2

   0, 2, 4, 6, …

2. $n \in E$ and $n > 0$, then $-n \in E$

   -2, -4, -6, …

all even numbers

Recursive Def: Extremal Clause

So, E contains the even integers

Anything Else? No!

Recursive Def: Extremal Clause

So, E contains the even integers

- $0 \in E$
- If $n \in E$ and $n \geq 0$, then $n+2 \in E$
- If $n \in E$ and $n > 0$, then $-n \in E$
- That's All!

Extremal Clause

(Implicit part of definition)

Example Definition: set E

So E is exactly the Even Integers

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 $[s]t \in M$

[]	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

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 $f, g \in$ F18, then

- $f + g$, $f \cdot g$, 2^f,
- the inverse, $f^{(-1)}$, of f, and
- $f \circ g$ (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

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

$\ln x = (2^{x \log e})^{(-1)}$

6.042J / 18.062J Mathematics for Computer Science
Spring 2015