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
      \[
      [s \text{]} t \in M
      \]

Strings in \( M \)

\[
\begin{align*}
[ & ] & s = \lambda & t = \lambda \\
[ ] & & s = [ ] & t = [ ] \\
[ ] & & s = \lambda & t = [ ] \\
[ [ ] ] & & s = [ ] & t = [ ] \\
[ [ ] ] & & s = [ ] & t = [ ] \\
[ [ [ ] ] ] & & s = [ [ ] ] & t = \lambda \\
& & & \\
& & &
\end{align*}
\]

Not in \( M \)

- Strings starting with \([ \) are not in \( M \) because
  - \( \lambda \) does not start with \([ \)
  - \([ s \text{]} t \) 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 \([s \text{]} t \in M \)

**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 $M$

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

Let $EQ ::= \{\text{strings with same number of } \}$

Lemma (restated): $M \subseteq EQ$

Construct 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 $s = \text{by } P(r) + \text{by } P(t)$
so $P(s)$ is true construct case is OK

so by struct. induct. $M \subseteq EQ$

QED
The 18.01 Functions, F18

The set $F_{18}$ of functions on $\mathbb{R}$:
- $\text{Id}_{\mathbb{R}}$, constant functions, and $\sin x$
  are in $F_{18}$.

if $f, g \in F_{18}$, then
- $f + g$, $\cdot g$, $e^f$ (the constant $e$)
- the inverse, $f^{-1}$, of $f$, and
- $f \circ g$ (the composition of $f$ and $g$)
  are in $F_{18}$.

Some functions in $F_{18}$:
- $-x = (-1) \cdot x$
- $\sqrt{x} = (x^2)^{-1}$ ---inverse
- $\cos x = (1 - (\sin x \cdot \sin x))^{1/2}$
- $\ln x = (e^x)^{-1}$

Lemma.

$F_{18}$ is closed under taking derivatives:
if $f \in F_{18}$, then $f' \in F_{18}$

Class Problem

Recursive function on $M$

Def. $\text{depth}(s)$ for $s \in M$
- $\text{depth}(\lambda) ::= 0$
- $\text{depth}(\,[s]^t):=\max{1+d(s), d(t)}$

Recursive Functions

summary:
- $f$: Data $\rightarrow$ Values
- $f(b)$ def'd directly for base $b$
- $f(\text{cnstr}(x))$ def'd using $f(x)$, $x$
positive powers of two

$2 \in \text{PP2}$

if $x, y \in \text{PP2}$, then $x \cdot y \in \text{PP2}$

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

$2, 4, 8, 16, 32 \ldots \in \text{PP2}$

loggy function on PP2

$loggy(2) ::= 1$

$loggy(x \cdot y) ::= x + loggy(y)$

for $x, y \in \text{PP2}$

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

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

$= 2 + 3 = 5$

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

$= 8 + 1 = 9$

loggy function on PP2

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

$\text{WAIT A SEC!}$

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

$= 2 + loggy(8) = 2 + 5$

$= 7$

ambiguous constructors

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

$\text{cnstrct}(x, y) = x \cdot y$

$16 = \text{cnstrct}(8, 2)$ but also

$16 = \text{cnstrct}(2, 8)$

ambiguous

Team Problems

Problems 1–3