Recursive Functions

To define a function, $f$, on a recursively defined set $R$, define

- $f(b)$ explicitly for each base case $b \in R$
- $f(c(x))$ for each constructor, $c$, in terms of $x$ and $f(x)$

Recursive function on $M$

Def. tree-depth($s$) for $s \in M$

- $td(\lambda) ::= 0$
- $td([s]t) ::= 1 + \max\{td(s), td(t)\}$

Recursive function on $\mathbb{N}$

- $expt(k, 0) ::= 1$
- $expt(k, n+1) ::= k \cdot expt(k, n)$

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

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

**Summary:**
- \( f : \text{Data} \rightarrow \text{Values} \)
- \( f(b) \) defined directly for base \( b \)
- \( f(\text{cnstr}(x)) \) defined using \( f(x) \), \( x \)

Length versus Depth

**Lemma:** \(|r| + 2 \leq 2^{td(r) + 1}\) for all \( r \in M \)

Proof by Structural Induction

**Base Case:** \([r = \lambda]\)

\[ |\lambda| + 2 = 0 + 2 = 2 = 2^{0+1} = 2^{td(\lambda)+1} \]

**OK!**

Size versus Depth

**Constructor Case:** \([r = [s]t]\)

By induction hypothesis:

\[ |s| + 2 \leq 2^{td(s)+1} \]

\[ |t| + 2 \leq 2^{td(t)+1} \]

\[ |r| + 2 = |[s]t| + 2 \quad \text{def. of } r \]

\[ = (|s| + |t| +2) + 2 \quad \text{def. of length} \]

\[ = (|s| + 2) + (|t| + 2) \]

\[ \leq 2^{td(s)+1} + 2^{td(t)+1} \quad \text{induction hyp.} \]

\[ \leq 2 \cdot \max(2^{td(s)}, 2^{td(t)}) + 2 \cdot \max(2^{td(s)}, 2^{td(t)}) + 1 \]

\[ = 2 \cdot 2^{\max(2^{td(s)}, 2^{td(t)})+1} \leq 2 \cdot 2^{td(r)} \quad \text{def. of } d(r) \]

\[ = 2^{td(r)+1} \quad \text{QED!} \]
positive powers of two

\[ 2 \in \text{PP2} \]

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

\[ 2, 2 \cdot 2, 4 \cdot 2, 4 \cdot 4, 4 \cdot 8, \ldots \]

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

log of \( \text{PP2} \)

\[ \log_2(2) ::= 1 \]

\[ \log_2(x \cdot y) ::= \log_2(x) + \log_2(y) \]

for \( x, y \in \text{PP2} \)

\[ \log_2(4) = \log_2(2 \cdot 2) = 1 + 1 = 2 \]

\[ \log_2(8) = \log_2(2 \cdot 4) = \log_2(2) + \log_2(4) = 1 + 2 = 3 \]

loggy function on \( \text{PP2} \)

\[ \loggy(2) ::= 1 \]

\[ \loggy(x \cdot y) ::= x + \loggy(y) \]

for \( x, y \in \text{PP2} \)

\[ \loggy(4) = \loggy(2 \cdot 2) = 2 + 1 = 3 \]

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

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

\[ \loggy(16) = \loggy(2 \cdot 8) = 9 \]

\[ \loggy(16) = \loggy(2 \cdot 8) = \]

\[ = 2 + \loggy(8) = 2 + 5 = 7 \]

WAIT A SEC!
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**

ambiguous recursive defs
problem to watch out for:
recursive function on datum, \( e \), is defined according to what constructor created \( e \).
If 2 or more ways to construct \( e \), then which definition to use?