

MGS Lambda Exercises 4

1. INTRODUCTION:

$$\frac{}{\text{leaf}_A : \text{Tree}_A} \quad \frac{a:A \quad l:\text{Tree}_A \quad r:\text{Tree}_A}{\text{node}(x, l, r) : \text{Tree}_A}$$

ELIMINATION: For every type X :

$$\frac{f: A \rightarrow \text{Tree}_A \rightarrow \text{Tree}_A \rightarrow X \rightarrow X \rightarrow X}{\text{rectree } f \ x_0 : \text{Tree}_A \rightarrow X}$$

REDUCTION:

$$\text{rectree } f \ x_0 \ \text{leaf} \rightsquigarrow x_0$$

$$\text{rectree } f \ x_0 \ \text{node}(a, l, r) \rightsquigarrow \\ f \ a \ l \ r \ (\text{rectree } f \ x_0 \ l) \ (\text{rectree } f \ x_0 \ r)$$

$$\text{mirror} : \text{Tree}_A \rightarrow \text{Tree}_A$$

$$\text{mirror} =$$

$$\text{rectree}$$

$$(\lambda a \ l \ r \ x_l \ x_r. \text{node}(a, x_r, x_l))$$

$$\text{leaf}$$

$treeList : Tree_A \rightarrow List_A$

$treeList =$

$recTree$

$(\lambda a \ l \ r \ x_l \ x_r. \text{append } x_l \ (a :: x_r))$
 nil

$append_A : List_A \rightarrow List_A \rightarrow List_A$

$append = \lambda t_1 \ t_2. recList (\lambda a \ l \ x. a :: x) t_2 t_1$

$listTree : List_{Nat} \rightarrow Tree_{Nat}$

$listTree = recList (\lambda n \ l \ x. treeInsert (eq \ n \ x) \ leaf$

$treeInsert_A : (A \rightarrow A \rightarrow Bool) \rightarrow A \rightarrow Tree_A \rightarrow Tree_A$

$treeInsert \ c \ v =$

$recTree$

$(\lambda a \ l \ r \ x_l \ x_r.$

$\text{if } c \ v \ a$

$\text{then } node(a, x_l, r)$

$\text{else } node(a, l, x_r))$

v

$leq : Nat \rightarrow Nat \rightarrow Bool$

$leq = \lambda n \ m. isZero (sub \ n \ m)$

$isZero : Nat \rightarrow Bool$

$isZero = rec (\lambda n \ t. false) true$

$\text{sub} : \text{Nat} \rightarrow \text{Nat} \rightarrow \text{Nat}$

$\text{sub} = \lambda n m. \text{rec} (\lambda i t. \text{pred } t) \ n \ m$

$\text{sortNats} : \text{List}_{\text{Nat}} \rightarrow \text{List}_{\text{Nat}}$

$\text{sortNats} = \lambda l. \text{treeList} (\text{listTree } l)$