

Solution Keys to Assignment 5

BU CAS CS520: Principles of Programming Languages, Fall 2002

Chiyan Chen

[Question 1]

There are two ways of implementing *append*:

1. $append = \Lambda T.\lambda l_1 : List(T).\lambda l_2 : List(T).l_1[List(T)] (l_2) Cons$
2. $append = \Lambda T.\lambda l_1 : List(T).\lambda l_2 : List(T).
 \Lambda X.\lambda nil : X.\lambda cons : T \rightarrow X \rightarrow X.l_1[X](l_2[X] (nil) cons) cons$

Also there are two ways of implementing *reverse*:

1. $reverse = \Lambda T.\lambda l : List(T).l[List(T)] (Nil[T])
 (\lambda x : T.\lambda l' : List(T).append[T] (l') (Cons[T] (x) Nil[T]))$
2. $reverse = \Lambda T.\lambda l : List(T).\Lambda X.\lambda n : T.\lambda cons : T \rightarrow X \rightarrow X.
 (l[X \rightarrow X] (\lambda x : X.x) (\lambda n' : T.\lambda k : X \rightarrow X.\lambda x : X.k (cons n' x))) n$

[Question 2]

Let $Gtree(T) = \forall X.X \rightarrow ((T \rightarrow X) \rightarrow X) \rightarrow X$.

Define the value constructors *E* and *B* as follows.

$$E = \Lambda T.\Lambda X.\lambda e : X.\lambda b : (T \rightarrow X) \rightarrow X.e$$

$$B = \Lambda T.\lambda f : T \rightarrow Gtree(T).\Lambda X.\lambda e : X.\lambda b : (T \rightarrow X) \rightarrow X.
 b (\lambda x : T.(f x)[X] e b)$$

Let us define a function unB as follows.

$$\begin{aligned}
unB = & \Lambda T. \lambda t : Gtree(T).fst[T \rightarrow Gtree(T)][Gtree(T)] \\
& (t[T \rightarrow Gtree(T) * Gtree(T)] \\
& pair[T \rightarrow Gtree(T)][Gtree(T)] (\lambda x : T.E[T] (E[T]) \\
& (\lambda k : T \rightarrow (T \rightarrow Gtree(T) * Gtree(T)). \\
& pair[T \rightarrow Gtree(T)][Gtree(T)] \\
& (\lambda x : T.snd[T \rightarrow Gtree(T)][Gtree(T)] (k x), \\
& B(\lambda x : T.snd[T \rightarrow Gtree(T)][Gtree(T)] (k x))))))
\end{aligned}$$

And finally $LeftGtree = \lambda t : Gtree(bool).(unB[bool] t) false$.