BU CAS CS 520 (Fall Semester, 2003)
Principles of Programming Languages

Assignment 5

Out: Friday, 07 November 2003
Due: Wednesday, 19 November 2003

Total: 140 points

Let us define $List(T)$ as $\forall X.X \to (T \to X \to X) \to X$. Then the two list constructors Nil and $Cons$ can be defined as follows.

$$\begin{array}{rcl} Nil & = & \Lambda X.\lambda nil : X.\lambda cons : T \to X \to X.nil \\ Cons(x:T)(xs:List(T)) & = & \Lambda X.\lambda nil : X.\lambda cons : T \to X \to X.cons(x)(xs[X](nil)(cons)) \end{array}$$

Exercise 1 *(40 points) Given the above list encoding, please implement the append and reverse functions on lists in System F. Notice that you may not use the fixed point operator in exercise.*

Exercise 2 *(60 points) Please encode the following datatype* `'a gtree` *and its associated constructors* `E` *and* `B` *in system F (30 points).*

```
datatype 'a gtree = E | B of ('a -> 'a gtree)
```

The type constructor `gtree` *can be used to form general tree types. For instance,* `bool gtree` *can be considered as a type for binary trees;* `B (fn (x:bool) => E)` *represents a binary tree t_1 whose left and right subtrees are empty;* `B (fn (x:bool) => if x then E else B (fn (x: bool) => E)` *represents a binary tree t_2 whose left subtree is empty and right subtree is t_1.*

Please implement the following function `leftGtree` *in system F (30 points).*

```
fun leftGtree E = E
  | leftGtree (B (f)) = f (false)
```

Note that you may not use fixed point operator in your implementation of `leftGtree`*.*

We can define a type erasure function $|\cdot|$ as follows, which translates a term in System F into an untyped λ-term.

$$|x| = x \qquad |\lambda x : T.t| = \lambda x.|t| \qquad |t_1(t_2)| = |t_1|(|t_2|) \qquad |\Lambda X.t| = |t| \qquad |t[T]| = |t|$$

Notice that for a value v in System F, $|v|$ may not necessarily be a value. Therefore, given a term t in System F, $|t| \to^* v_0$ does not necessarily imply that we have $t \to^* v$ for some value v such that $|v| = v_0$. To have this property, we can impose a restriction on **(T-Tabs)** by requiring that t be a value whenever the following rule is applied.

$$\frac{\vec{X}, X; \Gamma \vdash t : T \quad \vec{X} \vdash \Gamma\ [ctx]}{\vec{X}; \Gamma \vdash \Lambda X.t : \forall X.T}\ \textbf{(T-Tabs)}$$

This restriction is often called *value restriction*.

Exercise 3 *(40 points) Assume $\mathcal{D} :: \vec{X}; \Gamma \vdash t : T$ is derivable in System F with value restriction and $|t| \to u$. Show that $t \to^* t'$ for some term t' such that $|t'| = u$.*