Homework 14: The Simply Typed λ Calculus

Due:

When in doubt, fully parenthesize the λ calculus expression and the types you are working with.

When we discussed type checking in class, we started with a type, and then gave a proof that a term of the simply typed calculus had that type. It is also possible to use the type checking rules of the simply typed lambda calculus for type inference, as we discussed for ML. One can either guess the type and prove it, or use the rules to figure out the type. For example, to figure out the type of an application, figure out the type of the procedure and the argument, and then "follow the arrow". To figure out the type of a λ-expression, figure out the type of the body, and connect that to the type of the argument with an arrow.

1. Derive and prove the type of
$$\lambda f : \mathbf{o} \to \mathbf{o} \,.\, \lambda x : \mathbf{o} \,.\, f(x)$$
in the empty type context.

2. Derive and prove the type of
$$\lambda x : \mathbf{o} \to \mathbf{o} \to \mathbf{o} \,.\, \lambda y : \mathbf{o} \to \mathbf{o} \,.\, \lambda z : \mathbf{o} \,.\, xz(yz)$$
in the empty type context. (If the tree gets too bushy, use lemmas.)

3. Let ? be the type of the term in the previous problem; does the following term have a type (in the empty type context)?

   $(\lambda S :?. \lambda K : \mathbf{o} \to \mathbf{o} \to \mathbf{o} . S\ K\ K)(\lambda x : \mathbf{o} \to \mathbf{o} \to \mathbf{o} . \lambda y : \mathbf{o} \to \mathbf{o} . \lambda z : \mathbf{o} . xz(yz))(\lambda x : \mathbf{o} . \lambda y : \mathbf{o} . x)$

   If so state and prove the type. If not, clearly state why it does not check.

1 Extra Credit (Untyped λ calculus)

The following is about the untyped λ calculus. Read further in my handout, and also see Watt's section 5.1

4. Some fun with combinators. Let
$$\begin{aligned} I &\stackrel{\text{def}}{=} \lambda x \,.\, x \\ K &\stackrel{\text{def}}{=} \lambda x \,.\, \lambda y \,.\, x \\ S &\stackrel{\text{def}}{=} \lambda x \,.\, \lambda y \,.\, \lambda z \,.\, xz(yz) \\ B &\stackrel{\text{def}}{=} S(KS)K. \end{aligned}$$

   Using the equational rules, formally derive the normal forms (if any for the following). (Hint: it's convenient to use an editor to do this kind of work, as you can write out an expression, copy it, and apply the beta or eta rules without too much hand copying.) For example,

```
 (\f.(\y.z)f) ((\x.xx) (\x.xx))
==> <eta>
 (\y.z) ((\x.xx) (\x.xx))
==> <beta>
 z
```

   (a) $(SKK)x$
   (b) $Bxyz$
   (c) $(S(BBS)(KK))xyz$