Homework 1: Lambda Calculus

For your reference, we give the formalization of the λ-calculus in the appendix at the end of this document.

1. In the λ-calculus, does the following property hold?

   For any terms M and N, if $M \to N$, then $fv(M) = fv(N)$.

   If it holds, just answer yes; otherwise, please give a counterexample (that is, instantiate M and N such that $M \to N$ but $fv(M) \neq fv(N)$).

2. Reduce the following term to a normal form.

   $$(\lambda f.\, \lambda x.\, f\,(f\,x))\,(\lambda b.\, \lambda x.\, \lambda y.\, b\,y\,x)\,(\lambda z.\, \lambda w.\, z)$$

   Please do not skip steps.

3. Reduce the following term to a normal form.

   $$(\lambda f.\, f\,f)\,(\lambda f.\, \lambda x.\, f\,(f\,x))$$

   Please do not skip steps.

4. Reduce the following term to a normal form.

   $$(\lambda t.\, t\,(\lambda s.\, (\lambda t.\, t)\,s))(\lambda t.\, (\lambda s.\, s\,s\,s)(t\,(\lambda s.\, \lambda t.\, s)))(\lambda t.\, t)$$

   Please do not skip steps.

5. In this problem we add "let-bindings" to the λ-calculus.

   Syntax (the syntax for Values is unchanged):

   $$(\mathsf{Terms}) \quad M \quad ::= \quad \ldots \mid \mathsf{let}\ x = M\ \mathsf{in}\ M$$

   In the let-bindings $\mathsf{let}\ x = M_1\ \mathsf{in}\ M_2$, x scopes over M_2 but not M_1, so its free variables are:

   $$fv(\mathsf{let}\ x = M_1\ \mathsf{in}\ M_2) \stackrel{\text{def}}{=} fv(M_1) \cup (fv(M_2) - \{x\})$$

   New reduction rules (note v is a value):

   $$\frac{}{\mathsf{let}\ x = v\ \mathsf{in}\ M \to M[v/x]}$$

   $$\frac{M_1 \to M_1'}{\mathsf{let}\ x = M_1\ \mathsf{in}\ M_2 \to \mathsf{let}\ x = M_1'\ \mathsf{in}\ M_2}$$

(a) The first new rule uses substitution, but since we have extended the syntax of terms, the definition of substitution should be extended as well. Give the definition of the substitution $M[N/x]$ when M is in the form of let-bindings.

(b) Reduce the following term to a normal form.

$$\begin{array}{l} \textsf{let } c = \lambda n.\ \lambda m.\ \lambda f.\ \lambda x.\ n\ f\ (m\ f\ x) \textsf{ in} \\ \quad (\textsf{let } b = \lambda f.\ \lambda x.\ f\ (f\ x) \textsf{ in} \\ \quad\quad (\textsf{let } a = \lambda f.\ \lambda x.\ f\ x \textsf{ in} \\ \quad\quad\quad (c\ b\ a))) \end{array}$$

Please do not skip steps.

Appendix

A The λ-Calculus

Syntax:

$$\begin{array}{rrcl} (\mathsf{Terms}) & M & ::= & x \mid \lambda x.\, M \mid M\ N \\ (\mathsf{Values}) & v & ::= & \lambda x.\, M \end{array}$$

Free variables:

$$fv(x) \stackrel{\text{def}}{=} \{x\} \qquad fv(M\,N) \stackrel{\text{def}}{=} fv(M) \cup fv(N) \qquad fv(\lambda x.\,M) \stackrel{\text{def}}{=} fv(M) - \{x\}$$

α-equivalence:

$$\lambda x.\,M = \lambda y.\,M[y/x] \qquad \text{where } y \text{ fresh}$$

Substitution:

$$\begin{aligned} & x[N/x] \stackrel{\text{def}}{=} N \\ & y[N/x] \stackrel{\text{def}}{=} y \\ & (M\,N)[N'/x] \stackrel{\text{def}}{=} (M[N'/x])\,(N[N'/x]) \\ & (\lambda x.\,M)[N/x] \stackrel{\text{def}}{=} \lambda x.\,M \\ & (\lambda y.\,M)[N/x] \stackrel{\text{def}}{=} \lambda y.\,(M[N/x]), \quad \text{where } y \notin fv(N) \\ & (\lambda y.\,M)[N/x] \stackrel{\text{def}}{=} \lambda z.\,(M[z/y])[N/x], \quad \text{where } y \in fv(N) \text{ and } z \text{ fresh} \end{aligned}$$

Reduction rules:

$$\frac{}{(\lambda x.\ M)\ N \ \rightarrow\ M[N/x]} \qquad \frac{M \rightarrow M'}{\lambda x.\ M \ \rightarrow\ \lambda x.\ M'}$$

$$\frac{M \rightarrow M'}{M\ N \ \rightarrow\ M'\ N} \qquad \frac{N \rightarrow N'}{M\ N \ \rightarrow\ M\ N'}$$

Zero-or-more steps:

$$\begin{array}{lll} M \rightarrow^0 M' & \text{iff} & M = M' \\ M \rightarrow^{k+1} M' & \text{iff} & \exists M''.\ M \rightarrow M'' \wedge M'' \rightarrow^k M' \\ M \rightarrow^* M' & \text{iff} & \exists k.\ M \rightarrow^k M' \end{array}$$

Normal form: a term containing no redex.
Closed term: a term containing no free variables.