Assignment on Lambda Calculus

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

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

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

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

2. In the untyped λ-calculus, the *span* of a term is the minimal number of variables needed to write a term. It turns out that β-reduction can increase the span of a term. To show this, find a closed term M such that $M \to^* \lambda x.\ \lambda y.\ M\ (x\ y)$.

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

   Syntax (the syntax for Values is unchanged):

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

   New reduction rules (note v is a value):

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

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

   (a) The (LETV) 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.

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

   You can choose any reduction strategy. Please do not skip steps.

4. In the untyped λ-calculus, we have the following encoding of pairs:

$$\begin{array}{lcl} \mathsf{mkPair} & \stackrel{\text{def}}{=} & \lambda x.\ \lambda y.\ \lambda z.\ z\,x\,y \\ \mathsf{fst} & \stackrel{\text{def}}{=} & \lambda p.\ p\,(\lambda x.\ \lambda y.\ x) \\ \mathsf{snd} & \stackrel{\text{def}}{=} & \lambda p.\ p\,(\lambda x.\ \lambda y.\ y) \end{array}$$

We would expect a correct encoding to show $\mathsf{fst}\ (\mathsf{mkPair}\ z\ z)$ reduces to z. But the following sequence of steps allegedly shows that $\mathsf{fst}\ (\mathsf{mkPair}\ z\ z)$ reduces to True:

$$\begin{array}{ll} & (\lambda p.\ p\,(\lambda x.\ \lambda y.\ x))\ ((\lambda x.\ \lambda y.\ \lambda z.\ z\,x\,y)\ z\ z) \\ \to & (\lambda p.\ p\,(\lambda x.\ \lambda y.\ x))\ ((\lambda y.\ \lambda z.\ z\,z\,y)\ z) \\ \to & (\lambda p.\ p\,(\lambda x.\ \lambda y.\ x))\ (\lambda z.\ z\,z\,z) \\ \to & (\lambda z.\ z\,z\,z)\ (\lambda x.\ \lambda y.\ x) \\ \to & (\lambda x.\ \lambda y.\ x)\ (\lambda x.\ \lambda y.\ x)\ (\lambda x.\ \lambda y.\ x) \\ \to & (\lambda y.\ (\lambda x.\ \lambda y.\ x))\ (\lambda x.\ \lambda y.\ x) \\ \to & \lambda x.\ \lambda y.\ x \end{array}$$

(a) The sequence of steps is wrong. Find the wrong step(s).

(b) Show a correct sequence of steps that produces z but is otherwise very similar to the sequence of steps shown above.

(c) Extend the encoding to include a swap function. Given an encoding of the pair (v_1, v_2) as input, swap should return an encoding of the pair (v_2, v_1). You should write down a normal form, without using abbreviations.

Appendix

A The Untyped λ-Calculus

Syntax:

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

Reduction rules:

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

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

Substitution:

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

Free variables:

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

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.