Computation Answers 7: Lambda Calculus Part 1

1. (**Free and bound variables.**)

   (a) i. The binding occurrences of variables in this λ-term are:

   $$(\lambda\ \textcircled{x}\textcircled{y}\ .\ y\ (\lambda\ \textcircled{x}.\ x\ y)\ z)(x\ (\lambda\ \textcircled{z}\textcircled{x}.\ x\ z\ y))$$

   ii. The bound occurrences of variables in this λ-term are:

   $$(\lambda\ x\ y\ .\ \textcircled{y}(\lambda\ x\ .\ \textcircled{x}\textcircled{y})\ z)(x\ (\lambda\ z\ x\ .\ \textcircled{x}\textcircled{z}y))$$

   iii. The free occurrences of variables in this λ-term are:

   $$(\lambda\ x\ y\ .\ y\ (\lambda\ x\ .\ x\ y)\ \textcircled{z})(\textcircled{x}(\lambda\ z\ x\ .\ x\ z\ \textcircled{y}))$$

   (b) i. For $(\lambda x.\ xy)(x\lambda y.\ yx)(\lambda yz.\ zy)$:
   $FV = \{y, x\}$

   ii. For $(\lambda z.\ z(\lambda y.\ yzx)y)(\lambda xz.\ (\lambda\ y.\ zxy)x)$:
   $FV = \{x, y\}$

2. (**α-Equivalence.**)

   (a) The λ-terms which are α-equivalent to $(\lambda xy.\, y(\lambda x.\, xy)z)$ are iii and vii

   (b) If we can name our variables anything we want, then there are an infinite number of λ-terms which are α-equivalent to $(\lambda y.(\lambda x.\, xy)zxy)$. Here are three of them:

   i. $(\lambda y.(\lambda b.\, by)zxy)$
   ii. $(\lambda y.(\lambda c.\, cy)zxy)$
   iii. $(\lambda y.(\lambda d.\, dy)zxy)$

   Of course, we can also pick more interestingly α-equivalent terms:

   i. $(\lambda a.(\lambda x.\, xa)zxa)$
   ii. $(\lambda a.(\lambda y.\, ya)zxa)$
   iii. $(\lambda d.(\lambda z.\, zd)zxd)$

   (c) For each of the λ-terms I gave for part b, the FV set is $\{x, z\}$. Different α-equivalent λ-terms always have the same FV set. This is the point of α-equivalence. If I give you some code to use in your project, then you don't care what I call my local variables – there's no way that will affect the rest of your program. You care very much what global variables my code uses, since that will change the behaviour of your overall program.

3. (**Expression substitution.**)
   The results of the substitutions are as follows:

   (a) $(xy)[z/x] = zy$
   (b) $(xy)[\lambda x.\ xx/x] = (\lambda x.\ xx)y$
   (c) $(\lambda x.\ xy)[z/y] = \lambda x.\ (xy)[x/x][z/y] = \lambda x.\ xz$
   (d) $(\lambda x.\ xy)[z/x] = \lambda x.\ xy$

(e) $(\lambda x.\ xy)[x/y] = \lambda z.\ zx$

(f) $(\lambda x.\ xx)[\lambda x.\ xx/x] = \lambda x.\ xx$

(g) $(\lambda x.\ xy)[\lambda x.\ xy/y] = \lambda z.\ z(\lambda x.\ xy)$

(h) $(\lambda x.\ xy)[x(\lambda x.\ xy)/y] = \lambda z.\ z(x(\lambda x.\ xy))$

4. (β**-reduction.**)

(a) i.

$$\frac{}{(\lambda x. x)y \to y}$$

ii.

$$\frac{}{(\lambda x. \lambda y. xy)y \to \lambda z. yz}$$

Notice that in order to apply the function $(\lambda x. \lambda y. xy)$ to the argument y, we must perform the substitution $(\lambda y. xy)[y/x]$. If you look at the rules for substitution on slide 17 in your lecture notes, it is the fourth rule we must use. In order to perform this substitution we must avoid accidentally "capturing" the variable name y inside the function. We must therefore rename the argument of the function to some name that is not being used - here we choose the name z. So the substitution $(\lambda y. xy)[y/x]$ is equivalent to the substitution $(\lambda z. xz)[y/x]$, which gives us $(\lambda z. yz)$

iii.

$$\frac{}{(\lambda x. \lambda y. xy)z \to \lambda y. zy}$$

iv.

$$\frac{\dfrac{\dfrac{}{(\lambda x. x)y \to y}}{x((\lambda x. x)y) \to xy}}{\lambda x. x((\lambda x. x)y) \to \lambda x. xy}$$

(b) Let $M = (\lambda x. xx)((\lambda x. x)(\lambda x. xx))$
Let $N = ((\lambda x. x)(\lambda x. xx))(\lambda x. xx)$

(c) Both M and N can reduce in a single step to $(\lambda x. xx)(\lambda x. xx)$. It is also possible to reduce M as follows:

$$\begin{aligned}
&(\lambda x. xx)((\lambda x. x)(\lambda x. xx)) \\
&\to ((\lambda x. x)(\lambda x. xx))((\lambda x. x)(\lambda x. xx)) \\
&\to (\lambda x. xx)((\lambda x. x)(\lambda x. xx)) = M
\end{aligned}$$

This means that it is possible to evaluate M for an infinite number of steps, without ever producing an λ-term that can be reduced to from N. However, for any λ-term O such that $M \twoheadrightarrow O$, it is always the case that $O \twoheadrightarrow (\lambda x. xx)(\lambda x. xx)$.

(d)

$$\begin{aligned}
&(\lambda x. xxx)(\lambda x. xxx) \\
&\to (\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx) \\
&\to (\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx) \\
&\to (\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx)(\lambda x. xxx)
\end{aligned}$$

This function is self-replicating. With every β-reduction, it gets larger, and will never reduce to a normal form.

5. (β**-normal-forms.**)

(a) $(\lambda x.\ x)y$ has normal form y.

(b) $y(\lambda x.\ x)$ is already in normal form.

(c) $(\lambda x.\ x)(\lambda y.\ y)$ has normal form $\lambda y.\ y$.

(d) $(\lambda x.\ xx)(\lambda x.\ xx)$ has no normal form.

(e) $(\lambda x.\ xx)(\lambda x.\ x)$ has normal form $(\lambda x.\ x)$.

(f) $(\lambda x.\ x)(\lambda x.\ xx)$ has normal form $(\lambda x.\ xx)$.