PLC ASSIGNMENT-4

SIDDHANT CHAUDHARY

1. Let exp $:= \lambda pq.pq$. Prove that for all $m \geq 0$ and $n \geq 1$,

$$\exp[n][m] \to [m^n]$$

Hint: Prove the following claims in order:

(1) For $k, l \geq 0$, $(\lambda z.x^k z)^l y \to x^{kl} y$.
(2) For $m \geq 0$, $n \geq 1$, $(\lambda gy.g^m y)^n x \to (\lambda y.x^{m^n} y)$.
(3) From the above, show that for all $m \geq 0$ and $n \geq 1$, $[n][m] \to [m^n]$.
(4) Conclude that $\exp[n][m] \to [m^n]$.

Solution. Let us prove the claim as per the four steps given above.

(1) We will prove this step by induction on l. For the base case, suppose $l = 0$. Then, observe that

$$(\lambda z.x^k z)^l y = y = x^{k \cdot 0} y$$

and hence the base case is trivial. Next, suppose the statement is true for some l. Let $M = x^k y$. Then, observe that

$$\begin{aligned}
(\lambda z.x^k z)^{l+1} y &= (\lambda z.x^k z)^l ((\lambda z.x^k z) y) \\
&\to (\lambda z.x^k z)^l (x^k y) \\
&= (\lambda z.x^k z)^l M \\
&\to x^{kl} M && \text{(Induction hypothesis)} \\
&= x^{kl}(x^k y) \\
&= x^{kl+k} y \\
&= x^{k(l+1)} y
\end{aligned}$$

and this completes the inductive proof.

(2) We will prove this step by induction on n. For the base case, suppose $n = 1$. Then,

$$\begin{aligned}
(\lambda gy.g^m y)^1 x &\to \lambda y.x^m y \\
&= \lambda y.x^{m^1} y
\end{aligned}$$

Date: April 5, 2021.

and hence the base case is true. Now, suppose the statement is true for some $n \geq 1$. Let $M = \lambda y.x^m y$. Then, observe that

$$\begin{aligned}
(\lambda gy.g^m y)^{n+1} x &= (\lambda gy.g^m y)^n ((\lambda gy.g^m y)x) \\
&\to (\lambda gy.g^m y)^n (\lambda y.x^m y) \\
&\to (\lambda gy.g^m y)^n M \\
&\to (\lambda y.M^{m^n} y) && \text{(Induction Hypothesis)} \\
&\to \lambda y.(\lambda y.x^m y)^{m^n} y \\
&\to \lambda y.x^{m \cdot m^n} y && \text{(By part (1))} \\
&= \lambda y.m^{n+1} y
\end{aligned}$$

and this completes the inductive proof.

(3) Let $m \geq 0$ and $n \geq 1$. We use the encodings $[n] = \lambda fx.f^n x$ and $[m] = \lambda hy.h^m y$. Then, we have the following.

$$\begin{aligned}
[n][m] &= (\lambda fx.f^n x)(\lambda hy.h^m y) \\
&\to \lambda x.(\lambda hy.h^m y)^n x \\
&\to \lambda x.\lambda y.x^{m^n} y && \text{(By part (2))} \\
&= \lambda xy.x^{m^n} y \\
&= [m^n]
\end{aligned}$$

and hence the claim is proven.

(4) Let $m \geq 0$ and $n \geq 1$. Then, observe that

$$\begin{aligned}
\exp[n][m] &= (\lambda pq.pq)[n][m] \\
&\to (\lambda q.[n]q)[m] \\
&\to [n][m] \\
&\to [m^n]
\end{aligned}$$

where in the last step we have used step (3). This completes the solution to the problem.

■

2. What is the normal form of $[5](\exp[2])[2]$? What is the size (number of applications) of the normal form?

Solution. We will be using the fact that normal forms are unique, and hence we can β-reduce in any order. First, we see the following.

$$\begin{aligned}
[5](\exp[2])[2] &= [5]((\lambda pq.pq)[2])[2] \\
&\to [5](\lambda q.[2]q)[2]
\end{aligned}$$

Now, using the encoding $[5] = \lambda fx.f^5 x$, we get

$$\begin{aligned}
[5](\lambda q.[2]q)[2] &= (\lambda fx.f^5 x)(\lambda q.[2]q)[2] \\
&\to (\lambda x.(\lambda q.[2]q)^5 x)[2]
\end{aligned}$$

Now, we will use part (1) of problem **1.** We will use $k = 1$ and $l = 5$. Using it, we get

$$(\lambda q.[2]q)^5 x \to [2]^5 x$$

So, we see that

$$
\begin{aligned}
(\lambda x.(\lambda q.[2]q)^5 x)[2] &\to (\lambda x.[2]^5 x)[2] \\
&\to [2]^5[2] \\
&\to [2]^4([2][2]) \\
&\to [2]^4([2^2]) \qquad \text{(By part (3) of 1.)} \\
&\to [2]^3([2][2^2]) \\
&\to [2]^3([(2^2)^2]) \qquad \text{(By part (3) of 1.)} \\
&\to [2]^2([2][2^{2^2}]) \\
&\to [2]^2([(2^{2^2})^2]) = [2]^2[2^{2^3}] \\
&\quad \vdots \\
&\to [2^{2^5}] \\
&= \lambda f x.f^{2^{2^5}} x
\end{aligned}
$$

So, the above is the normal form for the given expression. The size (number of applications) in the normal form are 2^{2^5}. ■

3. This question has two parts.
(a) Find a lambda-expression F such that for all M, $FM = F$.

Solution. We claim that the required expression F is given by

$$F = (\lambda f x.ff)(\lambda f x.ff)$$

Let M be any lambda-expression. Then, we have the following.

$$
\begin{aligned}
FM &= (\lambda f x.ff)(\lambda f x.ff)M \\
&\xrightarrow{*} (\lambda x.(\lambda f x.ff)(\lambda f x.ff))M \\
&\xrightarrow{*} (\lambda f x.ff)(\lambda f x.ff) \\
&= F
\end{aligned}
$$

and hence it follows that $FM =_\beta F$, which is what we wanted to prove. ■

(b) Find a lambda-expression F such that for all M, $FM = MF$.

Solution. We claim that the required expression F is given by

$$F = (\lambda f x.xff)(\lambda f x.xff)$$

Let M be any expression. First, observe that

$$
\begin{aligned}
FM &= (\lambda f x.xff)(\lambda f x.xff)M \\
&\xrightarrow{*} (\lambda x.x(\lambda f x.xff)(\lambda f x.xff))M \\
&\xrightarrow{*} M(\lambda f x.xff)(\lambda f x.xff) \\
&= MF
\end{aligned}
$$

and hence we see that $FM =_\beta MF$, which is what we wanted to show. This completes the proof. ■

4. Prove that every expression in normal form M is of the form $\lambda x_1 \cdots \lambda x_n.yM_1M_2\cdots M_l$, where y is a variable and $M_1, ..., M_l$ are themselves in normal form.

Solution. We will prove this by induction on the length of the expression. For the base case, suppose the length of an expression M in normal form is 1. Clearly, $M = y$ for some variable y. In this case, we have

$$M = \lambda x_1 \cdots \lambda x_n.yM_1M_2\cdots M_l$$

with $n = l = 0$. So the base case is true.

Next, suppose the given statement is true for all expressions in normal form of length atmost n, where $n \in \mathbb{N}$. Let M be an expression in normal form of length $n+1$. There are two possible cases.

(1) In the first case, M is not an expression of the form $\lambda p.Q$. Since the length of M is greater than 1, M must then be of the form

$$M = A_1A_2...A_k$$

where each A_i for $1 \leq i \leq k$ is a lambda-expression, and such that the length of each M_i is strictly less than $n+1$ (which is equal to the length of M). Because of this, we see that $k \geq 2$. Since M is in normal form, each A_i must be in normal form as well. By induction hypothesis, we see that

$$A_1 = \lambda x_1 \cdots \lambda x_n yX_1X_2\cdots X_l$$

for some $n \geq 0$ and $l \geq 0$, and where y is a variable. Again, because M is in normal form and $k \geq 2$, we see that $n = 0$ (otherwise M will be β-reducible to some expression), i.e

$$A_1 = yX_1X_2\cdots X_l$$

So, we get

$$M = yX_1X_2...X_lA_2...A_k$$

which implies that M is of the form $\lambda x_1 \cdots \lambda x_n.yM_1M_2\cdots M_l$ (here $n = 0$).

(2) In the second case, M is of the form $\lambda p.Q$. So, suppose

$$M = \lambda x_1 \cdots \lambda x_n.A_1A_2...A_k$$

where each A_i for $1 \leq i \leq k$ is a lambda-expression, and $n, k \geq 1$. Note that, in this case because $n \geq 1$, the length of every A_i is less than $n+1$ (which is the length of M). Also note that, because M is in normal form, each A_i is in normal form as well. So, apply the inductive hypothesis to A_1, to obtain

$$A_1 = \lambda p_1 \cdots \lambda p_r.yM_1M_2...M_s$$

where $r, s \geq 0$, y is a variable and each M_i is in normal form. So, we get

$$M = \lambda x_1 \cdots \lambda x_n\lambda p_1 \cdots \lambda p_r.yM_1M_2...M_sA_2...A_k$$

and hence M is of the given form.

So, it follows that the given property is true for all expressions in normal form of length $n+1$, and hence the inductive proof is complete. ■

5. Find an encoding for the predecessor function in lambda calculus. The predecessor function is given by: $pred(0) = 0$ and $pred(n+1) = n$.

Solution. First, consider the function $pred' : \mathbb{N}^2 \to \mathbb{N}$ defined as follows.

$$pred'(0, m) = Z(m) = 0$$
$$pred'(n+1, m) = \pi_1^3(n, pred'(n, m), m) = n$$

and hence it is clear that $pred'$ is obtained via primitive recursion from the zero function Z and the projection function π_1^3. Then, observe that

$$pred(n) = pred'(n, Z(n))$$

for all $n \in \mathbb{N}$. So, $pred$ is defined by composing $pred'$ with $h_1 = \text{id}$ (identity function) and Z. First, we find the encoding $[pred']$.

To find $[pred']$, we will use the general primitive recursion encoding scheme which was discussed in class. We have

$$[pred'] = \lambda x x_1.[snd](x[Step][Init]) \tag{0.1}$$

where the following definitions are used:

$$[pair] = \lambda abc.cab \tag{0.2}$$
$$[fst] = \lambda p.p(\lambda de.d) \tag{0.3}$$
$$[snd] = \lambda p.p(\lambda fg.g) \tag{0.4}$$
$$[Init] = [pair][0]([Z]x_1) \to [pair][0][0] \tag{0.5}$$
$$[Step] = \lambda y.[pair]\ ([succ]([fst]y))\ ([\pi_1^3]\ ([fst]y)\ ([snd]y)\ x_1) \tag{0.6}$$

■

Now, the encodings $[Z]$ and $[\pi_1^3]$ are straightforward to find:

$$[Z] = \lambda s.[0] = \lambda s.(\lambda fx.x) \tag{0.7}$$
$$[\pi_1^3] = \lambda uvw.u \tag{0.8}$$

Using all this information, the encoding $[pred']$ can be found by using equation (0.1).

Now, as we remarked before, $pred$ is defined by composing $pred'$ with $h_1 = id$ (identity function) and Z, i.e

$$pred(n) = pred'(n, Z(n))$$

for all $n \in \mathbb{N}$. Now, the encoding [id] is easy to find.

$$[\text{id}] = \lambda x.x$$

So, the encoding $[pred]$ is as follows.

$$[pred] = \lambda x_1.[pred']\ ([\text{id}]x_1)\ ([Z]x_1)$$

Since $[pred']$, [id] and $[Z]$ are all known to us, we have found the encoding $[pred]$.

6. Find an encoding for the *Pow* function in lambda calculus. It is given by:

$$Pow(m,n) = \begin{cases} \text{true} & \text{if } \exists k \ : \ m^k = n \\ \text{false} & \text{otherwise} \end{cases}$$

Solution. Throughout, we assume that $m \geq 1$ and $n \geq 1$. First, we list the encodings that we will use to solve this problem.

The first encoding is $[subtr]$, i.e the subtraction function, which is defined as follows:

$$subtr(m,n) = \begin{cases} 0 & , \quad m \leq n \\ m - n & , \quad \text{otherwise} \end{cases}$$

In one of the lectures, it was shown in class that *subtr* is a primitive recursive function, and hence there is a λ-expression for *subtr*. We assume that it is $[subtr]$.

The next encoding we will use is the exponential function *exp*, defined as

$$exp(k,m) = m^k$$

In problem **1.**, we have already encoded this, and we assume that the encoding is $[exp]$.

Next, define the function $subtrexp : \mathbb{N}^3 \to \mathbb{N}$ by

$$subtrexp(k,n,m) = n - m^k$$

Clearly, we see that

$$subtrexp(k,n,m) = subtr(\pi_2^3(k,n,m), exp(\pi_1^3(k,n,m), \pi_3^3(k,n,m)))$$

Note that the functions *subtr*, π_2^3 and $exp \circ (\pi_1^3, \pi_3^3)$ are all primitive recursive. So, it follows that $subtr \circ (\pi_2^3, exp \circ (\pi_1^3, \pi_3^3))$ is also primitive recursive, and hence *subtrexp* is primitive recursive. Moreover, the λ-expression for *subtrexp* is given below.

$$[subtrexp] = \lambda knm.[subtr] \ ([\pi_2^3] \ k \ n \ m) \ ([exp] \ ([\pi_1^3] \ k \ n \ m) \ ([\pi_3^3] \ k \ n \ m)) \tag{0.9}$$

We will also use the standard encodings for $[true]$ and $[false]$, and the test $[iszero]$ given by

$$[iszero] = \lambda x.x(\lambda z.[false])[true]$$

Now, consider the function $mink : \mathbb{N}^2 \to \mathbb{N}$ defined as follows.

$$mink(m,n) = \text{smallest non-negative integer } k \text{ such that } subtrexp(k,n,m) = 0$$

In other words, $mink(m,n)$ is the smallest non-negative integer k such that $n \leq m^k$. One immediately recognizes that *mink* is defined by $\mu-$recursion from the function *subtrexp*, i.e

$$mink(m,n) = \mu i(subtrexp(i,m,n) = 0)$$

As covered in class, we need to find the encoding of this μ-recursion. So, first define

$$W = \lambda y. \ if \ ([iszero]([subtrexp] \ y \ m \ n)) \ then \ (\lambda w.y) \ else \ (\lambda w.w([succ]y)w)$$

Then, the encoding for *mink* is the following.

$$[mink] = \lambda mn.W \ [0] \ W$$

and the working of this was proven in one of the lectures.

Having found $[mink]$, we can now find the encoding of *Pow*. This is straightforward: to compute $Pow(m,n)$, we first compute $k = mink(m,n)$, and then we check whether $subtr(exp(k,m),n) = 0$. We already know that $n \leq m^k$. So, $subtr(exp(k,m),n) = m^k - n$ will be zero if and only if $m^k = n$. So, the encoding of *Pow* is as follows.

$$[Pow] = \lambda mn. \ if \ ([iszero]([subtr] \ ([exp] \ ([mink] \ m \ n) \ m) \ n)) \ then \ [true] \ else \ [false]$$

So, the required encoding $[Pow]$ has been found. ∎

7. In this problem, we will find combinators that satisfy the given behaviors.

(a) $\mathbf{I}$ such that $\mathbf{I}x \to x$.

Solution. Consider the combinator $\mathbf{I} = \mathbf{SKK}$. We then have

$$\begin{aligned}\mathbf{I}x &= \mathbf{SKK}x \\ &\to \mathbf{K}x(\mathbf{K}x) \\ &\to x\end{aligned}$$

∎

(b) $\mathbf{T}$ such that $\mathbf{T}xy \to yx$.

Solution. Here we will use the results of problem 8. To find the combinator $\mathbf{T}$, we will translate the lambda-expression

$$\lambda xy.yx$$

to its corresponding CL-term. We will use the following three definitions:

$$\begin{aligned}[x]x &= I \\ [x]y &= \mathbf{K}y \qquad (y \neq x) \\ [x](MN) &= \mathbf{S}([x]M)([x]N)\end{aligned}$$

Assuming $y \neq x$, we have the following.

$$\begin{aligned}CL(\lambda xy.yx) &= [x](CL(\lambda y.yx)) \\ &= [x]([y](CL(yx))) \\ &= [x]([y](CL(y)CL(x))) \\ &= [x]([y](yx)) \\ &= [x](\mathbf{S}([y]y)([y]x)) \\ &= [x](\mathbf{SI}(\mathbf{K}x)) \\ &= \mathbf{S}([x](\mathbf{SI}))([x](\mathbf{K}x)) \\ &= \mathbf{S}(\mathbf{K}(\mathbf{SI}))(\mathbf{S}([x]\mathbf{K})([x]x)) \\ &= \mathbf{S}(\mathbf{K}(\mathbf{SI}))(\mathbf{S}(\mathbf{KK})\mathbf{I})\end{aligned}$$

∎

(c) $\mathbf{B}$ such that $\mathbf{B}xyz \to x(yz)$.

Solution. In this problem, following the same procedure as above will be difficult. So, we will try to do something else. We can try to obtain the combinator by reversing the reductions, and this is what we will do here.

Observe the following.

$$\begin{aligned} x(yz) &= (\mathbf{K}xz)(yz) \\ &= \mathbf{S}(\mathbf{K}x)yz \\ &= ((\mathbf{KS})x)(\mathbf{K}x)yz \\ &= (\mathbf{S}(\mathbf{KS})\mathbf{K}x)yz \\ &= \mathbf{S}(\mathbf{KS})\mathbf{K}xyz \end{aligned}$$

and hence the required combinator **B** is $\mathbf{B} = \mathbf{S}(\mathbf{KS})\mathbf{K}$. ■

(d) $\mathbf{M}$ such that $\mathbf{M}x \to xx$.

Solution. In this problem, following the same procedure as in **(a)** will be easy. We want to translate the lambda-expression

$$\lambda x.xx$$

to its CL-term. So, we have the following.

$$\begin{aligned} CL(\lambda x.xx) &= [x](CL(xx)) \\ &= [x](CL(x)CL(x)) \\ &= [x](xx) \\ &= \mathbf{S}([x]x)([x]x) \\ &= \mathbf{SII} \end{aligned}$$

and hence the required combinator is $\mathbf{M} = \mathbf{SII}$. ■

8. First, we show that for any CL-term M, x does not occur in $[x]M$, where x is a variable.

The proof is by induction on the length of M. For the base case, suppose the length of M is 1, i.e $M = z$, where z is a variable or $z = \mathbf{S}$ or $\mathbf{K}$. Three cases are possible.

(1) In the first case, we have $z = x$. So,

$$[x]M = [x]z = [x]x = \mathbf{I}$$

and hence in $[x]M$, x does not occur.

(2) In the second case, we have $z \neq x$ and z is a variable. So,

$$[x]M = [x]z = \mathbf{K}z$$

and again in $[x]M$, x does not occur.

(3) In the last case, either $z = \mathbf{S}$ or $z = \mathbf{K}$. The proof is similar to case (2).

So, the base case is true.

Now, suppose the claim is true for all CL-terms of length atmost n. Let M be a CL-term of length $n+1$. Since M has length > 1, M can be written as an application of two CL-terms, i.e $M = XY$, where X, Y are CL-terms. In that case,

$$[x]M = [x](XY) = \mathbf{S}([x]X)([x]Y)$$

By the inductive hypothesis, both $[x]X$ and $[x]Y$ do not contain x, since the lengths of X and Y are atmost n. Hence, $[x]M$ does not contain x, and this completes the proof.

Next, we show that $([x]M)N \to M[x \leftarrow N]$. We will show this by induction on the length of M. For the base case, suppose the length of M is 1. So, either $M = z$ for some variable z, or $M = \mathbf{S}$ or $M = \mathbf{K}$.

(1) Suppose $M = z$ for some variable z. First, suppose $z \neq x$. Then

$$([x]M)N = ([x]z)N = (\mathbf{K}z)N \to z = M[x \leftarrow N]$$

where the last equality is true because there is no occurence of x in M. Next, suppose $z = x$. So,

$$([x]M)N = ([x]x)N = \mathbf{I}N \to N = M[x \leftarrow N]$$

(2) Next, suppose $M = \mathbf{S}$. Then,

$$([x]M)N = ([x]\mathbf{S})N = (\mathbf{KS})N \to \mathbf{S} = M[x \leftarrow N]$$

where again the last equality is true because x does not occur in M.

(3) Finally, suppose $M = \mathbf{K}$. Then,

$$([x]M)N = ([x]\mathbf{K})N = (\mathbf{KK})N \to \mathbf{K} = M[x \leftarrow N]$$

So, it follows that the base case is true. Now, suppose the statement is true for all CL-terms M of length atmost n. Let M be a CL-term of length $n+1$. Since M has length greater than 1, it can be written as an application of two terms, say $M = XY$, where X, Y are CL-terms of length atmost n. Then, we have the following.

$$([x]M)N = ([x](XY))N = (\mathbf{S}([x]X)([x]Y))N \to ([x]X)N(([x]Y)N)$$

Now by induction hypothesis, $([x]X)N \to X[x \leftarrow N]$ and $([x]Y)N \to Y[x \leftarrow N]$. So, we see that

$$([x]X)N(([x]Y)N) \to X[x \leftarrow N]Y[x \leftarrow N] = M[x \leftarrow N]$$

and hence it follows that

$$([x]M)N \to M[x \leftarrow N]$$

which completes the inductive proof.

We now find the combinatory logic terms to the given lambda terms.

(i) $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$

Solution. We have the following.

$$CL(\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))) = [f](CL((\lambda x.f(xx))(\lambda x.f(xx)))) \tag{0.10}$$
$$= [f](CL(\lambda x.f(xx))CL(\lambda x.f(xx))) \tag{0.11}$$

First let us compute $CL(\lambda x.f(xx))$.

$$CL(\lambda x.f(xx)) = [x](CL(f(xx))) \tag{0.12}$$
$$= [x](CL(f)CL(xx)) \tag{0.13}$$
$$= [x](f(CL(x)CL(x))) \tag{0.14}$$
$$= [x](f(xx)) \tag{0.15}$$
$$= \mathbf{S}([x]f)([x](xx)) \tag{0.16}$$
$$= \mathbf{S}(\mathbf{K}f)(\mathbf{S}([x]x)([x]x)) \tag{0.17}$$
$$= \mathbf{S}(\mathbf{K}f)(\mathbf{SII}) \tag{0.18}$$

Using the above result in equation (0.11), we get the following.

$$CL(\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))) = [f]((\mathbf{S}(\mathbf{K}f)(\mathbf{SII}))(\mathbf{S}(\mathbf{K}f)(\mathbf{SII}))) \tag{0.19}$$
$$= \mathbf{S}([f](\mathbf{S}(\mathbf{K}f)(\mathbf{SII})))([f](\mathbf{S}(\mathbf{K}f)(\mathbf{SII}))) \tag{0.20}$$

Now, we compute $[f](\mathbf{S}(\mathbf{K}f)(\mathbf{SII}))$. We have the following.

$$[f](\mathbf{S}(\mathbf{K}f)(\mathbf{SII})) = \mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))([f](\mathbf{SII})) \tag{0.21}$$
$$= \mathbf{S}(\mathbf{S}([f]\mathbf{S})([f](\mathbf{K}f)))(\mathbf{S}([f]\mathbf{S})([f](\mathbf{II}))) \tag{0.22}$$
$$= \mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}([f]\mathbf{K})([f]f)))(\mathbf{S}(\mathbf{KS})(\mathbf{S}([f]\mathbf{I})([f]\mathbf{I}))) \tag{0.23}$$
$$= \mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KI})(\mathbf{KI}))) \tag{0.24}$$

Using this result in equation (0.20), we get that

$$CL(\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))$$
$$= \mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KI})(\mathbf{KI}))))(\mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KI})(\mathbf{KI}))))$$

■

(ii) $\lambda f.(\lambda x.f(fx))$

Solution. First, we have the following.

$$CL(\lambda f.(\lambda x.f(fx))) = [f](CL(\lambda x.f(fx))) \tag{0.25}$$
$$= [f]([x](CL(f(fx)))) \tag{0.26}$$
$$= [f]([x](CL(f)CL(fx))) \tag{0.27}$$
$$= [f]([x](f(fx))) \tag{0.28}$$
$$= [f](\mathbf{S}([x]f)([x](fx))) \tag{0.29}$$
$$= [f](\mathbf{S}(\mathbf{K}f)(\mathbf{S}([x]f)([x]x))) \tag{0.30}$$
$$= [f](\mathbf{S}(\mathbf{K}f)(\mathbf{S}(\mathbf{K}f)\mathbf{I})) \tag{0.31}$$
$$= \mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))([f](\mathbf{S}(\mathbf{K}f)\mathbf{I})) \tag{0.32}$$
$$= \mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))(\mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))([f]\mathbf{I})) \tag{0.33}$$
$$= \mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))(\mathbf{S}([f](\mathbf{S}(\mathbf{K}f)))(\mathbf{KI})) \tag{0.34}$$

We now compute $[f](\mathbf{S}(\mathbf{K}f))$ as follows.

$$[f](\mathbf{S}(\mathbf{K}f)) = \mathbf{S}([f]\mathbf{S})([f](\mathbf{K}f)) \tag{0.35}$$
$$= \mathbf{S}(\mathbf{KS})(\mathbf{S}([f]\mathbf{K})([f]f)) \tag{0.36}$$
$$= \mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}) \tag{0.37}$$

Using this result in equation (0.34) we obtain

$$CL(\lambda f.(\lambda x.f(fx)))$$
$$= \mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{KI}))$$

■