PLC ASSIGNMENT-3

SIDDHANT CHAUDHARY
BMC201953

1. Recall the definition of *parallel reduction.* It is the relation $\implies$ over λ-terms defined by the following rules.

$$\frac{}{M \implies M}$$

$$\frac{}{(\lambda x.M)N \implies M[x := N]}$$

$$\frac{M \implies M'}{\lambda x.M \implies \lambda x.M'}$$

$$\frac{M \implies M' \qquad N \implies N'}{MN \implies M'N'}$$

$$\frac{M \implies M' \qquad N \implies N'}{(\lambda x.M)N \implies M'[x := N']}$$

Define M^* as follows.

$$\begin{aligned} x^* &= x \\ (\lambda x.M)^* &= \lambda x.M^* \\ (MN)^* &= M^*N^* \qquad (M \text{ not of the form } \lambda x.P) \\ ((\lambda x.P)N)^* &= P^*[x := N^*] \end{aligned}$$

Prove the following.

(a) If $M \to_\beta N$ then $M \implies N$.

Proof. Suppose $M \to_\beta N$, i.e M β-reduces to N in a single step. So, M contains a β-redux, say $(\lambda x.P)Q$, as a sub-term, and suppose the reduction $M \to_\beta N$ is the reduction of this redux. By the second rule above, we know that

$$(\lambda x.P)Q \implies P[x := Q]$$

and hence we conclude that $M \implies N$. This proves the claim. ∎

(b) If $M \implies N$ then $M \xrightarrow{*}_\beta N$.

Proof. Suppose $M \implies N$. We prove the claim by induction on the length of the λ-terms. For the base case, suppose $M \implies N$, and the length of the term M is 1. Clearly, $M = x$, where x is a variable, and the reduction $M \implies N$ is $x \implies x$. In this case, it is obvious that $x \xrightarrow{*}_\beta x$, i.e $M \xrightarrow{*}_\beta N$. So the base case is true.

Date: April 30, 2021.

Next, suppose the claim is true for all λ-terms M of length atmost $n-1$ for some $n-1 \in \mathbb{N}$. Then, suppose $M \implies N$, and suppose the length of the λ-term M is n. We have the following cases.

(1) In the first case, the reduction $M \implies N$ is of the form $M \implies M$. Clearly, in that case, we have $M \xrightarrow{*}_\beta M$ in zero steps.

(2) In the second case, suppose the reduction $M \implies N$ is of type two, i.e M contains a redux $(\lambda x.P)Q$, and the reduction $M \implies N$ involves the reduction $(\lambda x.P)Q \implies P[x := Q]$. Clearly, in this case, we have

$$(\lambda x.P)Q \xrightarrow{*}_\beta P[x := Q]$$

in one step, and hence $M \xrightarrow{*}_\beta N$ in one step.

(3) In the third case, suppose the reduction $M \implies N$ is of type three. So, M contains a term $\lambda x.P$ such that $P \implies Q$, and the reduction $M \implies N$ involves the reduction $\lambda x.P \implies \lambda x.Q$. Clearly, the length of the term P is atmost $n-1$, and by induction hypothesis, we know that $P \xrightarrow{*}_\beta Q$. So, it follows that

$$\lambda x.P \xrightarrow{*}_\beta \lambda x.Q$$

and hence

$$M \xrightarrow{*}_\beta N$$

in this case as well.

(4) In the next case, the reduction $M \implies N$ is of the fourth type. So, M contains a term of the form PS such that $P \implies P'$, $S \implies S'$, and the reduction $M \implies N$ involves the reduction $PS \implies P'S'$. Again, note that the length of the terms P and S is atmost $n-1$, and by induction hypothesis, we see that

$$P \xrightarrow{*}_\beta P' \quad , \quad S \xrightarrow{*}_\beta S'$$

This means that

$$PS \xrightarrow{*}_\beta P'S \xrightarrow{*}_\beta P'S'$$

and hence it follows that

$$M \xrightarrow{*}_\beta N$$

in multiple steps.

(5) In the next case, the reduction $M \implies N$ is of the fifth type. So, M contains a term $(\lambda x.P)Q$, where $P \implies P'$, $Q \implies Q'$, and the reduction $M \implies N$ involves the reduction $(\lambda x.P)Q \implies P'[x := Q']$. Again, note that the lengths of P and Q is atmost $n-1$. So, by the induction hypothesis, we see that $P \xrightarrow{*}_\beta P'$ and $Q \xrightarrow{*}_\beta Q'$. So, it follows that

$$(\lambda x.P)Q \xrightarrow{*}_\beta (\lambda x.P')Q \xrightarrow{*}_\beta (\lambda x.P')Q' \xrightarrow{*}_\beta P'[x := Q']$$

So, it follows that

$$M \xrightarrow{*}_\beta N$$

in multiple steps.

So by induction, the claim is true for all terms M. This completes the proof. ■

(c) $M \xrightarrow{*}_\beta N$ if and only if $M \overset{*}{\implies} N$.

Proof. First, suppose $M \xrightarrow{*}_\beta N$. So, there is a sequence $M_1, ..., M_k$ of λ-terms such that

$$M \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots \to_\beta M_k \to_\beta N$$

By part **(a)**, we see that

$$M \implies M_1 \implies M_2 \implies \cdots \implies M_k \implies N$$

and hence $M \overset{*}{\implies} N$.

Conversely, suppose $M \overset{*}{\implies} N$. So, there is a sequence $M_1, ..., M_k$ of λ-terms such that

$$M \implies M_1 \implies M_2 \implies \cdots \implies M_k \implies N$$

By part **(b)**, we see that

$$M \xrightarrow{*}_\beta M_1 \xrightarrow{*}_\beta M_2 \xrightarrow{*}_\beta \cdots \xrightarrow{*}_\beta M_k \xrightarrow{*}_\beta N$$

which implies that $M \xrightarrow{*}_\beta N$. This completes the proof. ■

Lemma 0.1. *$M \implies M^*$ for all M.*

Proof. We prove this by induction on the length of M. For the base case, suppose $M = x$ for a variable x. Clearly, we have $M^* = x^* = x = M$, and hence it follows that $M \implies M^*$, i.e the base case is true.

Now suppose the statement is true for all terms of length atmost $n - 1$, where $n - 1 \in \mathbb{N}$. Let M be a term of length n. We consider a few cases.

(1) Suppose $M = \lambda x.P$ for some P, where the length of P is atmost $n - 1$. Then, we see that

$$M^* = (\lambda x.P)^* = \lambda x.P^*$$

By the inductive hypothesis, we know that $P \implies P^*$. So, it follows that

$$M = \lambda x.P \implies \lambda x.P^* = M^*$$

and hence the claim is true in this case.

(2) Suppose $M = PQ$ where P is not of the form $\lambda x.S$. Clearly, P and Q have length atmost $n-1$. So, by the inductive hypothesis, we know that $P \implies P^*$ and $Q \implies Q^*$. Also,

$$M^* = (PQ)^* = P^*Q^*$$

So, it follows that

$$M = PQ \implies P^*Q^* = M^*$$

and the claim is true in this case as well.

(3) Finally, suppose M is of the form $(\lambda x.P)Q$. Clearly, the lengths of P and Q are atmost $n-1$, and hence by the inductive hypothesis we see that $P \implies P^*$ and $Q \implies Q^*$. Also, note that $M^* = ((\lambda x.P)Q)^* = P^*[x := Q^*]$. So, we have

$$M = (\lambda x.P)Q \implies P^*[x := Q^*] = M^*$$

So, by induction, the claim is true for all terms M, and this completes the proof. ■

(d) If $M \implies N$ then $N \implies M^*$.

Proof. We will prove this by induction on the size of M. For the base case, suppose M has size 1, i.e $M = x$ where x is a variable. Then, $N = x$, and hence $x \implies x$. In this case, note that $M^* = x$. So, it follows that $N \implies M^*$, and hence the base case is true.

Now, suppose the claim is true for all terms of size atmost $n-1$ for some $n-1 \in \mathbb{N}$. Let M be a term of size n such that $M \implies N$. We handle a couple of cases.

(1) Suppose the reduction $M \implies N$ is of the form $M \implies M$, i.e $N = M$. By **Lemma** 0.1, we know that $N = M \implies M^*$.

(2) Suppose $M = PQ$ where $P \implies P'$ and $Q \implies Q'$, and the reduction $M \implies N$ is of the form $PQ \implies P'Q'$. Since the lengths of both P and Q are atmost $n-1$, from the induction hypothesis we get that $P' \to P^*$ and $Q' \to Q^*$. Now, we have two subcases here.

   (a) If PQ is not a β-redux, then $M^* = P^*Q^*$. So $N = P'Q' \implies P^*Q^* = M^*$, and hence we are done.

   (b) Suppose PQ is a β-redux, say $P = \lambda x.S$, and hence $M^* = S^*[x := Q^*]$. Suppose the reduction $P \implies P'$ was of the form $P = \lambda x.S \implies \lambda x.S'$ where $S \implies S'$. By induction hypothesis, we see that $S' \implies S^*$, and since $Q' \implies Q^*$, it follows that $N = (\lambda x.S')Q' \implies S^*[x := Q^*] = M^*$.

(3) Suppose $M = (\lambda x.Q)P$ and $N = Q'[x := P']$ where $Q \implies Q'$ and $P \implies P'$. Then $M^* = Q^*[x := P^*]$, and $N \implies M^*$, because by the induction hypothesis we have $Q' \implies Q^*$ and $P' \implies P^*$.

So, by induction, the claim is true for all terms M of any size. This completes the proof. ■

(e) If $M \implies P$ and $M \implies Q$ then there exists N such that $P \implies N$ and $Q \implies N$.

Proof. This easily follows from part **(d)**. Suppose $M \implies P$ and $M \implies Q$. Invoking part **(d)**, we see that $P \implies M^*$ and $Q \implies M^*$. Setting $N = M^*$, this proves the claim. ■

2. Are the following expressions typable? If so, what are the most general types? If not, explain why.

(a) $\lambda fgx.f(gx)$

Solution. Yes, this term is typable. Let us derive the most general type for this. We begin with the following.

$$\tau_x = p_x, \quad \tau_g = p_g, \quad \tau_f = p_f$$
$$E_x = \phi, \quad E_g = \phi, \quad E_f = \phi$$

From these, we get

$$\tau_{gx} = a, \quad \tau_f = p_f$$
$$E_{gx} = \{p_g = p_x \to a\}, \quad E_f = \phi$$

Further, we get

$$\tau_{f(gx)} = b$$
$$E_{f(gx)} = \{p_g = p_x \to a, p_f = a \to b\}$$

Going ahead, we have the following.

$$\tau_{\lambda x.f(gx)} = c \to \tau_{f(gx)}[p_x := c] = c \to b$$
$$E_{\lambda x.f(gx)} = E_{f(gx)}[p_x := c] = \{p_g = c \to a, p_f = a \to b\}$$

Further, we have

$$\tau_{\lambda gx.f(gx)} = d \to \tau_{\lambda x.f(gx)}[p_g := d] = d \to c \to b$$
$$E_{\lambda gx.f(gx)} = E_{\lambda x.f(gx)}[p_g := d] = \{d = c \to a, p_f = a \to b\}$$

Finally, we have

$$\tau_{\lambda fgx.f(gx)} = e \to \tau_{\lambda gx.f(gx)}[p_f := e] = e \to d \to c \to b$$
$$E_{\lambda fgx.f(gx)} = E_{\lambda gx.f(gx)}[p_f := e] = \{d = c \to a, e = a \to b\}$$

So, it follows that the most general type of $\lambda fgx.f(gx)$ is

$$\tau_{\lambda fgx.f(gx)} = (a \to b) \to (c \to a) \to c \to b$$

and so we have found the required type. ■

(b) $\lambda xy.yx$

Solution. Yes, this term is also typable. Let us derive the most general type for this. We begin with the following.

$$\tau_x = p_x, \quad \tau_y = p_y$$
$$E_x = \phi, \quad E_y = \phi$$

From this, we get the following.

$$\tau_{yx} = a$$
$$E_{yx} = \{p_y = p_x \to a\}$$

From here, we can obtain the following.

$$\tau_{\lambda y.yx} = b \to \tau_{yx}[p_y := b] = b \to a$$
$$E_{\lambda y.yx} = E_{yx}[p_y := b] = \{b = p_x \to a\}$$

Finally, we get

$$\tau_{\lambda xy.yx} = c \to \tau_{\lambda y.yx}[p_x := c] = c \to b \to a$$
$$E_{\lambda xy.yx} = E_{\lambda y.yx}[p_x := c] = \{b = c \to a\}$$

So, it follows that the most general type of $\lambda xy.yx$ is

$$\tau_{\lambda xy.yx} = c \to (c \to a) \to a$$

and so we have found the required type. ■

(c) $\lambda fgx.g(fx)$

Solution. Yes, this term is typable. Let us derive the most general type for this. We begin with the following.

$$\tau_x = p_x, \quad \tau_g = p_g, \quad \tau_f = p_f$$
$$E_x = \phi, \quad E_g = \phi, \quad E_f = \phi$$

From these, we get

$$\tau_{fx} = a$$
$$E_{fx} = \{p_f = p_x \to a\}$$

Further, we get

$$\tau_{g(fx)} = b$$
$$E_{g(fx)} = \{p_f = p_x \to a, p_g = a \to b\}$$

Going ahead, we have the following.

$$\tau_{\lambda x.g(fx)} = c \to \tau_{g(fx)}[p_x := c] = c \to b$$
$$E_{\lambda x.g(fx)} = E_{g(fx)}[p_x := c] = \{p_f = c \to a, p_g = a \to b\}$$

Further, we have

$$\tau_{\lambda gx.g(fx)} = d \to \tau_{\lambda x.g(fx)}[p_g := d] = d \to c \to b$$
$$E_{\lambda gx.f(gx)} = E_{\lambda x.g(fx)}[p_g := d] = \{p_f = c \to a, d = a \to b\}$$

Finally, we have

$$\tau_{\lambda fgx.g(fx)} = e \to \tau_{\lambda gx.g(fx)}[p_f := e] = e \to d \to c \to b$$
$$E_{\lambda fgx.g(fx)} = E_{\lambda gx.g(fx)}[p_f := e] = \{e = c \to a, d = a \to b\}$$

So, it follows that the most general type of $\lambda fgx.g(fx)$ is

$$\tau_{\lambda fgx.g(fx)} = (c \to a) \to (a \to b) \to c \to b$$

and so we have found the required type. ■

3. Recall the following standard encodings: $f^0 x = x, f^{n+1}x = f(f^n x), [n] = (\lambda fx.f^n x), \textbf{true} = (\lambda xy.x), \textbf{false} = (\lambda xy.y), \textbf{pair} = (\lambda xyw.wxy), \textbf{fst} = (\lambda p.p\ \textbf{true}), \textbf{snd} = (\lambda p.p\ \textbf{false}), \textbf{ite} = (\lambda bxy.bxy)$ and $\textbf{iszero} = (\lambda x.(x(\lambda z.\textbf{false}))\textbf{true})$

Derive the most general types of each of the above expressions. If you feel that any of them is untypable, give a justification.

Solution. We will find the types individually below.

Type of $[n]$. If $n = 0$, then

$$[0] = \lambda fx.x$$

We will now derive the most general type of this using the following steps.

(1) $\tau_x = p_x, \tau_f = p_f$ and $E_f = \phi, E_x = \phi$.
(2) $\tau_{\lambda x.x} = a \to \tau_x[p_x := a] = a \to a$ and $E_{\lambda x.x} = E_x[p_x := a] = \phi$.
(3) $\tau_{\lambda fx.x} = b \to \tau_{\lambda x.x}[p_f := b] = b \to a \to a$ and $E_{\lambda fx.x} = E_{\lambda x.x}[p_f := b] = \phi$.

So, we see that the type of 0 is $b \to a \to a$.

Next, suppose $n \geq 1$. So,

$$[n] = \lambda fx.f^n x$$

Consider the following steps.

(1) $\tau_x = p_x, \tau_f = p_f$ and $E_f = \phi, E_x = \phi$.
(2) $\tau_{fx} = a_1$ and $E_{\lambda fx} = \{p_f = p_x \to a_1\}$.
(3) $\tau_{f^2x} = a_2$ and $E_{f^2x} = \{p_f = p_x \to a_1, p_f = a_1 \to a_2\}$.
(4) Continuing this way n times, we will obtain: $\tau_{f^n x} = a_n$ and $E_{f^n x} = \{p_f = p_x \to a_1, p_f = a_1 \to a_2, p_f = a_2 \to a_3, ..., p_f = a_{n-1} \to a_n\}$
(5) $\tau_{\lambda x.f^n x} = a \to \tau_{f^n x}[p_x := a] = a \to a_n$ and $E_{\lambda x.f^n x} = E_{f^n x}[p_x := a] = \{p_f = a \to a_1, p_f = a_1 \to a_2, ..., p_f = a_{n-1} \to a_n\}$.

(6) $\tau_{\lambda fx.f^n x} = b \to \tau_{\lambda x.f^n x}[p_f := b] = b \to a \to a_n$ and $E_{\lambda fx.f^n x} = E_{\lambda x.f^n x}[p_f := b] = \{b = a \to a_1, b = a_1 \to a_2, ..., b = a_{n-1} \to a_n\}$.

The only solution to this system is $a = a_1 = a_2 = \cdots a_n$. So, it follows that the type of $[n]$ in this case is $(a \to a) \to a \to a$.

Type of true. Consider the following steps.

(1) $\tau_x = p_x, \tau_y = p_y$ and $E_x = \phi, E_y = \phi$.
(2) $\tau_{\lambda y.x} = a \to p_x$ and $E_{\lambda y.x} = \phi$.
(3) $\tau_{\lambda xy.x} = b \to a \to b$ and $E_{\lambda xy.x} = \phi$.

So, the type of **true** is $b \to a \to b$.

Type of false. This derivation is very similar to the type derivation of **true**, and I won't repeat it. The type of **false** turns out to be $b \to a \to a$.

Type of pair. Consider the following steps.

(1) $\tau_w = p_w, \tau_y = p_y, \tau_x = p_x$ and $E_w = E_y = E_x = \phi$.
(2) $\tau_{wx} = a$ and $E_{wx} = \{p_w = p_x \to a\}$.
(3) $\tau_{wxy} = b$ and $E_{wxy} = \{p_w = p_x \to a, a = p_y \to b\}$.
(4) $\tau_{\lambda w.wxy} = c \to b$ and $E_{\lambda w.wxy} = \{c = p_x \to a, a = p_y \to b\}$.
(5) $\tau_{\lambda yw.wxy} = d \to c \to b$ and $E_{\lambda yw.wxy} = \{c = p_x \to a, a = d \to b\}$.
(6) $\tau_{\lambda xyw.wxy} = e \to d \to c \to b$ and $E_{\lambda eyw.wxy} = \{c = e \to a, a = d \to b\}$.

So, the type of **pair** is $e \to d \to (e \to (d \to b)) \to b$.

Type of fst. Consider the following steps. We will assume that the type of **true** (which we derived above) is $b \to a \to b$.

(1) $\tau_p = p_p$ and $E_p = \phi$.
(2) $\tau_{p\ \mathbf{true}} = c$ and $E_{p\ \mathbf{true}} = \{p_p = \tau_{\mathbf{true}} \to c = (b \to a \to b) \to c\}$.
(3) $\tau_{\lambda p.p\ \mathbf{true}} = d \to c$ and $E_{\lambda p.p\ \mathbf{true}} = \{d = (b \to a \to b) \to c\}$

With these steps, the type of **fst** is $(b \to a \to b) \to c \to c$.

Type of snd. This is very similar to the case of **fst**. If we follow the steps of type derivation, we will get that the type of **snd** is $(b \to a \to a) \to c \to c$.

Type of ite. Consider the following steps.

(1) $\tau_b = p_b, \tau_y = p_y, \tau_x = p_x$ and $E_b = E_y = E_x = \phi$.
(2) $\tau_{bx} = a$ and $E_{bx} = \{p_b = p_x \to a\}$.
(3) $\tau_{bxy} = b$ and $E_{bxy} = \{p_b = p_x \to a, a = p_y \to b\}$.
(4) $\tau_{\lambda y.bxy} = c \to b$ and $E_{\lambda y.bxy} = \{p_b = p_x \to a, a = c \to b\}$.
(5) $\tau_{\lambda xy.bxy} = d \to c \to b$ and $E_{\lambda xy.bxy} = \{p_b = d \to a, a = c \to b\}$.
(6) $\tau_{\lambda bxy.bxy} = e \to d \to c \to b$ and $E_{\lambda bxy.bxy} = \{e = d \to a, a = c \to b\}$.

It follows that the type of **ite** is $(d \to (c \to b)) \to d \to c \to b$.

Type of iszero. We assume that the types of **true** and **false** are $a \to b \to a$ and $c \to d \to d$ respectively. Consider the following steps.

(1) $\tau_{\mathbf{false}} = c \to d \to d$.
(2) $\tau_{\lambda z.\mathbf{false}} = e \to (c \to d \to d)$ and $E_{\lambda z.\mathbf{false}} = \phi$.
(3) $\tau_x = p_x$ and $E_x = \phi$.
(4) $\tau_{x(\lambda z.\mathbf{false})} = f$ and $E_{x(\lambda z.\mathbf{false})} = \{p_x = (e \to (c \to d \to d)) \to f\}$.
(5) $\tau_{(x(\lambda z.\mathbf{false}))\mathbf{true}} = g$ and $E_{(x(\lambda z.\mathbf{false}))\mathbf{true}} = \{p_x = (e \to (c \to d \to d)) \to f, f = (a \to b \to a) \to g\}$.

(6) $\tau_{\lambda x.(x(\lambda z.\textbf{false}))\textbf{true}} = h \to g$ and $E_{\lambda x.(x(\lambda z.\textbf{false}))\textbf{true}} = \{h = (e \to (c \to d \to d)) \to f, f = (a \to b \to a) \to g\}$.

So, it follows that the type of **iszero** is

$$(e \to (c \to d \to d)) \to ((a \to b \to a) \to g)) \to g$$

■