1 Introduction

When it comes to compiler design, there has been a dispute as to whether it is useful to transform the source code using the continuation passing style (CPS) transform first. Some have observed that it is easier to optimize CPS transformed code, while others maintain that direct style compilation is better. Sabry and Felleisen resolve the dispute by showing how to get the benefits of CPS without the CPS transform. They note that it is, indeed, sometimes easier to reason about CPS transformed terms than to reason directly about $\Lambda$-terms. They set about determining what axioms would make it just as easy to reason about $\Lambda$-terms. Their result is summarized in the following theorem, where $\lambda_C$ denotes the computational $\lambda$-calculus, and $cps(-)$ denotes the CPS transform.

**Theorem (SF):** $\lambda_C \vdash M = N$ iff $\lambda_{\beta, \eta} \vdash cps(M) = cps(N)$

Filinski and others have pointed out that the CPS transform can be seen as an instance of the monadic transform. (See table 5 for the monadic transform.) Instantiate $f^*$ and $\eta$ as follows to get the CPS transform from the monadic transform: $\eta = \lambda a. \lambda k. (ka), f^* = \lambda t. \lambda k. (t(\lambda a. ((fa)k)))$.

In this paper we generalize Sabry and Felleisen’s result and prove the following theorem.

**Theorem:** $\lambda_C \vdash M = N$ iff $\lambda_M \vdash \overline{M} = \overline{N}$

To prove this theorem, we follow Sabry and Felleisen in our approach. After proving that $\lambda_C \vdash M = N$ implies $\lambda_M \vdash \overline{M} = \overline{N}$, we define a grammar for monadic terms, and an inverse transform. We show that the inverse transform does, in fact, invert the monadic transform. The result then quickly follows.

2 The Computational $\lambda$-Calculus

The computation $\lambda$-calculus was introduced by Moggi. The call-by-value $\lambda$-calculus is too weak to prove all of the equations that one would like to prove. The computation $\lambda$-calculus provides a richer set of axioms from which to prove equations.

In the introduction, we state Sabry and Felleisen’s theorem as $\lambda_C \vdash M = N$ iff $\lambda_{\beta, \eta} \vdash cps(M) = cps(N)$. Actually, though, that is not exactly what they state. They state the theorem as follows.
\[
\begin{align*}
M & ::= V(MM) \\
V & ::= x\|\lambda x.M
\end{align*}
\]

Let \( \Lambda \) denote the language \( \mathcal{L} \).

Table 1: The terms of the \( \lambda \) calculus.

\[
\begin{align*}
(\lambda x.M)V & = M[x := V] \\
\lambda x.Vx & = V & \text{where } x \not\in \text{fv}(V) \\
\text{let } x = M \text{ in } x & = M
\end{align*}
\]

\[
\begin{align*}
(MN) & = \text{let } x = M \text{ in } (xN) & \text{where } x \not\in \text{fv}(N) \\
(VM) & = \text{let } y = M \text{ in } (Vy) & \text{where } y \not\in \text{fv}(V)
\end{align*}
\]

Table 2: The computational \( \lambda \) calculus \( \lambda_C \).

\[
\begin{align*}
(\lambda x.M)V & \rightarrow M[x := V] & \text{($\beta_V$)} \\
\lambda x.Vx & \rightarrow V & \text{where } x \not\in \text{fv}(V) & \text{(eta)} \\
(\lambda x.M) & \rightarrow M & \text{(unit)} \\
E((\lambda x.M)N) & \rightarrow ((\lambda x.E[M])N) & \text{where } x \not\in \text{fv}(E), E \neq [] & \text{(lift)} \\
(zLM) & \rightarrow ((\lambda u.(uL))(zM)) & \text{where } u \not\in \text{fv}(L) & \text{(liftr)} \\
(\lambda x.E[y])M & \rightarrow E[yM] & \text{where } x \not\in \text{fv}(E[y]) & \text{(b_{\text{structural}})}
\end{align*}
\]

The context \( E \) is defined as follows.

\[
E ::= []\|([V]\cdot)(E\cdot M)
\]

or equivalently as

\[
E ::= []\|E([[]]\cdot)]E([[]]\cdot M)]
\]

Table 3: The reduction version of the computational \( \lambda \) calculus \( \lambda_{C^\prime} \).

\[
\begin{align*}
(\lambda x.M)V & = M[x := V] & \text{($\beta_V$)} \\
\lambda x.Vx & = V & \text{where } x \not\in \text{fv}(V) & \text{(etaV)} \\
(\lambda x.M) & = M & \text{(id)} \\
E((\lambda x.M)N) & = ((\lambda x.E[M])N) & \text{where } x \not\in \text{fv}(E), E \neq [] & \text{(lift)} \\
(\lambda x.E[y])M & = E[yM] & \text{where } x \not\in \text{fv}(E[y]) & \text{(b_{\text{structural}})}
\end{align*}
\]

The context \( E \) is defined as follows.

\[
E ::= []\|([V]\cdot)(E\cdot M)
\]

or equivalently as

\[
E ::= []\|E([[]]\cdot)]E([[]]\cdot M)]
\]

Table 4: The equation version of the computational \( \lambda \) calculus \( \lambda_{C^\prime} \).
\[
\begin{align*}
\overline{V} &= \eta V^\circ \\
\overline{(MN)} &= ((\lambda f.(f^\circ \overline{N}))^* \overline{M})
\end{align*}
\]

\[
x^\circ = x \\
(\lambda x.M)^\circ = \lambda x.\overline{M}
\]

Sometimes it is convenient to write \(\mathcal{M}[M] = \overline{M}\).

Table 5: The monadic transform.

\[
\begin{align*}
P &::= (WW)|(W^*P) \\
W &::= x[\eta]\lambda x.P
\end{align*}
\]

Table 6: The monadic terms.

\[
\begin{align*}
((\lambda x.M)N) &\rightarrow M[x := N] \\
\lambda x.Mx &\rightarrow M \quad \text{where } x \notin \text{fv}(M) \\
(M^*(\eta N)) &\rightarrow (MN) \\
(\eta^*M) &\rightarrow M \\
(M_1^*(M_2^*N)) &\rightarrow ((\lambda x.(M_1^*(M_2x))^*)^*N) \quad \text{where } x \notin \text{fv}(M_1) \cup \text{fv}(M_2)
\end{align*}
\]

\(M, N \in L_P \cup L_W\)

The reductions can occur in any context.

Table 7: The monadic reductions.

\[
\begin{align*}
((\lambda x.P)W) &\rightarrow P[x := W] \\
\lambda x.Wx &\rightarrow W \quad \text{where } x \notin \text{fv}(W) \\
(W_1^*(\eta W_2)) &\rightarrow (W_1W_2) \\
(\eta^*P) &\rightarrow P \\
(W_1^*(W_2^*P)) &\rightarrow ((\lambda x.(W_1^*(W_2x))^*)^*P) \quad \text{where } x \notin \text{fv}(W_1) \cup \text{fv}(W_2)
\end{align*}
\]

The reductions can occur in any context.

Table 8: The monadic reductions.
Theorem (SF'): \( \lambda_{C'} \vdash M = N \) iff \( \lambda_{\beta, \eta} \vdash cps(M) = cps(N) \)

The computational \( \lambda \)-calculus that they define is \( \lambda_{C'} \), and it is given in tables 3 and 4. The version that we chose to use was Kucan’s variation, which we call \( \lambda_C \). It is given in table 2. As would be expected, \( \lambda_{C'} \) and \( \lambda_C \) are equivalent. We prove the following theorem.

**Theorem (1):** \( \lambda_C \rightleftarrows \lambda_{C'} \)

Thus the theorem as written in the introduction is equivalent to the theorem as Sabry and Felleisen stated it.

### 3 The Monadic Terms and Reductions

The grammar for the monadic terms is given in table 6. Although there are many possible ways to write a grammar for monadic terms, we found that this grammar simplified the problem of finding an inverse for the monadic transform. The monadic reductions are given in tables 7 and 8. Table 7 states the reductions without regard for the grammatical structure of monadic terms. These reductions follow from the usual \( \beta \) and \( \eta \) reductions with reduction forms of the monadic equations \( f^* \eta = f \), \( \eta^* = \text{id} \), and \((f \circ g)^* = f^* \circ g^*\). The reductions \((\ast \eta)\) and \((\eta \ast)\) seem to follow naturally from the corresponding monadic equations. The reduction for \((\ast)\) is less obvious. We base this reduction loosely on the corresponding reduction in Hatcliff and Danvy. Because of the grammar, the reductions in table 7 are too general. Table 8 specializes the reductions so that it is possible for the terms to be generated by the grammar. This form of the monadic reductions also makes it clear that the grammar is closed with respect to the reductions. We prove the following theorem.

**Theorem (3):** If \( P \in L_P \) and \( P \rightarrow Q \) then \( Q \in L_P \).

### 4 One Direction

To prove the if and only if result, as usual, we prove the ‘if’ and the ‘only if’ parts separately. The ‘if’ part is fairly straightforward. The proof consists of showing that the \( \lambda_C \) equations, when transformed, are derivable equations in \( \lambda_M \). Thus we have the following result.

**Theorem (2):** \( \lambda_C \vdash M = N \implies \lambda_M \vdash \overline{M} = \overline{N} \)

### 5 The Inverse Transform

The inverse transform is given in table 9. It is surprisingly simple. It essentially involves simply removing the \( \ast \)'s and \( \eta \)'s. We use the notation \( M^{-1}[\cdot] \) to suggest that the inverse transform does, in fact, invert \( M[\cdot] \), with respect to the appropriate notion of derivable equality. The following theorems establish that property of \( M^{-1}[\cdot] \).

**Theorem (4):** Let \( M \in \Lambda \). Then \( \lambda_C \vdash (M^{-1} \circ M)[M] = M \).

**Theorem (5):** If \( P \in L_P \) then \( \lambda_M \vdash (M \circ M^{-1})[P] \ast \rightarrow P \).
Corollary (6): For any $P \in L_P$, there is an $M \in \Lambda$ such that $\lambda_M \vdash \overline{M} \xrightarrow{\eta} P$.

6 The Other Direction

To establish the reverse implication, we first prove a more general result. In particular, we show that if two monadic terms are equal, then the Λ-terms obtained by applying the inverse transform to the monadic terms are also equal.

The proof consists of showing that the $\lambda_M$ equations, when transformed by the inverse transform, are derivable equations in $\lambda_C$. The theorem is as follows.

Theorem (7): If $\lambda_M \vdash P = Q$ then $\lambda_C \vdash \overline{P} = \overline{Q}$.

Since $M[M] \in L_P$, the following corollary follows immediately.

Corollary (8): If $\lambda_M \vdash M[M] = M[N]$ then $\lambda_C \vdash \overline{M}[M] = \overline{M}[N]$.

However, $M[-]$ and $M^{-1}[-]$ 'cancel out'. Thus we get the implication in the other direction.

Corollary (9): If $\lambda_M \vdash M[M] = M[N]$ then $\lambda_C \vdash M = N$.

Since we now have the implication going in both directions, we get the promised result.

Theorem (10): $\lambda_C \vdash M = N$ iff $\lambda_M \vdash \overline{M} = \overline{N}$

Furthermore, we get the following corollary which establishes that the CPS terms are an initial model for the monadic calculus.

Corollary: $\lambda_M \vdash \overline{M} = \overline{N}$ iff $\lambda_{\beta, \eta} \vdash cps(M) = cps(N)$

7 Conclusion and Future Work

We have established a generalization of Sabry and Felleisen’s result; namely that equational reasoning in the computational $\lambda$-calculus is equivalent to equational reasoning of the transformed terms in the monadic calculus. This result, together with Sabry and Felleisen’s result, establishes that the CPS terms are an initial model for the monadic calculus.

However, part of what makes CPS terms particularly interesting is that one can create contexts that cause strange results by passing in odd continuations (either by using call/cc or by explicitly using terms that are not in $cps(\Lambda)$). We would like to construct a notion of context for monadic terms, and establish the equivalence of contextual equivalence of monadic terms and CPS terms.

A Theorems with Proofs

Theorem (1): $\lambda_C \Downarrow \lambda_C'$

Proof:

Lemma (1.1): $\lambda_C \vdash \lambda_C'$

Proof:
Lemma (1.1.1): $\lambda_C \vdash E[(\lambda x. M)N] = (\lambda x. E[M])N$

Proof:

case(i): Suppose $E = []$.
Then $E[(\lambda x. M)N] = (\lambda x. E[M])N$ holds trivially.

case(ii): Suppose $E = (VE')$.

$$\begin{align*}
((\lambda x. (VE'[M]))) & \overset{1.1.1}{=} \text{IH. } (V((\lambda x. E'[M])N)) \\
& \overset{\text{let.2}}{=} \text{let } y = ((\lambda x. E'[M])N) \text{ in } (V y) \\
& = \text{let } y = (\text{let } x = N \text{ in } E'[M]) \text{ in } (V y) \\
& = \text{assoc } \text{let } x = N \text{ in } (\text{let } y = E'[M] \text{ in } (V y)) \\
& \overset{\text{let.2}}{=} \text{let } x = N \text{ in } (V E'[M]) \\
& = ((\lambda x. (V E'[M])[M])N)
\end{align*}$$

case(iii): Suppose $E = (E'L)$.

$$\begin{align*}
(E'[(\lambda x. M)N]L) & \overset{1.1.1}{=} \text{IH. } ((\lambda x. E'[M])N)L \\
& \overset{\text{let.1}}{=} \text{let } z = ((\lambda x. E'[M])N) \text{ in } (zL) \\
& = \text{let } z = (\text{let } x = N \text{ in } E'[M]) \text{ in } (zL) \\
& = \text{assoc } \text{let } x = N \text{ in } (\text{let } z = E'[M] \text{ in } (zL)) \\
& \overset{\text{let.1}}{=} \text{let } x = N \text{ in } (E'[M]L) \\
& = ((\lambda x. (E'[M]L)[M])N)
\end{align*}$$

QED

Lemma (1.1.2): $\lambda_C \vdash ((\lambda x. E[x])M)] = E[M]

Proof:

case(i): Suppose $E = []$.
Then $((\lambda x. E[x])M)] = E[M]$ follows directly from unit.

case(ii): Suppose $E = (VE')$.

$$\begin{align*}
((\lambda x. (VE'[x]))M) & = \text{let } x = M \text{ in } (VE'[x]) \\
& \overset{\text{let.2}}{=} \text{let } x = M \text{ in } (\text{let } y = E'[x] \text{ in } (V y)) \\
& = \text{assoc } \text{let } y = (\text{let } x = M \text{ in } E'[x]) \text{ in } (V y)) \\
& = ((\lambda y. (V y))[((\lambda x. E'[x])M)] \\
& \overset{1.1.1}{=} \text{IH. } ((\lambda y. (V y))E'[M]) \\
& = \text{eta}_v. \text{ (} V E'[M])
\end{align*}$$

case(iii): Suppose $E = (E'L)$.

$$\begin{align*}
(E'[M]N) & \overset{1.1.1}{=} \text{IH. } (((\lambda x. E'[x])M)N) \\
& = (\text{let } x = M \text{ in } E'[x])N \\
& \overset{\text{let.1}}{=} \text{let } z = (\text{let } x = M \text{ in } E'[x]) \text{ in } (zN) \\
& = \text{assoc } \text{let } x = M \text{ in } (\text{let } z = E'[x] \text{ in } (zN)) \\
& \overset{\text{let.1}}{=} \text{let } x = M \text{ in } (E'[x]N) \\
& = ((\lambda x. (E'[x]N))M)
\end{align*}$$

QED

Observe that the equations $\beta_v$, $\text{eta}_v$, $\text{unit}$ immediately entail the equations $\beta_v$, $\text{eta}_v$, and $\beta_{id}$. The lemmas take care of the other two cases.
**QED**

**Lemma (1.2):** $\lambda_C \vdash \lambda_C$

**Proof:**

Observe that the equations $\beta_V$, $\text{eta}_V$, $\beta_{id}$ immediately entail the equations $\beta_V$, $\text{eta}_V$, and $\text{unit}$.

For $\text{assoc}$, let $E = ((\lambda y. B)[[]])$. Then $E[(\lambda x. M)N] = \beta_{\text{lift}}((\lambda x. E[M])N)$.

Thus $\lambda_C \vdash ((\lambda y. B)((\lambda x. M)N)) = ((\lambda x.((\lambda y. B)M))N)$, which is the unsugared form of $\lambda_C \vdash \text{let } y = (\text{let } x = N \text{ in } M) \text{ in } B = \text{let } x = N \text{ in } (\text{let } y = M \text{ in } B)$.

For $\text{let.1}$, let $E = ([[]])$. Then $((\lambda x. E[x])M) = \beta_{\text{eta}} E[M]$. Thus $\lambda_C \vdash ((\lambda x.(xN))M) = (MN)$, which is the unsugared form of $\lambda_C \vdash \text{let } x = M \text{ in } (xN) = (MN)$.

For $\text{let.2}$, let $E = (V[[]])$. Then $((\lambda y. E[y])M) = \beta_{\text{eta}} E[M]$. Thus $\lambda_C \vdash ((\lambda y.(V y))M) = (VM)$, which is the unsugared form of $\lambda_C \vdash \text{let } y = M \text{ in } (V y) = (VM)$.

**QED**

**Theorem (2):** $\lambda_C \vdash M = N \implies \lambda_M \vdash \overline{M} = \overline{N}$

**Proof:**

**Lemma (2.1):** $\lambda_M \vdash (\lambda x. M)V = \overline{M[x := V]}$

**Proof:**

**Lemma (2.1.1):** $\overline{M[x := V^0]} = \overline{M[x := V]}$

**Proof:**

case(i) Suppose $M = y$.

1. Suppose $y = x$.

   $\overline{M[x := V^0]} = \eta x[x := V^0]$

   $= \eta V^0$

   $= \overline{V}$

   $= x[x := V]$

   $= \overline{M[x := V]}$

2. Suppose $y \neq x$.

   $\overline{M[x := V^0]} = \overline{M} = \overline{M[x := V]}$

   case(ii) Suppose $M = (\lambda z. N)$.

1. Suppose $z = x$.

   $\overline{M[x := V^0]} = \overline{M} = \overline{M[x := V]}$

7
(2) Suppose \( z \neq x \).

(a) Suppose \( x \notin \text{fv}(N) \) or \( z \notin \text{fv}(V) \).

\[
\overline{M[x := V^o]} = (\eta(\lambda z.\overline{N}[x := V^o]) = \eta(\lambda z.\overline{N}[x := V])
\]

\[\text{I.H.} \eta(\lambda z.\overline{N}[x := V]) = (\lambda z.\overline{N}[x := V]) = M[x := V] \]

(b) Suppose \( x \in \text{fv}(N) \) and \( z \in \text{fv}(V) \).

\[
\overline{M[x := V^o]} = (\eta(\lambda z.\overline{N}[x := V^o]) = \eta(\lambda y.(\overline{N}[z := y])[x := V^o])
\]

\[\text{I.H.} \eta(\lambda y.(\overline{N}[z := y])[x := V^o]) = \eta(\lambda y.(\overline{N}[z := y])[x := V]) = M[x := V] \]

case(iii) Suppose \( M = (N_1,N_2) \).

\[
\overline{M[x := V^o]} = (((\lambda f.(f^*\overline{N}_2)[x := V^o])^*\overline{N}_1[x := V^o]) = \text{I.H.} ((\lambda f.(f^*\overline{N}_2)[x := V^o])^*\overline{N}_1[x := V])
\]

\[\text{I.H.} ((\lambda f.(f^*\overline{N}_2)[x := V^o])^*\overline{N}_1[x := V]) = \overline{N_1[x := V] N_2[x := V]} = M[x := V] \]

QED

\[
((\lambda x.M)V) = ((\lambda f.(f^*\overline{V}))^*\eta(\lambda x.\overline{M}))
\]

\[= \beta \eta(\lambda x.\overline{M})\overline{V} = (\lambda x.\overline{M})^*\eta V^o \]

\[= \beta \eta((\lambda x.\overline{M})^*\eta V^o)\overline{M[x := V]} = M[x := V] \]

QED

Lemma (2.2): \( \lambda M \vdash (\lambda x.V x) = \overline{V} \) where \( x \notin \text{fv}(V) \)

Proof:

\[
(\lambda x.V x) = \eta(\lambda x.\overline{V x}) = \eta(\lambda x.(\lambda f.(f^*\eta x))^*(\eta V^o))
\]

\[= \eta(\lambda x.(\lambda f.(f^*\eta x)V^o)) = \eta(\lambda x.(\lambda f.(f^\eta x)V^o)) = \eta(\lambda x.(\lambda f.(f^\eta x)V^o)) = \beta \eta(\lambda x.(V^o x)) \]

8
\[= \text{eta } \eta V^\circ \]
\[= V \]

**QED**

**Lemma (2.3):** \( \lambda M \vdash (\lambda x.x)M = M \)

**Proof:**

\[
(\lambda x.x)M = ((\lambda f.f^*M)^*\eta(\lambda x.x)) \\
=_{\ast \eta} (\lambda f.f^*M)(\lambda x.x) \\
= \text{eta } ((\lambda f.f^*M)\eta) \\
=_{\beta} \eta^*M \\
=_{\eta} M
\]

**QED**

**Lemma (2.4):** \( \lambda M \vdash (\lambda x.(xN))M = (MN) \) where \( x \not\in \text{fv}(N) \)

**Proof:**

\[
(\lambda x.(xN))M = ((\lambda f.f^*M)^*\eta(\lambda x.(\lambda g.g^*N)^*\eta x)) \\
=_{\ast \eta} (\lambda f.f^*M)^*\eta((\lambda g.g^*N)x)) \\
=_{\ast \eta} (\lambda f.f^*M)(\lambda x.(\lambda g.g^*N)x) \\
=_{\beta} (\lambda f.f^*M)((\lambda x.(x^*N))) \\
=_{\beta} (\lambda x.(x^*N))^*M \\
= (MN)
\]

**QED**

**Lemma (2.5):** \( \lambda M \vdash (\lambda y.(Vg))M = (VM) \) where \( y \not\in \text{fv}(V) \)

**Proof:**

\[
(\lambda y.(Vg))M = ((\lambda f.f^*M)^*\eta(\lambda y.(\lambda g.g^*y)^*V)) \\
=_{\ast \eta} ((\lambda f.f^*M)^*\eta(\lambda y.(\lambda g.g^*y)^*V)) \\
=_{\ast \eta} (\lambda f.f^*M)((\lambda y.(\lambda g.g^*y)^*V)) \\
= (\lambda f.f^*M)(\lambda y.(\lambda g.g^*y)^*V) \\
=_{\beta} (\lambda f.f^*M)((\lambda g.(V^\circ y))) \\
= \text{eta } ((\lambda f.f^*M)V^\circ) \\
=_{\ast \eta} ((\lambda f.f^*M)^*\eta V^\circ) \\
= ((\lambda f.f^*M)^*V) \\
= (VM)
\]

**QED**
\textbf{Lemma (2.6):} $\lambda_M \vdash (\lambda x_2.M)((\lambda x_1.M_2)M_1) = (\lambda x_1.((\lambda x_2.M_2)M_1))$

where $x_1 \notin \text{fv}(M)$

\textbf{Proof:}

\[
(\lambda x_2.M)((\lambda x_1.M_2)M_1) = ((\lambda f.f^*((\lambda g.g^*M_1)^*\eta(\lambda x_1.M_2)))^*\eta(\lambda x_2.M))
\]

\[
= \eta ((\lambda f.f^*((\lambda g.g^*M_1)^*\eta(\lambda x_1.M_2)))((\lambda x_2.M))
\]

\[
= \beta ((\lambda x_2.M)^*((\lambda g.g^*M_1)^*\eta(\lambda x_1.M_2)))
\]

\[
= \beta (\lambda x_2.M)^* (\lambda x_2.\overline{M})
\]

\[
= (\lambda x_2.((\lambda x_2.M)^*\overline{M})) (\lambda x_2.\overline{M})
\]

\[
= (\lambda x_1.((\lambda x_2.M)^*\overline{M})) (M_1)
\]

\[
QED
\]

\textbf{QED}

\textbf{Theorem (3):}

(i) If $P \in L_P$ and $P \rightarrow Q$ then $Q \in L_P$.

(ii) If $W \in L_W$ and $W \rightarrow W'$ then $W' \in L_W$.

\textbf{Proof:}

\textbf{Lemma (3.1):}

(i) If $P \in L_P$ and $W \in L_W$ then $P[x := W] \in L_P$.

(ii) If $W \in L_W$ and $W' \in L_W$ then $W[x := W'] \in L_W$.

\textbf{Proof:}

\textbf{case(i):} Suppose $P = (W_1W_2)$.

Since $P \in L_P$, it must be that $W_1, W_2 \in L_W$. Observe that $(W_1W_2)[x := W] = (W_1[x := W]W_2[x := W])$. By the inductive hypothesis, $W_1[x := W], W_2[x := W] \in L_W$. Thus $W_1[x := W]W_2[x := W] \in L_P$.

\textbf{case(ii):} Suppose $P = (W^*_1P')$.

Since $P \in L_P$, it must be that $W_1 \in L_W$, and $P' \in L_P$. Observe that $(W^*_1P')[x := W] = (W_1[x := W]^*P'[x := W])$. By the inductive hypothesis, $W_1[x := W] \in L_W$, and $P'[x := W] \in L_P$. Thus $(W_1[x := W]^*P'[x := W]) \in L_P$. 

10
case (iii): Suppose $W = y$.

(1) Suppose $y = x$.

Then $W[x := W'] = W' \in L_W$.

(2) Suppose $y \neq x$.

Then $W[x := W'] = W \in L_W$.

case (iv): Suppose $W = \eta$.

Then $W[x := W'] = W \in L_W$.

case (v): Suppose $W = \lambda y. P$.

(1) Suppose $y = x$.

Then $W[x := W'] = W \in L_W$.

(2) Suppose $y \neq x$.

(a) Suppose $x \notin \text{fv}(P)$ or $y \notin \text{fv}(W')$.

Since $W \in L_W$, it must be that $P \in L_P$. Observe that $(\lambda y. P)[x := W'] = (\lambda y. P[x := W'])$. By the inductive hypothesis, $P[x := W'] \in L_P$. Thus $(\lambda y. P[x := W']) \in L_W$.

(b) Suppose $x \in \text{fv}(P)$ and $y \in \text{fv}(W')$.

Since $W \in L_W$, it must be that $P \in L_P$. Observe that $(\lambda y. P)[x := W'] = (\lambda z. (P[y := z])[x := W'])$. By the inductive hypothesis, $(P[y := z])[x := W'] \in L_P$. Thus $(\lambda z. (P[y := z])[x := W']) \in L_W$.

QED

case (i): Suppose the reduction is $\beta$.

(1) Suppose $P = (W_1 W_2)$.

If the reduction occurred in $W_i$, then the result follows from the induction hypothesis. Otherwise $W_1 = \lambda x. P'$, and so $Q = P'[x := W_2]$. It follows from lemma(3.1) that $P'[x := W_2] \in L_P$. Thus $Q \in L_P$.

(2) Suppose $P = (W^* P')$.

In this case, a reduction can only occur in $W$ or $P'$. Thus the result follows from the induction hypothesis.

(3) Suppose $W = \lambda x. P$.

In this case, a reduction can only occur in $P$. Thus the result follows from the induction hypothesis.

case (ii): Suppose the reduction is eta.

(1) Suppose $P = (W_1 W_2)$.

In this case, a reduction can only occur in $W_i$. Thus the result follows from the induction hypothesis.

(2) Suppose $P = (W^* P')$.

In this case, a reduction can only occur in $W$ or $P'$. Thus the result follows from the induction hypothesis.
(3) Suppose $W = \lambda x. P$.
If the reduction occurs in $P$, then the result follows from the induction hypothesis. Otherwise, $P = (W''x)$. And so $W' = W''$. Since $W'' \in L_W$, $W' \in L_W$.

case(iii): Suppose the reduction is $\ast \eta$.
(1) Suppose $P = (W_1 W_2)$.
In this case, a reduction can only occur in $W_1$. Thus the result follows from the induction hypothesis.
(2) Suppose $P = (W^* P')$.
If the reduction occurs in $W$ or $P'$, then the result follows from the induction hypothesis. Otherwise, $P' = (W_1 W_2)$ and $W_1 = \eta$. Thus $P = (W^* (\eta W_2))$, and $Q = (WW_2)$. Since $W, W_2 \in L_W$, it follows that $(WW_2) \in L_P$. Thus $Q \in L_P$.
(3) Suppose $W = \lambda x. P$.
In this case, a reduction can only occur in $P$. Thus the result follows from the induction hypothesis.

case(iv): Suppose the reduction is $\eta \ast$.
(1) Suppose $P = (W_1 W_2)$.
In this case, a reduction can only occur in $W_1$. Thus the result follows from the induction hypothesis.
(2) Suppose $P = (W^* P')$.
If the reduction occurs in $W$ or $P'$, then the result follows from the induction hypothesis. Otherwise, $W = \eta$, and so $Q = P'$.
Since $P' \in L_P$, $Q \in L_P$.
(3) Suppose $W = \lambda x. P$.
In this case, a reduction can only occur in $P$. Thus the result follows from the induction hypothesis.

case(v): Suppose the reduction is $\ast$.
(1) Suppose $P = (W_1 W_2)$.
In this case, a reduction can only occur in $W_1$. Thus the result follows from the induction hypothesis.
(2) Suppose $P = (W^* P')$.
If the reduction occurs in $W$ or $P'$, then the result follows from the induction hypothesis. Otherwise, $P' = (W_2 x)\in L_P$. And so, since $W_1 \in L_W$ and $(W_2 x) \in L_P$, $(W_2 x) \in L_P$. Thus $(\lambda x. (W_1 W_2 x)) \in L_W$, and so, since $P'' \in L_P$, $(\lambda x. (W_1 W_2 x))\ast (P'') \in L_P$. Thus $Q \in L_P$.
(3) Suppose $W = \lambda x. P$.
In this case, a reduction can only occur in $P$. Thus the result follows from the induction hypothesis.

QED
**Theorem (4):** Let $M \in \Lambda$. Then $\lambda C \vdash (M^{-1} \circ M)[M] = M$.

**Proof:**

- **case (i):** Suppose $M = x$.
  
  \[
  (M^{-1} \circ M)[M] = (M^{-1} \circ M)[x] \\
  = M^{-1}[M[x]] \\
  = M^{-1}[\eta x] \\
  = ((\lambda y.y)x) \\
  = \text{unit } x \\
  = M
  \]

- **case (ii):** Suppose $M = \lambda x.M'$.
  
  \[
  (M^{-1} \circ M)[M] = (M^{-1} \circ M)[\lambda x.M'] \\
  = M^{-1}[\lambda x.M'][x] \\
  = M^{-1}[\eta(\lambda x.M'[M])] \\
  = ((\lambda y.y)(\lambda x.(M^{-1} \circ M)[M])) \\
  = \text{I.H. } ((\lambda y.y)(\lambda x.M')) \\
  = \text{unit } (\lambda x.M') \\
  = M
  \]

- **case (iii):** Suppose $M = (M'M'')$.
  
  \[
  (M^{-1} \circ M)[M] = (M^{-1} \circ M)[M'M''] \\
  = M^{-1}[M'[M'']] \\
  = M^{-1}[\lambda f.(f(M^{-1} \circ M')[M])][M'] \\
  = \text{I.H. } ((\lambda f.(f(M^{-1} \circ M')[M])M') \\
  = \text{let } 1 (M'M'') \\
  = M
  \]

QED

**Theorem (5):**

- (i) If $P \in L_P$ then $\lambda M \vdash (M \circ M^{-1})[P] \vdash* P$.

- (ii) If $W \in L_W$ then $\lambda M \vdash (W_\circ)^* \vdash* W$.

**Proof:**

- **case (i):** Suppose $P = (W_1W_2)$.
  
  \[
  (M \circ M^{-1})[P] = (M \circ M^{-1})[(W_1W_2)] \\
  = M[[W_1W_2]] \\
  = \text{I.H. } ((\lambda f.\nu f(W_2)\circ \nu(W_1))\circ \nu W_1) \\
  \]

- **case (ii):** Suppose $W = (W_1W_2)$. 
  
  \[
  (M \circ M^{-1})[W] = (M \circ M^{-1})[(W_1W_2)] \\
  = M[[W_1W_2]] \\
  = \text{I.H. } ((\lambda f.\nu f(W_2)\circ \nu(W_1))\circ \nu W_1) \\
  \]

13
\[ \eta \circ (\lambda f. (f^* \eta W_2))W_1 \]
\[ \eta \circ (\lambda f. (fW_2))W_1 \]
\[ \beta (W_1W_2) = P \]

**case(ii):** Suppose \( P = (W^*P') \).

\[
(M \circ M^{-1})[P] = (M \circ M^{-1})[(W^*P')] \\
= M[M^{-1}[(W^*P')]] \\
= M[[W_\circ M^{-1}([P'])]] \\
= ((\lambda f. (f^*(M \circ M^{-1}))^*\eta(W_\circ))^*)^\circ) \\
\rightarrow_{\text{I.H.}} ((\lambda f. (f^*(P'))^*\eta(W_\circ))^*) \\
\rightarrow_{\eta} ((\lambda f. (f^*(P')))((W_\circ)^*)) \\
\rightarrow_{\text{I.H.}} ((\lambda f. (f^*(P')))W) \\
\rightarrow \beta (W^*P') \\
= P
\]

**case(iii):** Suppose \( W = x \).

\[
(W_\circ)^\circ = (x_\circ)^\circ \\
= x^\circ \\
= x \\
= W
\]

**case(iv):** Suppose \( W = \eta \).

\[
(W_\circ)^\circ = (\eta_\circ)^\circ \\
= (\lambda y. y)^\circ \\
= \lambda y. \eta y \\
\rightarrow_{\text{eta}} \eta \\
= W
\]

**case(v):** Suppose \( W = \lambda x. P \).

\[
(W_\circ)^\circ = ((\lambda x. P)_\circ)^\circ \\
= ((\lambda x. M^{-1}[P])^\circ) \\
= \lambda x. (M \circ M^{-1})[P] \\
\rightarrow_{\text{I.H.}} \lambda x. P \\
= W
\]

**QED**

**Corollary (6):** For any \( P \in L_P \), there is an \( M \in \Lambda \) such that \( \lambda M \vdash M^* \rightarrow P \).

**Proof:**

Simply pick \( M = M^{-1}[P] \).

**QED**

**Theorem (7):**

(i) If \( \lambda_M \vdash P = Q \) then \( \lambda_C \vdash P = Q \).
If $\lambda M \vdash W = W'$ then $\lambda C \vdash W_0 = W'_0$.

**Proof:**

**Lemma (7.1):** $\lambda C \vdash (\lambda x.M)N = P[x := N]$ 

**Proof:**

**Lemma (7.1.1):**

(i) If $P \in L_P$ and $W \in L_W$ then $P[x := W_0] = P[x := W]$.

(ii) If $W \in L_W$ and $W' \in L_W$ then $W_0[x := W'_0] = (W[x := W'])_0$.

**Proof:**

case (i): Suppose $P = (W_1W_2)$,

$P[x := W_0] = (W_1W_2)[x := W_0]$

$\overset{\text{I.H.}}{=} (W_1_0W_2_0)[x := W_0]$ 

$\overset{\text{I.H.}}{=} (W_1_0[x := W_0]W_2_0[x := W_0])$ 

$\overset{\text{I.H.}}{=} (W_1_0[x := W_0])_0(W_2_0[x := W])_0$ 

$\overset{\text{I.H.}}{=} (W_1[x := W]W_2[x := W])_0$ 

$\overset{\text{I.H.}}{=} (W_1W_2)[x := W]$ 

$\overset{\text{I.H.}}{=} P[x := W]$ 

case (ii): Suppose $P = (W'^*P')$.

$P[x := W_0] = (W'^*P')[x := W_0]$

$\overset{\text{I.H.}}{=} (W'^*P')_0[x := W_0]$ 

$\overset{\text{I.H.}}{=} (W'^*_0[x := W_0]P'_0[x := W_0])$ 

$\overset{\text{I.H.}}{=} (W'_0[x := W_0])_0P'_0[x := W]$ 

$\overset{\text{I.H.}}{=} (W'[x := W])_0P'[x := W]$ 

$\overset{\text{I.H.}}{=} (W'^*P')[x := W]$ 

$\overset{\text{I.H.}}{=} P[x := W]$ 

case (iii): Suppose $W = y$.

(1) Suppose $y = x$.

$W_0[x := W'_0] = x_0[x := W'_0]$ 

$\overset{\text{I.H.}}{=} x[x := W'_0]$ 

$\overset{\text{I.H.}}{=} W'_0$ 

$\overset{\text{I.H.}}{=} (x[x := W'_0])_0$ 

$\overset{\text{I.H.}}{=} (W[x := W'])_0$ 

(2) Suppose $y \neq x$.

$W_0[x := W'_0] = y_0[x := W'_0]$ 

$\overset{\text{I.H.}}{=} y[x := W'_0]$ 

$\overset{\text{I.H.}}{=} y$ 

$\overset{\text{I.H.}}{=} y_0$ 

$\overset{\text{I.H.}}{=} (y[x := W'])_0$ 

$\overset{\text{I.H.}}{=} (W[x := W'])_0$ 

15
case (iv): Suppose \( W = \eta \).
\[
W_0[x := W'_0] = \eta_0[x := W'_0] = (\lambda y.y)[x := W'_0] = (\lambda y.y) = \eta_0 = (\eta[x := W'_0])_0 = (W[x := W'_0])_0
\]

case (v): Suppose \( W = \lambda y.P \).

1) Suppose \( y = x \).
\[
W_0[x := W'_0] = (\lambda x.P)_0[x := W'_0] = (\lambda x.P)[x := W'_0] = (\lambda x.P) = (\lambda x.P)_0 = ((\lambda x.P)[x := W'_0])_0 = (W[x := W'_0])_0
\]

2) Suppose \( y \neq x \).
\[
W_0[x := W'_0] = (\lambda y.P)_0[x := W'_0] = (\lambda y.P)(x := W'_0) = (\lambda y.\overline{P}[x := \overline{W'_0}]) = \text{I.H.} \quad (\lambda y.P[x := W'_0])_0 = ((\lambda y.P)[x := W'_0])_0 = ((\lambda y.P)[x := W'_0])_0 = (W[x := W'_0])_0
\]

**QED**

\[
(\lambda x.M)N = (\lambda x.M)N_0 =_\beta M_0 = M[x := N_0] \text{ by lemma(7.1.1)}
\]

**QED**

**Lemma 7.2**: \( \lambda \vdash (\lambda x.Mx)_0 = M_0 \)

**Proof**:

\[
(\lambda x.Mx)_0 = \lambda x.(M_0x) = \text{eta} \quad M_0
\]

**QED**

**Lemma 7.3**: \( \lambda \vdash (M^*(\eta N)) = (MN) \)

**Proof**:

\[
(M^*(\eta N)) = (M_0((\lambda y.y)N_0)) = \text{unit} \quad (M_0N_0) = (MN)
\]
Lemma (7.4): \( \lambda \vdash (\eta^*M) = M \)

Proof:

\[
\begin{align*}
(\eta^*M) &= ((\lambda y.y)M) \\
&= \text{unit } M
\end{align*}
\]

QED

Lemma (7.5): \( \lambda \vdash (\eta_1^*(M\eta_2^*N)) = ((\lambda x.(\eta_1^*(\eta_2^*x)))^*N) \) where \( x \notin \text{fv}(M_1) \cup \text{fv}(M_2) \).

Proof:

\[
\begin{align*}
(M_1^*(M_2^*N)) &= (M_1^*(M_2^*N)) \\
&= \text{eta } ((\lambda z.(M_1^*z))(M_2^*N)) \\
&= \text{let } 2 ((\lambda z.(M_1^*z))((\lambda x.(M_2^*x))N)) \\
&= \text{assoc } (\lambda x.(\lambda z.(M_1^*z))(M_2^*x))N) \\
&= \text{eta } (\lambda x.((M_1^*(M_2^*x)))N) \\
&= ((\lambda x.(M_1^*(M_2^*x)))N) \\
&= ((\lambda x.(M_1^*(M_2^*x)))N) \\
&= ((\lambda x.(M_1^*(M_2^*x)))N) \\
&= ((\lambda x.(M_1^*(M_2^*x)))N)
\end{align*}
\]

QED

Corollary (8): If \( \lambda \vdash M[N] = M[N] \) then \( \lambda \vdash M[N] = M[N] \).

Corollary (9): If \( \lambda \vdash M[N] = M[N] \) then \( \lambda \vdash M = N \).

Proof:

This result follows from theorem(2) and corollary(8).

QED

Corollary (10): \( \lambda \vdash M = N \) iff \( \lambda \vdash M = N \)

Proof:

\( \implies \): theorem(2)

\( \impliedby \): corollary(9)

QED
\[
\begin{align*}
(WW') & = (W_0W'_0) \\
(W^2P) & = (W_0P) \\
\end{align*}
\]

\[
\begin{align*}
x_\circ &= x \\
\eta_\circ &= \lambda x.x \\
(\lambda x.P)_\circ &= \lambda x.\underline{P}
\end{align*}
\]

Sometimes it is convenient to write \( M^{-1}[P] = \underline{P} \).

Table 9: The inverse transform.