ISSN 1346-5597

National Institute of Informatics

NII Technical Report

Types for Hereditary Permutators

Makoto Tatsuta

NII-2007-010E
Nov. 2007

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

Types for Hereditary Permutators

Makoto Tatsuta

National Institute of Informatics
2-1-2 Hitotsubashi, Tokyo 101-8430, Japan
tatsuta@nii.ac.jp

Abstract

This paper answers the open problem of finding a type system that characterizes hereditary permutators, which is the problem 20 in the TLCA list of open problems. First this paper shows that there does not exist such a type system by showing that the set of hereditary permutators is not recursively enumerable. The set of positive primitive recursive functions is used to prove it. Secondly this paper gives a best-possible solution by providing a set of types such that a term has every type in the set if and only if the term is a hereditary permutator. Intersection types and the Omega type is used to handle infinite computation in lambda terms.

1 Introduction

A hereditary permutator is a lambda term that represents finite or infinite nests of permutations. Finite hereditary permutators have been proved to characterize the invertible terms in λ-calculus with $\beta\eta$-reduction [3]. [1] proved that hereditary permutators characterize the invertible terms in Scott's model D_∞. Invertible terms were used to characterize type isomorphisms [2].

A characterization of some class of lambda terms by a type system is always an interesting question [4, 7]. The question of finding a characterization of hereditary permutators by a type system has been studied intensively and is the problem 20 on the TLCA list of open problems [5].

This paper answers this question. First this paper shows that there does not exist such a type system by showing that the set of hereditary permutators is not recursively enumerable. The set of hereditary permutators cannot be characterized by any type system and any type with a recursively enumerable language and a recursively enumerable set of inference rules. Secondly this paper gives a best-possible solution by providing a set of types such that a term has every type in the set if and only if the term is a hereditary permutator.

The set of positive primitive recursive functions is used to prove the first claim. This set will be shown to be not recursively enumerable. For a give primitive recursive function f, we will construct a lambda term obtained from some hereditary permutator by modifying some node in its Böhm tree for all n such that the Böhm tree of the modified node is the same as that of the original node if and only if $f(n)$ is positive. Then we could decide whether f is positive by deciding whether this term is a hereditary permutator.

For the second claim, we will use intersection types and the Omega type to handle infinite computation in lambda terms. They will also enable us to use the subject reduction property and the subject expansion property.

Section 2 describes basic definitions and the problem. Non-existence of the solution is proved in Section 3. Section 4 gives permutator schemes that neatly characterizes hereditary permutators. Section 5 defines the type system with the set of types that characterizes hereditary permutators. One direction of the characterization is proved in Section 6. Section 7 proves the other direction and completes the characterization theorem.

2 Hereditary Permutators

In this section, we will give the problem 20 in the TLCA list of open problems [5] as well as basic definitions.

Definition 2.1 (λ-Calculus). We have variables $x, y, z, \ldots$. λ-terms $M, N, \ldots$ are defined by:
$M, N, \ldots ::= x|\lambda x.M|MM$.

FV(M) denotes the set of free variables in M. $M[x := N]$ denotes a standard substitution. $M \equiv N$ denotes the syntactical equality modulo renaming bound variables. Vars is the set of variables. Λ is the set of λ-terms.

One-step β-reduction $M \to_\beta N$ is defined by the compatible closure of

$$(\beta)\quad (\lambda x.M)N \to_\beta M[x := N].$$

β-reduction $M \to_\beta^* $ is defined as the reflexive transitive closure of the relation $\to_\beta$. β-equality $M =_\beta N$ is defined as the least equivalence relation including $\to_\beta^*$. We say M reduces N if $M \to_\beta^* N$. A λ-term of the shape $(\lambda x.M)N$ is called a *redex*. A λ-term M is called *normal* if there is not any λ-term N such that $M \to_\beta N$.

A λ-term M is called *head normal* if M is of the shape $\lambda x_1 \ldots x_n.yN_1 \ldots N_m$. A λ-term M is called *head normalizing* if there is some head normal term N such that $M \to_\beta^* N$.

Definition 2.2 (Böhm Trees). We suppose $\perp$ is a constant. N^* is defined to be the set of a finite sequence of natural numbers. $\langle m_0, m_1, \ldots, m_n\rangle$ denotes an element in N^*, $\langle\rangle$ denotes the empty sequence, and $s * t$ denotes the concatenation of sequences s and t. $s \le t$ is defined to hold for $s, t \in N^*$ if s is an initial segment of t.

A tree is defined by a partial function f from N^* to the set $\{\lambda x_1 \ldots x_n.y | x_1, \ldots, x_n, y \in \text{Vars}, n \ge 0\} \cup \{\perp\}$ such that
(1) $f(t)$ is undefined if $f(s)$ is undefined and $s \le t$,
(2) for any s there is a number i such that $f(s * \langle i\rangle)$ is undefined,
(3) $f(s * \langle i\rangle)$ is undefined if $f(s * \langle j\rangle)$ is undefined and $j < i$,
(4) $f(s * \langle i\rangle)$ is undefined if $f(s) = \perp$.

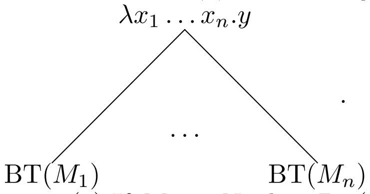
We will write $\perp$ for the tree f when $f(\langle\rangle) = \perp$. We will also write $[\lambda x_1 \ldots x_n.y, T_0, \ldots, T_m]$ for the tree f when $f(\langle\rangle) = \lambda x_1 \ldots x_n.y$ and $f(\langle m\rangle)$ is defined, $f(\langle m+1\rangle)$ is undefined, and the tree T_i is the function f_i such that $f_i(s) = f(\langle i\rangle * s)$.

When the tree T is $[\lambda x_1 \ldots x_n.y, T_1, \ldots, T_m]$, $\lambda x_1 \ldots x_n.y$ is called a node of the tree T, and a node of some subtree T_i is also called a node of T. $\lambda x_1 \ldots x_n.y$ is called the root in the tree T. If the tree T is given by a function f, the node $f(s * \langle i\rangle)$ is called a child node of the node $f(s)$.

The depth of the node in the tree is defined by:
(1) The depth of the node $\lambda x_1 \ldots x_n.y$ in the tree $[\lambda x_1 \ldots x_n.y, T_1, \ldots, T_m]$ is 0,
(2) The depth of a node in the tree $[\lambda x_1 \ldots x_n.y, T_1, \ldots, T_m]$ is $d + 1$ if the depth of the node in some subtree T_i is d.

Böhm tree BT(M) of a λ-term M is defined by:
(1) BT$(M) = \perp$ if M is not head normalizing,
(2) BT$(M) = [\lambda x_1 \ldots x_n.y, \text{BT}(M_1), \ldots, \text{BT}(M_m)]$ if $M =_\beta \lambda x_1 \ldots x_n.yM_1 \ldots M_m$.

Remark. (1) The tree $[\lambda x_1 \ldots x_n.y, \text{BT}(M_1), \ldots, \text{BT}(M_m)]$ is traditionally drawn by the picture



.

(2) If $M =_\beta N$, then BT$(M) = $ BT(N).
(3) BT$(M) = \perp$ if and only if M is not head normalizing.

Example. BT$(\lambda w.(\lambda x.y)x((\lambda z.z)z)w)$ = BT$(\lambda w.yzw)$ = $[\lambda w.y, z, w]$. BT$(\lambda w.(\lambda x.xx)(\lambda x.xx)((\lambda z.z)w)$ = $\perp$. BT$(\lambda w.(\lambda x.y)x((\lambda x.xx)(\lambda x.xx))w)$ = $[\lambda w.y, \perp, w]$. BT$(\lambda w.(\lambda x.y)x((\lambda z.y)((\lambda x.xx)(\lambda x.xx)))) = [\lambda w.y, y]$.

Definition 2.3 (Hereditary Permutators). A λ-term M is called a *hereditary permutator* if BT(M) satisfies the following conditions:

(H1) Its root has the shape $\lambda zx_1 \ldots x_n.z$ and the multiset of its child nodes is $\{\lambda w_1^i \ldots w_{n_i}^i.x_i | 1 \le i \le n\}$ for some variables $w_j^i \ne x_i$.

(H2) A node except the root has the shape $\lambda x_1 \ldots x_n.y$ and the multiset of its child nodes is $\{\lambda w^i_1 \ldots w^i_{n_i}.x_i | 1 \le i \le n\}$ for some variables $w^i_j \ne x_i$.

The problem 20 in the TLCA list of open problems is finding a type that characterizes the set of hereditary permutators. This question expects that there is some type system T with some type A such that $M : A$ is provable in T if and only if M is a hereditary permutator.

We will answer this question. First, in Section 3 we will show that the set of hereditary permutators is not recursively enumerable. Hence we will conclude that there does not exist any type that characterizes hereditary permutators if the system has a recursively enumerable language and a recursively enumerable set of inference rules. Secondly, in Section 5 we will present an intersection type system with a set of types which characterizes hereditary permutators.

In the description of the problem 20 in the TLCA list of open problems, a hereditary permutator M is defined to satisfy

(H0') M is closed and each variable occurs once in M,

(H1') M has the shape $\lambda zx_1 \ldots x_n.zM_1 \ldots M_m$,

(H2') In $\mathrm{BT}(M)$, the set of the child nodes of a node $\lambda x_1 \ldots x_n.y$ includes the set $\{\lambda w^i_1 \ldots w^i_l.x_i | 1 \le i \le n\}$ for some variables $w^i_j \ne x_i$.

By this definition, we might think that $\lambda zx.z((\lambda w.x)z)$ is not a hereditary permutator because of (H0') and $\lambda z.(\lambda y.z)y$ is not a hereditary permutator either because of (H1'). We understood the conditions (H0') and (H1') as not those for M, but those for $\mathrm{BT}(M)$. Then this definition is equivalent to Definition 2.3.

3 Non-existence of a type for hereditary permutators

We will show that there does not exist any type that characterizes the set of hereditary permutators.

Notation. We will write HP for the set of hereditary permutators. N is the set of natural numbers. We will use a vector notation $\vec{e}$ to denote a sequence $e_1, \ldots, e_n$ ($n \ge 0$). For example, we will use $\vec{M}$ to denote a sequence of λ-terms $M_1, \ldots, M_n$ ($n \ge 0$). $M\vec{N}$ denotes $MN_1 \ldots N_n$. $\lambda\vec{x}.M$ denotes $\lambda x_1 \ldots x_n.M$. $f(\vec{x})$ denotes $f(x_1, \ldots, x_n)$ if $\vec{x}$ denotes the sequence $x_1, \ldots, x_n$. We will write $\overline{n}$ for the n-th Church numeral $\lambda fx.f^n x$ where $f^n x$ denotes $f(f(\ldots(fx)\ldots))$ (n times of f). We will also write $\tilde{n}$ for $S^n\overline{0}$ for a number n.

First, we give several notations for partial recursive functions.

Definition 3.1. We write $\{n\}^{pr}(x)$ for the n-th unary primitive recursive function. $\langle x, y\rangle$ denotes the standard primitive recursive pairing, and $\pi_0(x)$ and $\pi_1(x)$ are the first and second projections respectively. The n-th unary partial recursive function $\{n\}(x)$ is defined by $\{\pi_1(n)\}^{pr}(\mu y.(\{\pi_0(n)\}^{pr}(\langle x, y\rangle) = 0))$. We also define $u(x, y) = \{x\}^{pr}(y)$.

Remark. The function u is a universal function for unary primitive recursive functions. u is a total recursive function.

Definition 3.2. For a function $f : N^n \to N$, we say that a λ-term F represents f when $f(m_1, \ldots, m_n) = m$ iff $F\overline{m_1} \ldots \overline{m_n} \to^*_\beta \overline{m}$.

Theorem 4.15 in Page 53 in [6] showed the following claim.

Theorem 3.3 ([6]). For every recursive function f, there is some λ-term F such that F represents f.

Definition 3.4. PPR is defined to be the set $\{n \in N | \forall x(\{n\}^{pr}(x) > 0)\}$.

PPR is the set of indexes for positive primitive recursive functions.

Proposition 3.5. The set PPR is not recursively enumerable.

Proof. By the standard result from recursion theory, we have the recursive function $S : N^2 \to N$ defined by $\{S(n,m)\}^{pr}(x) = \{n\}^{pr}(\{m\}^{pr}(x))$, and the recursive function $P : N \to N$ defined by $\{P(n)\}^{pr}(m) = \langle n, m\rangle$.

Assume that PPR is recursively enumerable. We will show contradiction.

Define a partial function $f : N \to N$ by $f(x) = 1$ if $S(\pi_0(x), P(x))$ is in PPR, and $f(x)$ is undefined otherwise. Then f is partial recursive. There is a number e such that for all x, both $\{e\}(x)$ and $f(x)$ has the same value or both are undefined.

Then we show that $f(x)$ is defined if and only if $\{x\}(x)$ is undefined. It is proved as follows: $f(x)$ is defined iff $S(\pi_0(x), P(x))$ is in PPR by the definition of f, iff $\forall y(\{\pi_0(x)\}^{pr}(\{P(x)\}^{pr}(y)) > 0)$ by the definition of S and PPR, iff $\forall y(\{\pi_0(x)\}^{pr}(\langle x, y\rangle) > 0)$ by the definition of P, iff $\{\pi_1(x)\}^{pr}(\mu y.(\{\pi_0(x)\}^{pr}(\langle x, y\rangle) = 0))$ is undefined, iff $\{x\}(x)$ is undefined.

If $\{e\}(e)$ is defined, then $f(e)$ is defined by the definition of e, and hence $\{e\}(e)$ is undefined by the above. Hence $\{e\}(e)$ is undefined. However, $f(e)$ is undefined by the definition of e, and hence $\{e\}(e)$ is defined by the above, which is contradiction.

Consequently, the set PPR is not recursively enumerable. □

We define

$$\begin{aligned}
&\mathrm{S} \equiv \lambda yfx.f(yfx),\\
&Y_0 \equiv \lambda xy.y(xxy),\\
&Y \equiv Y_0Y_0.
\end{aligned}$$

The term S is the successor for Church numerals. Y is Turing's fixed point operator.

Remark. $YM \to^*_\beta M(YM)$.

We now prove the first main theorem by using PPR.

Theorem 3.6. The set HP of hereditary permutators is not recursively enumerable.

Proof. Assume HP is recursively enumerable. We will show contradiction.

Define P as $Y(\lambda pz_0z_1.z_0(pz_1))$. Then $\mathrm{BT}(\lambda z_0.Pz_0)$ is $\begin{array}{c}\lambda z_0z_1.z_0\\ | \\ \lambda z_2.z_1\\ | \\ \lambda z_3.z_2\\ | \\ \vdots\end{array}$, since $Pz_n \to^*_\beta (\lambda z_0z_1.z_0(Pz_1))z_n =_\alpha (\lambda z_nz_{n+1}.z_n(Pz_{n+1}))z_n \to_\beta \lambda z_{n+1}.z_n(Pz_{n+1})$. Hence $\lambda z_0.Pz_0$ is in HP.

By Theorem 3.3, we have a λ-term U that represents the function u. Define T by

$$\begin{aligned}
&\Delta \equiv \lambda x.xx,\\
&T \equiv Y(\lambda txyz_0z_1.Uxy(\lambda w.z_0(tx(\mathrm{S}y)z_1))(\Delta\Delta)).
\end{aligned}$$

Then we have $T\bar{e}\tilde{n}z_n \to^*_\beta \lambda z_{n+1}.U\bar{e}\tilde{n}(\lambda w.z_n(T\bar{e}\widetilde{(n+1)}z_{n+1}))(\Delta\Delta)$ in a similar way to P. Hence $\mathrm{BT}(T\bar{e}\tilde{n}z_n)$ is $[\lambda z_{n+1}.z_n, \mathrm{BT}(T\bar{e}\widetilde{(n+1)}z_{n+1})]$ if $\{e\}^{pr}(n) > 0$ since $U\bar{e}\tilde{n} =_\beta \overline{k+1}$ for some k and $\mathrm{BT}(T\bar{e}\tilde{n}z_n)$ is $\perp$ if $\{e\}^{pr}(n) = 0$ since $U\bar{e}\tilde{n} =_\beta \bar{0}$.

We show that $\lambda z_0.T\bar{e}\bar{0}z_0$ is in HP if and only if e is in PPR. The direction form the right to the left is proved by $\mathrm{BT}(\lambda z_0.T\bar{e}\bar{0}z_0) = \mathrm{BT}(\lambda z_0.Pz_0)$ when e is in PPR, and hence $\mathrm{BT}(\lambda z_0.T\bar{e}\bar{0}z_0)$ is in HP. In order to show the direction form the left to the right, first we assume e is not in PPR and will show $\lambda z_0.T\bar{e}\bar{0}z_0$ is not in HP. From $e \notin$ PPR, we have a number m_0 such that $\{e\}^{pr}(m_0) = 0$ and $\{e\}^{pr}(m) > 0$ for all $m < m_0$. Then $\mathrm{BT}(\lambda z_0.T\bar{e}\bar{0}z_0) = \begin{array}{c}\lambda z_0z_1.z_0\\ | \\ \lambda z_2.z_1\\ | \\ \lambda z_3.z_2\\ | \\ \vdots\\ | \\ \lambda z_{m_0}.z_{m_0-1}\\ | \\ \perp\end{array}$, and hence $\mathrm{BT}(\lambda z_0.T\bar{e}\bar{0}z_0)$ is not in HP.

If HP were recursively enumerable, then PPR would be recursively enumerable, which would lead to contradiction. Therefore HP is not recursively enumerable. □

Non-existence of solutions for the problem follows immediately from the previous theorem.

Theorem 3.7 (Non-existence of a type for hereditary permutators). There does not exist any type system T with any type A such that its language and the set of its inference rules are recursively enumerable, and the set of hereditary permutators is the same as $\{M \in \Lambda | \Gamma \vdash M : A \text{ is provable in } T \text{ for some } \Gamma\}$.

Proof. If we had such a type system T, then $\{M \in \Lambda | \Gamma \vdash M : A \text{ is provable in } T \text{ for some } \Gamma\}$ would be recursively enumerable, and therefore HP would be recursively enumerable, which would contradict to Theorem 3.6. □

4 Permutator Schemes

We will define a permutator scheme that has the same Böhm tree as the application of a hereditary permutator to a variable. Permutator schemes will neatly characterize hereditary permutators and we will discuss some set-theoretic properties of hereditary permutators by using them.

Definition 4.1. We write $\mathcal{S}_m$ for the symmetric group of order m. We define the set $\mathrm{PS}_n(z)$ for $n \geq 0$ and a variable z by

$$\mathrm{PS}_0(z) = \Lambda,$$
$$\mathrm{PS}_{n+1}(z) = \{M \in \Lambda | M \to_\beta^* \lambda x_1 \dots x_m.zM_{\pi(1)} \dots M_{\pi(m)}, \pi \in \mathcal{S}_m, M_i \in \mathrm{PS}_n(x_i) \quad (1 \leq i \leq m)\}.$$

Remark. (1) $M \in \mathrm{PS}_k(z)$ iff there is a hereditary permutator N such that $\mathrm{BT}(\lambda z.M)$ and $\mathrm{BT}(N)$ are the same at any node at depth $< k$.

(2) $\mathrm{PS}_{k+1}(z) \subseteq \mathrm{PS}_k(z)$.

Lemma 4.2. (1) Let $n > 0$. M is in $\mathrm{PS}_n(z)$ if and only if the following hold in $\mathrm{BT}(M)$.

(a) Its root has the shape $\lambda \vec{x}.z$,
(b) If it has a node $\lambda x_1 \dots x_m.y$ at depth $< n-1$, the multiset of its child nodes is $\{\lambda \vec{w}_i.x_i | 1 \leq i \leq m\}$ for some variables $\vec{w}_i$ different from x_i,
(c) If it has a node $\lambda x_1 \dots x_m.y$ at depth $n-1$, the node has m child nodes.

(2) $\lambda z.M$ is in HP if and only if M is in $\mathrm{PS}_n(z)$ for all $n > 0$.

Proof. (1) The left-hand side to the right-hand side. By induction on n.

Case $n = 1$. (a) $M \to_\beta^* \lambda x_1 \dots x_m.zM_{\pi(1)} \dots M_{\pi(m)}$, so the claim holds.

(b) There is not any node at depth < 0.

(c) The node at depth 0 is the root and has m child nodes.

Case $n > 1$. Suppose $M \in \mathrm{PS}_n(z)$. Let $M \to_\beta^* \lambda x_1 \dots x_m.zM_{\pi(1)} \dots M_{\pi(m)}$ and $M_i \in \mathrm{PS}_{n-1}(x_i)$.

(a) The claim holds since its head variable is z.

(b) If the node is the root, the claim holds because the root of $\mathrm{BT}(M_i)$ is $\lambda \vec{w}_i.x_i$ by induction hypothesis (a) for $n-1$ with $M_i \in \mathrm{PS}_{n-1}(x_i)$. If the node is not the root, the node at depth $< n-1$ is some node at depth $< n-2$ of $\mathrm{BT}(M_i)$ for some i. By induction hypothesis (b) for $n-1$ with $M_i \in \mathrm{PS}_{n-1}(x_i)$, the claim holds.

(c) Any node at depth $= n-1$ is some node at depth $= n-2$ of $\mathrm{BT}(M_i)$ for some i. By induction hypothesis (c) for $n-1$ with $M_i \in \mathrm{PS}_{n-1}(x_i)$, the claim holds.

The right-hand side to the left-hand side. By induction on n.

Case $n = 1$. The claim holds from (a) and (c).

Case $n > 1$. By (a), we have $M \to_\beta^* \lambda x_1 \dots x_m.z\vec{M}$. By (b), $z\vec{M}$ is $zM_{\pi(1)} \dots M_{\pi(m)}$ for some $\pi \in \mathcal{S}_m$ and $M_i \to_\beta^* \lambda \vec{w}_i.x_i\vec{N}_i$.

We show that $\mathrm{BT}(M_i)$ satisfies (a) to (c) for x_i and $n-1$.

(a) Its root is $\lambda \vec{w}_i.x_i$.

(b) Any node at depth $< n-2$ in $\mathrm{BT}(M_i)$ is some node at depth $< n-1$ in $\mathrm{BT}(M)$, so the claim holds.

(b) Any node at depth $= n-2$ in $\mathrm{BT}(M_i)$ is some node at depth $= n-1$ in $\mathrm{BT}(M)$, so the claim holds.

By induction hypothesis for $n-1$, we have $M_i \in \mathrm{PS}_{n-1}(x_i)$. Hence M is in $\mathrm{PS}_n(z)$

(2) (A) The left-hand side to the right-hand side. For each n, we will show $M \in \mathrm{PS}_n(z)$. If $n = 0$, then the claim holds since $\mathrm{PS}_0(z) = \Lambda$. Suppose $n > 0$. We will show that $\mathrm{BT}(M)$ satisfies (a) to (c) in (1) for z and n.

(a) The claim holds by (H1).

(b)(c) If the node is the root, the claim follows from (H1). If the node is not the root, the claim follows from (H2).

Hence we have $M \in \mathrm{PS}_n(z)$ from (1).

(B) The right-hand side to the left-hand side.

(H1) The claim follows from $M \in \mathrm{PS}_2(z)$ and (a) and (b) in (1).

(H2) Let d be the depth of the node. The claim follows from $M \in \mathrm{PS}_{d+2}(z)$ and (b) in (1). □

5 Types for hereditary permutators

This section will present a type system with a set of types which characterizes hereditary permutators.

Definition 5.1. We define the type system $\mathcal{T}$.

We have type constants $p_n, q_m \quad (n \geq 0, m \geq 1)$, and Ω. Types $A, B, \ldots$ are defined by:
$A, B, \ldots ::= p_n | q_m | \Omega | A \to A | A \cap A \quad (n \geq 1, m \geq 0)$.

Type partial equivalence $A \sim_n B$ for $n > 0$ is defined by:

$$\Omega \sim_0 \Omega \qquad \frac{A_i \sim_n B_i \qquad (1 \leq i \leq m)}{B_{\pi(1)} \to \ldots \to B_{\pi(m)} \to q_k \sim_{n+1} A_1 \to \ldots \to A_m \to q_k}$$

where $\pi \in \mathcal{S}_m$ and $\mathrm{TC}(A_i, B_i) - \{\Omega\} \quad (1 \leq i \leq m), \{q_k\}$ are disjoint in the second rule.

A type declaration is a finite set of the form $\{x_1 : A_1, \ldots, x_n : A_n\}$ where x_i's are distinct variables and A_i's are types. We will write $\Gamma, \Delta, \ldots$ for a type declaration. A judgment is $\Gamma \vdash M : A$. We will also write $x_1 : B_1, \ldots, x_n : B_n \vdash M : A$ for $\{x_1 : B_1, \ldots, x_n : B_n\} \vdash M : A$, and $\Gamma, y : C \vdash M : A$ for $x_1 : B_1, \ldots, x_n : B_n, y : C \vdash M : A$, when Γ is $\{x_1 : B_1, \ldots, x_n : B_n\}$.

Typing rules are given by:

$$\frac{}{\Gamma, x : A \vdash x : A}\,(Ass) \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\,(\to I) \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}\,(\to E)$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash M : B}{\Gamma \vdash M : A \cap B}\,(\cap I) \qquad \frac{\Gamma \vdash M : A \cap B}{\Gamma \vdash M : A}\,(\cap E_1) \qquad \frac{\Gamma \vdash M : A \cap B}{\Gamma \vdash M : B}\,(\cap E_2)$$

$$\frac{}{\Gamma \vdash M : \Omega}\,(\Omega) \qquad \frac{\Gamma, z : A \vdash M : B \quad A \sim_n B}{\Gamma \vdash \lambda z.M : p_n}\,(p_n I)$$

Notation. $\mathrm{TC}(\vec{A})$ is defined as the set of type constants in the types $\vec{A}$.

Remark. The relation $\sim_n$ is proved to be a partial equivalence relation.

This is a standard intersection type system except for the type partial equivalence $\sim_n$ and the constants p_n, q_m. The intended meaning of the relation $\sim_n$ and the constants p_n are the set $\{(A, B) | z : A \vdash M : B \text{ iff } M \in \mathrm{PS}_n(z)\}$ and the set $\{\lambda z.M | M \in \mathrm{PS}_n(z)\}$ respectively. The constants q_m are used for keeping hierarchy in $\sim_n$. Our discussion will also go well in the same way when we add the type preorder $\leq$ with set-theoretically-sound rules such as $A \cap B \leq B \cap A$. Intersection types and the type Ω are necessary since we need the subject expansion property in our proof.

We have a characterization theorem of HP by this type system with the set of the types p_n.

Theorem 5.2 (Characterization Theorem). *M is a hereditary permutator if and only if $\vdash M : p_n$ is provable in the type system $\mathcal{T}$ for all n.*

We will finish the proof of this theorem in Section 7. The soundness of this characterization will be proved in Section 6 and its completeness will be shown in Section 7.

6 Soundness

We will prove the soundness part of Theorem 5.2 by using permutator schemes.

First we will show basic properties for the system $\mathcal{T}$.

Definition 6.1. $[A]$ is defined as a type of the shape defined by:

$$[A] ::= A|[A] \cap B|B \cap [A].$$

$[A]$ ambiguously denotes some type obtained from A by intersection.

We have a standard generation lemma.

Lemma 6.2 (Generation Lemma). (1) If $\Gamma, x : A \vdash x : B$ and $B \neq \Omega$, then A is $[B]$.

(2) If $\Gamma \vdash \lambda x.M : [A \to B]$, then $\Gamma, x : A \vdash M : B$.

(3) If $\Gamma \vdash MN : [B]$ and $B \neq \Omega$, then there is some A such that $\Gamma \vdash M : A \to B$ and $\Gamma \vdash N : A$.

(4) If $\Gamma, x : A_1 {\to} \ldots {\to} A_n {\to} B \vdash xM_1 \ldots M_m : C_{l+1} {\to} \ldots {\to} C_n {\to} B$, $n > 0$, and $C_{l+1} {\to} \ldots {\to} C_n {\to} B \neq \Omega$, then $l = m$, $A_i = C_i \quad (l < i \leq n)$, and $\Gamma, x : A_1 \to \ldots \to A_n \to B \vdash M_i : A_i \quad (1 \leq i \leq l)$.

Proof. (1)(2)(3) By induction on the proof.

(4) The claim follows from (1) and (3). □

Next we will show the subject reduction property. The next lemma is an auxiliary lemma for that.

Lemma 6.3. $\Gamma, x : A \vdash M : B$ and $\Gamma \vdash N : A$ imply $\Gamma \vdash M[x := N] : B$.

Proof. By induction on $\Gamma, x : A \vdash M : B$ □

Proposition 6.4 (Subject Reduction). If $\Gamma \vdash M : A$ and $M \to_\beta M'$, then $\Gamma \vdash M' : A$.

Proof. Induction on the proof. Consider cases according to the last rule. By induction hypothesis, we can immediately show cases where M is not the redex. We will show only interesting cases.

Case $({\to}E)$ where M is the redex. The proof is

$$\frac{\Gamma \vdash \lambda x.M : B \to A, \quad \Gamma \vdash N : B}{\Gamma \vdash (\lambda x.M)N : A}.$$

and M' is $M[x := N]$.

By Lemma 6.2 (2) for $\Gamma \vdash \lambda x.M : B \to A$, we have $\Gamma, x : B \vdash M : A$. By Lemma 6.3, we have $\Gamma \vdash M[x := N] : A$.

Case (Ip_n). The proof is

$$\frac{\Gamma, z : A \vdash N : B \quad A \sim_n B}{\Gamma \vdash \lambda z.N : p_n}$$

and M' is $\lambda z.N'$ where $N \to_\beta N'$. By induction hypothesis, we have $\Gamma, z : A \vdash N' : B$. Hence $\Gamma \vdash \lambda z.N' : p_n$ holds. □

Then we will show the subject expansion property. For that, we need the following lemma.

Lemma 6.5. If $\Gamma \vdash M[x := N] : A$, then there is some type B such that $\Gamma, x : B \vdash M : A$ and $\Gamma \vdash N : B$.

Proof. Induction on the proof. Consider cases according to the last rule. We will show only interesting cases.

Case $x \notin \mathrm{FV}(M)$. Let B be Ω.

Case $({\to}E)$. The proof is

$$\frac{\Gamma \vdash M_1 : C \to A \quad \Gamma \vdash M_2 : C}{\Gamma \vdash M_1 M_2 : A}.$$

By induction hypothesis, we have B_1 such that $\Gamma, x : B_1 \vdash M_1 : C \to A$ and $\Gamma \vdash N : B_1$. By induction hypothesis, we also have B_2 such that $\Gamma, x : B_2 \vdash M_2 : C$ and $\Gamma \vdash N : B_2$. Let B be $B_1 \cap B_2$.

Case $(\cap E)$. Similar to Case $({\to}E)$.

Case (Ω). Let B be Ω. □

In order to have this lemma, we need intersection types and the Ω type. The subject expansion property is proved by using this lemma.

Proposition 6.6 (Subject Expansion). $\Gamma \vdash M : A$ and $M' \to_\beta M$ implies $\Gamma \vdash M' : A$.

Proof. Induction on the proof. Consider cases according to the last rule. We will discuss only interesting cases.

Case M' is the redex. Let M' be $(\lambda x.L)N$ and M be $L[x := N]$. By Lemma 6.5, there is a type B such that $\Gamma, x : B \vdash L : A$ and $\Gamma \vdash N : B$. Then we have $\Gamma \vdash \lambda x.L : B \to A$ and hence we have $\Gamma \vdash M' : A$.

Case $(\to I)$ and M' is not the redex. We can put $M' = \lambda x.N'$, $M = \lambda x.N$, and $N' \to_\beta N$. By induction hypothesis for $\Gamma, x : A \vdash N : B$, we have $\Gamma, x : A \vdash N'$. Hence $\Gamma \vdash M' : A$ holds.

Case $(\to E)$ and M' is not the redex. We can put $M' = N'L'$, $M = NL$. The claim follows from induction hypothesis.

Cases $(\cap I)$, $(\cap E_1)$, and $(\cap E_2)$. By induction hypothesis.

Case (Ω). The claim holds trivially.

Case $(p_n I)$ and M' is not the redex. We can put $M' = \lambda z.N'$, $M = \lambda z.N$, and $N' \to_\beta N$. By induction hypothesis, $\Gamma, z : A \vdash N' : B$ holds. Hence we have $\Gamma \vdash M' : p_n$. □

Remark. We will use the subject reduction property and the subject expansion property for proving soundness and completeness respectively.

Definition 6.7. right(A) is defined as the rightmost type constant in A. HN is defined to be the set of head normalizing terms. $X \to Y$ is defined as the set $\{M \in \Lambda | \forall N \in X(MN \in Y)\}$ for sets $X, Y \subseteq \Lambda$.

The interpretation $[\![A]\!]$ of a type A is defined by:

$$
\begin{aligned}
&[\![q_n]\!] = [\![p_{n+1}]\!] = \mathrm{HN} \qquad (n \geq 0),\\
&[\![\Omega]\!] = \Lambda,\\
&[\![A \to B]\!] = [\![A]\!] \to [\![B]\!],\\
&[\![A \cap B]\!] = [\![A]\!] \cap [\![B]\!].
\end{aligned}
$$

The interpretation $[\![\sim_n]\!]$ is defined by

$$
\begin{aligned}
&[\![\sim_0]\!] = \{(\Lambda, \Lambda)\},\\
&[\![\sim_{n+1}]\!] = p(\mathrm{HN}) \times p(\mathrm{HN}).
\end{aligned}
$$

Example. right$(A \to B \to q_0, (C \to q_1) \cap (C \to q_2)) = \{q_0, q_2\}$.

Lemma 6.8. (1) $x\vec{M}$ is in $[\![A]\!]$.

(2) right$(A) \neq \Omega$ implies $[\![A]\!] \subseteq \mathrm{HN}$.

(3) $[\![A]\!]$ is closed under $=_\beta$.

Proof. (1) By induction on A.

(2) By induction on A.

Case $A \to B$. Suppose right$(A \to B) \neq \Omega$ and $M \in [\![A \to B]\!]$. By (1) we have $x \in [\![A]\!]$. Hence Mx is in $[\![B]\!]$. By induction hypothesis for B, we have $Mx \in \mathrm{HN}$. Hence M is in HN.

Case $A \cap B$. From right$(A \cap B) \neq \Omega$, we have right$(B) \neq \Omega$. By induction hypothesis for B, we have $[\![B]\!] \subseteq \mathrm{HN}$. Hence we get $[\![A \cap B]\!] \subseteq \mathrm{HN}$.

(3) By induction on A. □

Definition 6.9. A variable assignment ρ is defined by $\rho : \mathrm{Vars} \to \Lambda$. A variable assignment $\rho[x := M]$ is defined by $(\rho[x := M])(x) = M$ and $(\rho[x := M])(y) = \rho(y)$ if x is not y. The interpretation $[\![M]\!]\rho$ of a term M with ρ is defined as $M[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n)]$ where $\mathrm{FV}(M) = \{x_1, \ldots, x_n\}$.

Lemma 6.10. If we have $\overrightarrow{x : B} \vdash M : A$ and $\rho(x_i) \in [\![B_i]\!]$ $(\forall i)$, then we have $[\![M]\!]\rho \in [\![A]\!]$.

Proof. It is proved by induction on the proof. We consider cases according to the last rule. We will show only interesting cases.

Case $(\to I)$. Assume $N \in [\![A]\!]$. We will show $[\![\lambda x.M]\!]\rho N \in [\![B]\!]$. Let ρ' be $\rho[x := N]$. By induction hypothesis, we have $[\![M]\!]\rho' \in [\![B]\!]$. Since we have $[\![\lambda x.M]\!]\rho N \to_\beta [\![M]\!]\rho'$, from Lemma 6.8 (3), we get $[\![\lambda x.M]\!]\rho N \in [\![B]\!]$. Hence $[\![\lambda x.M]\!]\rho$ is in $[\![A \to B]\!]$.

Cases $(\to E)$, $(\cap I)$, $(\cap E_i)$ are proved by induction hypothesis.

Case (Ω) is proved by $[\![\Omega]\!] = \Lambda$.

Case (Ip_n). The proof is

$$\frac{\Gamma, z : A \vdash M : B \quad A \sim_n B}{\Gamma \vdash \lambda z.M : p_n}.$$

Let ρ' be $\rho[z := z]$. By Lemma 6.8 (1), $\rho'(z)$ is in $[A]$. By induction hypothesis, we have $M\rho' \in [B]$. Since $n > 0$ and $A \sim_n B$, we have right$(B) \neq \Omega$. By Lemma 6.8 (2), we have $[B] \subseteq$ HN. Since $(\lambda z.M)\rho$ is $\lambda z.M\rho'$, we have $(\lambda z.M)\rho \in$ HN. □

Proposition 6.11. If $\overrightarrow{x : B} \vdash M : A$ and right$(A) \neq \Omega$, M is head normalizing.

Proof. Let $\rho(x) = x$. By Lemma 6.8 (1), $\rho(x_i) = x_i$ is in $[B_i]$. By Lemma 6.10, we have $M\rho \in [A]$. By Lemma 6.8 (2) we have $[A] \subseteq$ HN. Since $M\rho = M$, we have $M \in$ HN. □

Definition 6.12. We define the set core(A) of type constants for a type A by induction on A by

$$\begin{aligned}
&\mathrm{core}(c) = \{c\} \qquad (c = q_n, p_n, \Omega),\\
&\mathrm{core}(A \to B) = \mathrm{core}(B),\\
&\mathrm{core}(A \cap B) = \mathrm{core}(A) \cup \mathrm{core}(B).
\end{aligned}$$

We will also write core$(A_1, \ldots, A_n)$ and core$(x_1 : A_1, \ldots, x_n : A_n)$ for $\cup\{\mathrm{core}(A_i) | 1 \leq i \leq n\}$.

Example. core$(A \to B \to q_0, (C \to q_1) \cap (C \to q_2)) = \{q_0, q_1, q_2\}$.

Lemma 6.13. If $\Gamma, x : A \vdash x\vec{M} : B$, then core$(A) \supseteq$ core(B).

Proof. By induction on the proof. □

Lemma 6.14. If $A \sim_n B$ and $\Gamma, z : A \vdash M : B$ are provable and core$(\Gamma) \cap (\mathrm{TC}(A, B) - \{\Omega\}) = \phi$, then M is in PS$_n(z)$.

Proof. By induction on n.

Case $n = 0$. The claim holds since PS$_0(z) = \Lambda$.

Case $n + 1$. Let A be $B_{\pi(1)} \to \ldots \to B_{\pi(m)} \to q_k$ and B be $A_1 \to \ldots \to A_m \to q_k$. By Proposition 6.11, M is head normalizing. Let $M \to_\beta^* \lambda\vec{x}.y\vec{M}$ and $\vec{x}$ be $x_1, \ldots, x_l$.

By Proposition 6.4, we have $\Gamma, z : A \vdash \lambda\vec{x}.y\vec{M} : B$. By Lemma 6.2 (2), we have $\Gamma, z : A, \overrightarrow{x : A} \vdash y\vec{M} : C$ where we put $C = A_{l+1} \to \ldots \to A_m \to q_k$.

Since core$(\Gamma) \cap (\mathrm{TC}(A, B) - \{\Omega\}) = \phi$, we have $q_k \notin$ core(Γ) and hence y is z or x_i from Lemma 6.13. Since $q_k \notin \mathrm{TC}(A_i, B_i) - \{\Omega\}$, we have $q_k \notin$ core(A_i) and hence y is z from Lemma 6.13. Then we have $\Gamma, z : A, \overrightarrow{x : A} \vdash z\vec{M} : C$.

If $m = 0$, then the claim holds since $l = 0$ and $z \in$ PS$_n(z)$. Suppose $m > 0$. By Lemma 6.2 (4), the length of $\vec{M}$ is l and we have $B_{\pi(i)} = A_i \quad (l + 1 \leq i \leq m)$ and $\Gamma, z : A, \overrightarrow{x : A} \vdash M_i : B_i \quad (1 \leq i \leq l)$ where we put $z\vec{M} = zM_{\pi(1)} \ldots M_{\pi(l)}$.

By definition $M \in$ PS$_1(z)$ iff $M \to_\beta^* \lambda\vec{x}.z\vec{M}$ and the lengths of $\vec{x}$ and $\vec{M}$ are the same. Hence we have $M \in$ PS$_1(z)$. If $n = 1$, the claim immediately follows from this.

Suppose $n > 1$. When $i \neq j$, core$(A_i) \neq$ core(B_j) holds since these cores are some type constants other than Ω and $\mathrm{TC}(A_i, B_i) - \{\Omega\}$ and $\mathrm{TC}(A_j, B_j) - \{\Omega\}$ are disjoint. From $B_{\pi(i)} = A_i \quad (l+1 \leq i \leq m)$ we have $\pi(i) = i$ for $l + 1 \leq i \leq m$. Therefore π is in $\mathcal{S}_l$.

Since we already have $A_i \sim_n B_i$, $\Gamma, z : A, \overrightarrow{x : A} \vdash M_i : B_i$, and core$(\Gamma, z : A, x_j : A_j \quad (i \neq j)) \cap (\mathrm{TC}(A_i, B_i) - \{\Omega\}) = \phi$, by induction hypothesis for n, we have $M_i \in$ PS$_n(x_i)$. From $M \to_\beta^* \lambda x_1 \ldots x_l.zM_{\pi(1)} \ldots M_{\pi(l)}$, $\pi \in \mathcal{S}_l$, and $M_i \in$ PS$_n(x_i)$, we have $M \in$ PS$_{n+1}(z)$. □

Lemma 6.15. $\vdash \lambda z.M : [p_n]$ implies $M \in$ PS$_n(z)$.

Proof. Induction on the proof. Consider cases according to the last rule.

We do not have cases (Ass), $(\to I)$, $(\to E)$, nor (Ω).

Cases $(\cap I)$ and $(\cap E_i)$ are proved by induction hypothesis.

Case (Ip_n). The proof is

$$\frac{z : A \vdash M : B \quad A \sim_n B}{\vdash \lambda z.M : p_n}.$$

Let Γ be ϕ in Lemma 6.14, we have $M \in$ PS$_n(z)$. □

Proposition 6.16 (Soundness). If $\vdash M : p_n$ for all n, then M is in HP.

Proof. By Proposition 6.11, we have $M \to_\beta^* \lambda x_1 \ldots x_l.y\vec{M}$. By Proposition 6.4, we have $\vdash \lambda x_1 \ldots x_l.y\vec{M} : p_n$.

If $l = 0$, we have contradiction from Lemma 6.2 (1) and (3). So we have $l > 0$. Let $\lambda z.N$ be $\lambda x_1 \ldots x_l.y\vec{M}$.

We have $\vdash \lambda z.N : p_n$ for all n. By Lemma 6.15, we have $N \in \mathrm{PS}_n(z)$ for all n. By Lemma 4.2 (2), we have $\lambda z.N \in$ HP. Hence M is in HP. □

7 Completeness

We will show the completeness and finish the proof of the characterization theorem.

Lemma 7.1. If $M \in \mathrm{PS}_n(z)$, there are A and B such that $z : A \vdash M : B$ and $A \sim_n B$.

Proof. Induction on n.

Case $n = 0$. Take Ω for A, B.

Case $n + 1$. Suppose $M \to_\beta^* \lambda x_1 \ldots x_m.zM_{\pi(1)} \ldots M_{\pi(m)}$, $\pi \in \mathcal{S}_m$, and $M_i \in \mathrm{PS}_n(x_i)$.

Induction hypothesis for n, we have A_i and B_i such that $x_i : A_i \vdash M_i : B_i$ and $A_i \sim_n B_i$. By choosing appropriate q_l for each (A_i, B_i), we can suppose that $\mathrm{TC}(A_i, B_i) - \{\Omega\} \quad (1 \le i \le m)$ are disjoint.

Let q_k be fresh in A_i, B_i and $A = B_{\pi(1)} \to \ldots \to B_{\pi(m)} \to q_k$ and $B = A_1 \to \ldots \to A_m \to q_k$. Then we have $A \sim_{n+1} B$.

Then we have $z : A, \overrightarrow{x : A} \vdash zM_{\pi(1)} \ldots M_{\pi(m)} : q_k$. Hence we get $z : A \vdash \lambda\vec{x}.zM_{\pi(1)} \ldots M_{\pi(m)} : B$. By Proposition 6.6, we have $z : A \vdash M : B$. □

Proposition 7.2 (Completeness). If $M \in$ HP, then we have $\vdash M : p_n$.

Proof. Let $M \to_\beta^* \lambda z.N$. By Lemma 4.2 (2), we have $N \in \mathrm{PS}_n(z)$ for all $n > 0$. By Lemma 7.1, there are A and B such that $z : A \vdash N : B$ and $A \sim_n B$. By the rule (Ip_n), we get $\vdash \lambda z.N : p_n$. By Proposition 6.6, we have $\vdash M : p_n$. □

Now we complete the proof of the characterization theorem.

Proof of Theorem 5.2. The implication from the right-hand side to the left-hand side is proved by Proposition 6.16. The implication form the left-hand side to the right-hand side is proved by Proposition 7.2. □

Example. Let

$$
\begin{aligned}
&Y = (\lambda xy.y(xxy))(\lambda xy.y(xxy)),\\
&Q = Y(\lambda fxy.x(fy)),\\
&P = \lambda x_0.Qx_0.
\end{aligned}
$$

We have

$$
\begin{aligned}
&YF \to_\beta^* F(YF),\\
&Qx_0 \to_\beta^* \lambda x_1.x_0(Qx_1) \to_\beta^* \lambda x_1.x_0(\lambda x_2.x1(Qx_2)) \to_\beta^* \lambda x_1.x_0(\lambda x_2.x1(\lambda x_3.x_2(Qx_3))) \to_\beta^* \ldots.
\end{aligned}
$$

Then BT(P) is $\begin{array}{c} \lambda x_0 x_1.x_0 \\ | \\ \lambda x_2.x_1 \\ | \\ \lambda x_3.x_2 \\ | \\ \vdots \end{array}$ and P is a hereditary permutator.

Let

$$
\begin{aligned}
&A_0 = \Omega,\\
&A_{n+1} = A_n \to q_{n+1}.
\end{aligned}
$$

Then $\{A_n \to A_n | n > 0\}$ is the set of types for P. That is, we have $\vdash P : A_n \to A_n$ for all n. On the other hand, we can show that if $\vdash M : A_n \to A_n$ for all n, then M β-equals to P or the finite hereditary permutators P_m defined by

$$P_0 = \lambda z.z,$$
$$P_{n+1} = \lambda z x_1.z(P_n x_1).$$

Acknowledgments

We would like to thank Professor Koji Nakazawa for comments on a draft of this paper.

References

[1] J.A. Bergstra and J.W. Klop, Invertible terms in the lambda calculus, *Theoretical Computer Sicence* **11** (1980) 19–37.

[2] K. Bruce, R. Di Cosmo, and G. Longo, Provable isomorphisms of types, *Mathematical Structures in Computer Science* **2** (2) (1992) 231–247.

[3] M. Dezani-Ciancaglini, Characterization of normal forms possessing inverse in the lambda-beta-eta-calculus, *Theoretical Computer Science* **2** (1976) 323–337.

[4] M. Dezani-Ciancaglini, F. Honsell, and Y. Motohama, Compositional Characterization of λ-terms using Intersection Types, *Theoretical Computer Science* **340** (3) (2005) 459–495.

[5] R. Hasegawa, L. Paolini, and P. Urzyczyn, The TLCA list of Open Problems, `http://tlca.di.unito.it/opltlca/`, 2007.

[6] J.R. Hindley and J.P. Seldin, *Introduction to Combinators and λ-Calculus* (Cambridge University Press, Cambridge, 1986).

[7] M. Tatsuta and M. Dezani-Ciancaglini, Normalisation is Insensible to lambda-term Identity or Difference, In: Proceedings of Twenty First Annual IEEE Symposium on Logic in Computer Science (2006) 327–336.