Types for Hereditary Head Normalizing Terms

Makoto Tatsuta

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

Abstract. Klop's Problem is finding a type for characterizing hereditary head normalizing terms, that is, lambda-terms whose Böhm trees do not contain the bottom. This paper proves that this problem does not have any solution by showing that the set of those terms is not recursively enumerable. This paper also gives a best-possible solution by providing an intersection type system with a countably infinite set of types such that typing in all these types characterizes hereditary head normalizing terms. By using the same technique, this paper also shows that the set of lambda-terms normalizing by infinite reduction is not recursively enumerable.

1 Introduction

Klop's Problem [4] is finding a type that characterizes the set of hereditary head normalizing terms, that is, λ-terms whose Böhm trees do not contain the bottom. 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 hereditary head normalizing. When we study infinite computation by using λ-calculus, Böhm trees give computational meaning to non-normalizing λ-terms. Then the bottom means non-informative computation, and a hereditary head normalizing term is a nice term that does not contain any meaningless computation. It is an interesting subject to find a type-theoretic characterization of terms having a nice property [3,9]. So finding a type-theoretic characterization of hereditary head normalizing terms is an important subject. Several researchers in European theoretical computer science community have been trying to answer this question for several years [4].

We will show that Klop's Problem does not have any solution, and give a best-possible solution. By using the same technique used to show the first claim, we will also show that the set of λ-terms normalizing by infinite reduction is not recursively enumerable.

To show the first claim, we will show that the set of hereditary head normalizing terms is not recursively enumerable. For each unary primitive recursive function f, we will construct a λ-term whose Böhm-tree computes the values $f(0), f(1), f(2), \ldots$ so that the term is hereditary head normalizing if and only if f is positive, that is, $f(x) > 0$ for all x. The set of positive primitive recursive functions is proved to be not recursively enumerable since it solves the halting problem.

J. Garrigue and M. Hermenegildo (Eds.): FLOPS 2008, LNCS 4989, pp. 195–209, 2008.

To show the second claim, we will think the set HN_n of λ-terms whose Böhm trees do not contain the bottom at depth less than n, and we will give some type p_n that characterizes HN_n. We use axioms for type constants and type preorder so that our intended model will interpret those constants by the least fixed points of those axioms. We will also need intersection types because of subject expansion [1]. Then we have a characterization of hereditary head normalizing terms by the set of the types p_n because a λ-term is hereditary head normalizing if and only if this term has the type p_n for all n.

A formal system for infinite computation has been studied actively [10]. Infinite λ-calculus [6,2] is an important subject among those formal systems. Infinite λ-calculus gives a theoretical foundation to infinite computation in functional programming languages, as finite λ-calculus did for finite computation in functional programming languages.

The set of λ-terms normalizing by infinite reduction is a fundamental notion in infinite λ-calculus. By applying the tree construction for the first claim to infinite λ-calculus, for each primitive recursive function f, we will construct a λ-term whose reducts compute the values $f(0), f(1), f(2), \ldots$ so that the term has a normal form by infinite reduction if and only if f is positive. By this, we can show that the set of λ-terms normalizing by infinite reduction is not recursively enumerable.

There have been two papers [8] and [7] on Klop's Problem. [8] gave a sufficient condition for hereditary head normalizing terms by using a type system. [7] gave the type system $\Lambda\vee_\star$ and the types a_n that characterize HN_n. We found our system independently from his paper and our system is simpler than his system. Though his system was another best-possible solution, the relationship between his result and the solution of Klop's Problem has not been discussed, because we did not know that there was no solution for Klop's Problem.

Section 2 defines hereditary head normalizing terms, and describes Klop's Problem. Section 3 shows that the set HHN of hereditary head normalizing terms is not recursively enumerable, and there does not exist any solution for Klop's Problem. We will discuss set theoretic properties of HHN in Section 4. Section 5 gives the set of types that characterizes HHN. The soundness is shown in Section 6 and the completeness is proved in Section 7. In Section 8, we discuss infinite λ-calculus and prove that the set of λ-terms normalizing by infinite reduction is not recursively enumerable.

2 Hereditary Head Normalizing Terms

In this section, we will give Klop's Problem 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. $M = N$ denotes the syntactical equality modulo renaming bound variables. Vars is the set of variables and 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 $M \to_\beta^* N$ 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 to 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 $M \to_\beta N$.

One-step head reduction $M \to_h N$ is defined by:

$$(head) \quad \lambda x_1 \dots x_n.(\lambda y.M)NN_1 \dots N_m \to_h \lambda x_1 \dots x_n.M[y := N]N_1 \dots N_m.$$

Head reduction $M \to_h^* N$ is defined as the reflexive transitive closure of the relation $\to_h$. We will write $M_0 \to_h^n M_n$ for some n-step head reduction sequence $M_0 \to_h M_1 \to_h \dots \to_h M_n$. The relation $M \to_i N$ is defined to hold if $M \to_\beta N$ holds and $M \to_h N$ does not hold. Inner reduction $\to_i^*$ is defined as the reflexive transitive closure of the relation $\to_i$. A λ-term M is called *head normal* if there is not any λ-term N such that $M \to_h N$. A λ-term M is called *head normalizing* if there is some head normal term N such that $M \to_\beta^* N$.

Example. $\lambda w.(\lambda x.y)x((\lambda z.z)z) \to_h \lambda w.y((\lambda z.z)z)$ holds, but $\lambda w.(\lambda x.y) x((\lambda z.z)z) \to_h \lambda w.(\lambda x.y)xz$ does not hold. $\lambda w.y((\lambda z.z)z)$ is a head normal form. $\lambda w.(\lambda x.y)xz$ is not a head normal form, but it is head normalizing since it reduces to the head normal form $\lambda w.yz$. $(\lambda x.xx)(\lambda x.xx)$ is not head normalizing.

Remark

(1) $M \to_h N$ implies $M \to_\beta N$.

(2) If M is head normalizing, then there is a head normal form N such that $M \to_h^* N$.

(3) M is a head normal form if and only if M is of the shape $\lambda x_1 \dots x_n.yN_1 \dots N_m$.

(4) M is a normal form if and only if M is of the shape $\lambda x_1 \dots x_n.yN_1 \dots N_m$ where N_i is a normal form for all i.

Definition 2.2 (Böhm Trees)
We suppose $\perp$ is a constant. A Böhm tree is defined as a (possibly infinite) tree with node labels in $\{\lambda x_1 \dots x_n.y | x_1, \dots, x_n, y \in \text{Vars}\} \cup \{\perp\}$. Böhm tree BT($M$) of a λ-term M is defined by:

(1) $\mathrm{BT}(M) = \perp$ if M is not head normalizing,

(2) BT(M) is

$$\begin{array}{ccc} & \lambda x_1 \dots x_n.y & \\ \swarrow & \cdots & \searrow \\ \mathrm{BT}(M_1) & & \mathrm{BT}(M_m) \end{array},$$

if $M =_\beta \lambda x_1 \dots x_n.yM_1 \dots M_m$.

We will sometimes write $[\lambda x_1 \dots x_n.y, \mathrm{BT}(M_1), \dots, \mathrm{BT}(M_m)]$ for $\mathrm{BT}(M)$ given in (2), for saving space.

Remark
(1) If $M =_\beta N$, then $\mathrm{BT}(M) = \mathrm{BT}(N)$.
(2) $\mathrm{BT}(M) = \perp$ if and only if M is not head normalizing.

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

Definition 2.3 (Hereditary Head Normalizing). A λ-term M is called *hereditary head normalizing* if $\mathrm{BT}(M)$ does not contain $\perp$. HHN is defined to be the set of hereditary head normalizing λ-terms.

Klop's Problem [4] is finding a type that characterizes the set of hereditary head normalizing terms. 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 hereditary head normalizing.

We will answer this question. First, in Section 3 we will show that the set of hereditary head normalizing terms is not recursively enumerable. Hence we will conclude that there does not exist any type that characterizes hereditary head normalizing terms 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 systems with a countably infinite set of types which characterizes hereditary head normalizing terms.

3 Non-existence of a Type for HHN

We will show that there does not exist any type that characterizes the set HHN of hereditary head normalizing terms. First we will show the set PPR of indices of positive primitive recursive functions is not recursively enumerable, by using a diagonal argument. We will define the λ-term T so that $T\overline{e}\overline{0}$ checks if $\{e\}^{pr}(n) > 0$ for each n one by one, and show that $e \in \mathrm{PPR}$ iff $T\overline{e}\overline{0} \in \mathrm{HHN}$. Combining those, we will prove HHN is not recursively enumerable.

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

First, we give several notations for primitive recursive functions.

Definition 3.1. $\langle x, y\rangle$ denotes the standard primitive recursive surjective pairing, and $\pi_0(x)$ and $\pi_1(x)$ are the first and second projections respectively.

We fix some listing of all the unary primitive recursive functions so that the n-th unary primitive recursive function is effectively obtained from n.

For a number n, $\{n\}^{pr}(x)$ is defined as the n-th unary primitive recursive function. 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. $\pi_0(\langle n, m\rangle) = n$ and $\pi_1(\langle n, m\rangle) = m$ hold. For a total function $g(x)$, $\mu y.(g(y) = 0)$ returns n if $g(y) = 0$ for some y such that $g(n) = 0$ and $g(x) > 0$ for all $x < n$, and is undefined if $g(x) > 0$ for all x. 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 $F\overline{m_1} \ldots \overline{m_n} =_\beta \overline{m}$ for all $m, m_1, \ldots, m_n \in N$.

Theorem 4.15 in Page 53 in [5] showed the following claim.

Theorem 3.3 ([5]). *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 indices for positive primitive recursive functions.

Proposition 3.5. *The set* PPR *is not recursively enumerable.*

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

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

Define a partial function $f : N \to N$ by $f(x) = 1$ if $Q(\pi_0(x), P(x))$ 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 $Q(\pi_0(x), P(x))$ is in PPR by the definition of f, iff $\forall y(\{\pi_0(x)\}^{pr}(\{P(x)\}^{pr}(y)) > 0)$ by the definition of Q 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 by the definition of the x-th unary partial recursive function.

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 leads to contradiction.

Consequently, the set PPR is not recursively enumerable. □

We define

$$
\begin{aligned}
&\mathrm{S} = \lambda y f x.f(y f x),\\
&Y_0 = \lambda x y.y(x x y),\\
&Y = Y_0 Y_0,\\
&\Delta = \lambda x.x x.
\end{aligned}
$$

The term S is the successor for Church numerals. Y is Turing's fixed point operator and we have $YM \to_\beta^* M(YM)$.

We now prove the first main theorem by using PPR.

Theorem 3.6. *The set* HHN *of hereditary head normalizing terms is not recursively enumerable.*

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

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

Fix variables a and w. Define T by

$$T = Y(\lambda txy.Uxy(\lambda w.a)(\Delta\Delta)(tx(\mathrm{S}y))).$$

We show that $T\overline{e}\overline{0}$ is in HHN if and only if e is in PPR. The direction from the right to the left is proved by $\mathrm{BT}(T\overline{e}\overline{0}) = [a, [a, [a, [\ldots]]]]$ when e is in PPR, and hence $T\overline{e}\overline{0}$ is in HHN. In order to show the direction from the left to the right, first we assume e is not in PPR and will show $T\overline{e}\overline{0}$ is not in HHN. 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 $\mathrm{BT}(T\overline{e}\overline{0}) = [\overbrace{a, [a, [\ldots [a}^{m_0}, \bot]\ldots]]]$, and hence $T\overline{e}\overline{0}$ is not in HHN.

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

Non-existence of solutions for Klop's Problem follows immediately from the previous theorem.

Theorem 3.7. *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 head normalizing terms is the same as $\{M | \Gamma \vdash M : A$ is provable in T for some $\Gamma\}$.*

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

4 Set-Theoretic Properties of HHN

We will discuss some set-theoretic properties of hereditary head normalizing terms. We will define the sets HN_n and Q_n as the set of λ-terms whose Böhm trees do not contain ⊥ at depth $< n$, and the set of terms in HN_n such that the roots of their Böhm trees do not contain abstractions, respectively. We will show some set-theoretic inequality for those, which will be used in Section 6.

We will write $A^n \to B$ for $A \to \ldots \to A \to B$ (n times of A).

Proposition 4.1. HHN *is the greatest fixed point $X \subseteq \Lambda$ of the equation:*

(1) if $M \in X$, then there exist $N_1, \ldots, N_m \in X$ ($m \geq 0$) such that $M =_\beta \lambda x_1 \ldots x_n.yN_1 \ldots N_m$.

Proof. First, we will show HHN is a solution of (1). Suppose M in HHN. Then M has a head normal form $\lambda\vec{x}.y\vec{N}$ such that M reduces to $\lambda\vec{x}.y\vec{N}$. Then $\mathrm{BT}(N_i)$ does not contain $\perp$. Therefore N_i is in HHN.

Secondly, we will show HHN is the greatest solution of (1). Assume X is a solution. We will show $X \subseteq \mathrm{HHN}$. Assume $M \in X$. We will show that any node at depth $\leq k$ in $\mathrm{BT}(M)$ is not $\perp$ by induction on k.

Case $k = 0$. Since M has a head normal form, the node is not $\perp$.

Case $k + 1$. We have $M =_\beta \lambda\vec{x}.y\vec{N}$, $N_i \in X$. By induction hypothesis, a node at depth $\leq k$ in $\mathrm{BT}(N_i)$ is not $\perp$. Hence a node at depth $\leq k+1$ in $\mathrm{BT}(\lambda\vec{x}.y\vec{N})$ is not $\perp$. Then a node at depth $\leq k+1$ in $\mathrm{BT}(M)$ is not $\perp$.

Therefore a node at depth $\leq k$ in $\mathrm{BT}(M)$ is not $\perp$ for any k. Hence we have $M \in \mathrm{HHN}$. □

Definition 4.2. HN_0 is the set Λ.

HN_{k+1} is the set $\{M | M \to_\beta^* \lambda\vec{x}.y\vec{N}, N_i \in \mathrm{HN}_k\}$.

Remark

(1) $M \in \mathrm{HN}_k$ iff any node at depth $< k$ in $\mathrm{BT}(M)$ is not $\perp$.
(2) $\mathrm{HN}_{k+1} \subseteq \mathrm{HN}_k$.
(3) $\mathrm{HHN} = \cap_{k=0}^{\infty}\mathrm{HN}_k$.

Definition 4.3. Q_n is defined as $\{M | M \to_\beta^* x\vec{N}, x \in \mathrm{Vars}, N_i \in \mathrm{HN}_{n-1}\}$ for $n \geq 1$.

$X \to Y$ is defined as $\{M | \forall N \in X(MN \in Y)\}$, for $X, Y \subseteq \Lambda$.

Remark. $Q_n \subseteq \mathrm{HN}_n$.

Lemma 4.4

(1) HN_n *is closed under* $\to_\beta$.
(2) Q_n *is closed under* $\to_\beta$.
(3) $M \in Q_n$ *and* $M \to_\beta^* x\vec{N}$ *imply* $N_i \in \mathrm{HN}_{n-1}$.

Proof.

(1) By induction on n. Case $n+1$. Suppose $M \in \mathrm{HN}_{n+1}$ and $M \to_\beta N$. We have $M \to_\beta^* \lambda\vec{x}.y\vec{L}$, $L_i \in \mathrm{HN}_n$. By Church-Rosser, we have $N \to_\beta^* \lambda\vec{x}.y\vec{L}'$, $L_i \to_\beta^* L'_i$. By induction hypothesis, $L'_i \in \mathrm{HN}_n$ holds. Then we have $N \in \mathrm{HN}_{n+1}$.
(2) Suppose $M \in Q_n$ and $M \to_\beta N$. We have $M \to_\beta^* x\vec{L}$, $L_i \in \mathrm{HN}_{n-1}$. By Church-Rosser, we have $N \to_\beta^* x\vec{L}'$ and $L_i \to_\beta^* L'_i$. By (1), we have $L'_i \in \mathrm{HN}_{n-1}$. Therefore we have $N \in Q_n$.
(3) By (2), we have $x\vec{N} \in Q_n$. Then we have $x\vec{N} \to_\beta^* x\vec{L}$ and $L_i \in \mathrm{HN}_{n-1}$. Hence $N_i \to_\beta^* L_i$ and $N_i \in \mathrm{HN}_{n-1}$. □

Proposition 4.5

(1) $Q_{n+1} \subseteq Q_n \quad (n \geq 1)$.
(2) $Q_{n+1} \subseteq \mathrm{HN}_n \to Q_{n+1} \quad (n \geq 0)$.
(3) $\mathrm{HN}_n \supseteq Q_n^m \to Q_n \quad (n \geq 1, m \geq 0)$.

Proof.

(1) From $\mathrm{HN}_n \subseteq \mathrm{HN}_{n-1}$.
(2) Suppose $M \in Q_{n+1}$ and $N \in \mathrm{HN}_n$. We have $M \rightarrow_\beta^* x\vec{L}$, $L_i \in \mathrm{HN}_n$. Hence $MN \rightarrow_\beta^* x\vec{L}N$. Then we have $MN \in Q_{n+1}$.
(3) Assume $M \in Q_n^m \rightarrow Q_n$. By Definition 4.3, we have $x_i \in Q_n$. Then $M\vec{x}^{(m)} \in Q_n$ holds where $\vec{x}^{(m)}$ is $x_1, \ldots, x_m$. Hence M is head normalizing and we have $M \rightarrow_\beta^* \lambda\vec{x}^{(l)}.y\vec{N}$. If $m < l$, then we have $M\vec{x}^{(m)} \rightarrow_\beta^* \lambda x_{m+1} \ldots x_l.y\vec{N}$ and $M\vec{x}^{(m)} \rightarrow_\beta^* z\vec{L}$ does not hold by Church-Rosser, which contradicts to $M\vec{x}^{(m)} \in Q_n$. Hence we get $m \geq l$. Then we have $M\vec{x}^{(m)} \rightarrow_\beta^* y\vec{N}x_{l+1} \ldots x_m$. By $M\vec{x}^{(m)} \in Q_n$, from Lemma 4.4 (3), we have $N_i \in \mathrm{HN}_{n-1}$. Hence $M \in \mathrm{HN}_n$. □

5 Types for HHN

This section will present a type system with a countably infinite set of types which characterizes hereditary head normalizing terms.

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

We have type constants p_n, q_m, q, and Ω $(n \geq 0, m \geq 1)$. Types $A, B, \ldots$ are defined by:

$A, B, \ldots ::= p_n | q_m | q | \Omega | A \rightarrow A | A \cap A \quad (n \geq 0, m \geq 1)$.

Type preorder $A \leq B$ is defined by:

$$A \leq A \qquad \frac{A \leq B \quad B \leq C}{A \leq C} \qquad A \cap B \leq A \qquad A \cap B \leq B$$

$$p_0 = \Omega$$
$$q_{n+1} \leq p_n \rightarrow q_{n+1} \quad (n \geq 0)$$
$$p_n \geq q_n^m \rightarrow q_n \quad (n \geq 1, m \geq 0)$$
$$q_{n+1} \leq q_n \quad (n \geq 1)$$
$$q_n \geq q \quad (n \geq 1)$$

A type declaration is a finite set of $x : A$ where x is a variable and A is a type. 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:

$$\frac{}{\Gamma, x : A \vdash x : A}\,(Ass)$$

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

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

Notation. We will write $\{x_1, \ldots, x_n\} : A$ for $\{x_1 : A, \ldots, x_n : A\}$.

This is a standard intersection type system except for constants p_n, q_n, q. Our intended meaning of the constants p_n, q_n, and q are the set HN_n, the set Q_n, and $\cap_n Q_n$ respectively. Our discussion will also go well in the same way when we add other set-theoretically-sound rules for the type preorder such as $A \cap B \leq B \cap A$. Intersection types are necessary since we need the subject expansion property in our proof.

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

Theorem 5.2. *M is hereditary head normalizing if and only if* $\mathrm{FV}(M) : q \vdash M : p_n$ *is provable in the type theory $\mathcal{T}$ 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.

[7] gave a similar type system to our system $\mathcal{T}$ so that his type system is an intersection type system with type constants a_n, b_n, c_n and type preorder, each type a_n characterizes HN_n, and the set of the types a_n characterizes HHN. We found our system independently from his work. Our system is simpler because we used only two sets of type constants p_n, q_n with five axioms for $\leq$, while his system used three sets of type constants a_n, b_n, c_n with six axioms for $\leq$ and $\sim$.

6 Soundness

We will prove the soundness part of Theorem 5.2 by using set-theoretic semantics. We will interpret a type by a $=_\beta$-closed set of λ-terms. In particular, p_n is interpreted by HN_n. The soundness of this interpretation will be proved by induction on derivations.

Definition 6.1. The interpretation $[\![A]\!]$ of a type A is defined by:

$$
\begin{aligned}
&[\![p_n]\!] = \mathrm{HN}_n, \\
&[\![q_n]\!] = Q_n, \\
&[\![q]\!] = \cap_{n=1}^{\infty} Q_n, \\
&[\![\Omega]\!] = \Lambda, \\
&[\![A \to B]\!] = [\![A]\!] \to [\![B]\!], \\
&[\![A \cap B]\!] = [\![A]\!] \cap [\![B]\!],
\end{aligned}
$$

where $X \to Y = \{M \in \Lambda | \forall N \in X (MN \in Y)\}$ for $X, Y \subseteq \Lambda$.

Proposition 6.2. *$[\![A]\!]$ is closed under $=_\beta$.*

Proof. By induction on A. If A is either p_n, q_n, or q, the claim holds because their interpretations are closed under β-reduction by Lemma 4.4 (1) and (2), and their interpretations are also closed under β-expansion by their definitions. If A is Ω, the claim holds trivially. If A is $B \cap C$ or $B \to C$, the claim is proved by induction hypothesis. $\square$

Proposition 6.3. *If $A \le B$ is provable, then $[\![A]\!] \subseteq [\![B]\!]$ holds.*

Proof. By induction on the derivation of $A \le B$. Cases are considered according to the last rule.

The case $q_{n+1} \le p_n \to q_{n+1}$ is proved by Proposition 4.5 (2). The case $p_n \ge q_n^m \to q_n$ is proved by Proposition 4.5 (3). The case $q_{n+1} \le q_n$ is proved by Proposition 4.5 (1). □

Definition 6.4. 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 $\mathrm{FV}(M) = \{x_1, \ldots, x_n\}$.

Proposition 6.5 (Soundness of Interpretation). *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 Proposition 6.2, we get $[\![\lambda x.M]\!]\rho N \in [\![B]\!]$. Hence $[\![\lambda x.M]\!]\rho$ is in $[\![A \to B]\!]$.

Case $(\le)$ is proved by Proposition 6.3. □

Proposition 6.6 (Soundness). *M is hereditary head normalizing if $\mathrm{FV}(M) : q \vdash M : p_n$ is provable in the type theory $\mathcal{T}$ for all n.*

Proof. Suppose $\mathrm{FV}(M) : q \vdash M : p_n$. Define ρ by $\rho(x) = x$. We have $x \in [\![q]\!]$. By Proposition 6.5, we get $[\![M]\!]\rho \in [\![p_n]\!]$. Hence M is in HN_n for all n. Consequently M is in HHN. □

7 Completeness

We will show the completeness and finish the proof of the characterization theorem. First we will show the subject expansion property. By using this property, we will show the completeness of the set HN_n for the type p_n.

Lemma 7.1. *If $\Gamma \vdash M[x := N] : A$, then there exists some B such that $\Gamma \vdash N : B$ and $\Gamma, x : B \vdash M : A$.*

Proof. By induction on the proof. Cases are considered according to the last rule. We will show only interesting cases.

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

Case $(\to E)$ and $M = M_1 M_2$. Then we have

$$\frac{\Gamma \vdash M_1[x := N] : C \to A \quad \Gamma \vdash M_2[x := N] : C}{\Gamma \vdash (M_1 M_2)[x := N] : A}$$

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

Case $(\cap I)$. This case is proved similarly to Case $(\rightarrow E)$ by using an intersection type. □

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

Proposition 7.2 (Subject Expansion). *If $M \rightarrow_\beta N$ and $\Gamma \vdash N : A$, then we have $\Gamma \vdash M : A$.*

Proof. By induction on the proof. We consider cases according to the last rule. We will discuss only interesting cases.

Case $M = (\lambda x.M_1)M_2$ and $N = M_1[x := M_2]$. By Lemma 7.1, we have B such that $\Gamma \vdash M_2 : B$ and $\Gamma, x : B \vdash M_1 : A$. Then we get $\Gamma \vdash (\lambda x.M_1)M_2 : A$. □

Remark. Subject reduction also holds. However, we do not need subject reduction property for our proof.

Lemma 7.3. *If $M \in \mathrm{HN}_n$, then we have $\mathrm{FV}(M) : q_n \vdash M : p_n$.*

Proof. By induction on n.

Case $n = 0$. The claim holds since $p_0 = \Omega$.

Case $n+1$. Assume $M \in \mathrm{HN}_{n+1}$. We have $M \rightarrow_\beta^* \lambda \vec{x}.y\vec{N}$, $\mathrm{length}(\vec{x}) = m$, $\mathrm{length}(\vec{N}) = l$, and $N_i \in \mathrm{HN}_n$. By induction hypothesis for n, we have $\mathrm{FV}(N_i) : q_n \vdash N_i : p_n$. By $q_{n+1} \le q_n$, we have $\mathrm{FV}(M), \vec{x} : q_{n+1} \vdash N_i : p_n$. Then we have $y : q_{n+1}$ under the same type declaration. By using $q_{n+1} \le p_n \rightarrow q_{n+1}$ l times, we have $\mathrm{FV}(M), \vec{x} : q_{n+1} \vdash y\vec{N} : q_{n+1}$. Then we get $\mathrm{FV}(M) : q_{n+1} \vdash \lambda \vec{x}.y\vec{N} : q_{n+1}^m \rightarrow q_{n+1}$. By $q_{n+1}^m \rightarrow q_{n+1} \le p_{n+1}$, we have $\lambda \vec{x}.y\vec{N} : p_{n+1}$ under the same type declaration. By Proposition 7.2, we have $\mathrm{FV}(M) : q_{n+1} \vdash M : p_{n+1}$. □

By $q_{n+1} \ge q$ and this lemma, we get the next proposition.

Proposition 7.4 (Completeness). *If $M \in \mathrm{HN}_n$, then we have $\mathrm{FV}(M) : q \vdash M : p_n$.*

Remark. We have a similar property:

– If $M \in Q_{n+1}$, then we have $\mathrm{FV}(M) : q \vdash M : q_{n+1}$.

However, we do not need this for our proof.

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 6.6. The implication from the left-hand side to the right-hand side is proved by Proposition 7.4. □

8 Normalizing Terms in Infinite λ-Calculus

We will discuss infinite λ-calculus [6,2], and show that the set of normalizing λ-terms by infinite reduction is not recursively enumerable. We will define the

λ-term T such that $T\overline{e}\overline{0}$ checks if $\{e\}^{pr}(n) > 0$ for each n one by one, and show that $e \in$ PPR iff $T\overline{e}\overline{0}$ is normalizing by infinite reduction. Combining it with non-recursive enumerability of PPR shown in Section 3, we will prove the set of normalizing λ-terms by infinite reduction is not recursively enumerable.

Normalizing terms by infinite reduction are important in infinite λ-calculus since they play a role of values in the same way as normal forms do in usual λ-calculus. When we construct a formal system that describes infinite λ-calculus, we might want to have a system that also characterizes normalizing terms by infinite reduction. The theorem in this section shows that we cannot have such a system.

Definition 8.1. A *position*, denoted by u, is defined to be a finite string of positive integers.

For a λ-term M and a position u, the subterm $M|u$ of M is defined by induction on u by

$$\begin{array}{l} M|\langle\rangle = M, \\ (\lambda x.M)|1 \cdot u = M|u, \\ (MN)|1 \cdot u = M|u, \\ (MN)|2 \cdot u = N|u. \end{array}$$

The distance $d(M, N)$ for $M, N \in \Lambda$ is defined as 0 if $M = N$, and $\frac{1}{2^l}$ if $M \neq N$ and l is the minimum length of u such that $M|u$ and $N|u$ are both defined, and they are distinct variables, or of different syntactic types.

The set Λ^∞ is defined as $\{(M_0, M_1, M_2, \ldots) | M_i \in \Lambda, \forall \epsilon > 0 \exists n \forall i, j \geq n(d(M_i, M_j) < \epsilon)\}$ where $(M_0, M_1, M_2, \ldots)$ is a countably infinite sequence of λ-terms.

The equality $(M_0, M_1, M_2, \ldots) \equiv_\infty (N_0, N_1, N_2, \ldots)$ on Λ^∞ is defined by $\lim_{n\to\infty} d(M_n, N_n) = 0$.

Remark

(1) d is proved to be actually a distance [6].
(2) The quotient set $\Lambda^\infty / \equiv_\infty$ is the completion of Λ with the distance d.
(3) The tree $T(M)$ of a λ-term M is defined by

$$T(x) = x,$$

$$T(\lambda x.M) = \begin{array}{c} \lambda x \\ | \\ T(M) \end{array},$$

$$T(MN) = \begin{array}{ccc} & @ & \\ \diagup & & \diagdown \\ T(M) & & T(N) \end{array}.$$

We will identify a λ-term M and its tree $T(M)$. Then the depth of the subterm $M|u$ in M is the length of u. We will extend trees of λ-terms to infinite trees. An infinite λ-term corresponding to an infinite tree T can be described by

$(M_0, M_1, \ldots)$ in Λ^∞ such that for any depth k, if $d(M_i, M_j) < \frac{1}{2^k}$ for all $i, j \geq n$, then $T(M_n)$ and T are the same at depth $\leq k$.

Definition 8.2. For $M \in \Lambda$ and $N \in \Lambda^\infty$, $M \to^\infty N$ is defined to hold if $N = (M_0, M_1, \ldots)$, $M = M_0 \to_\beta M_1 \to_\beta \ldots$, and $\lim_{i\to\infty} d_i = \infty$ where d_i is the depth of the redex for $M_i \to_\beta M_{i+1}$.

Proposition 8.3. *If we have $M_0 \to_\beta M_1 \to_\beta \ldots$ and $\lim_{i\to\infty} d_i = \infty$ where d_i is the depth of the redex for $M_i \to_\beta M_{i+1}$, then $(M_0, M_1, \ldots)$ is in Λ^∞.*

Proof. For a given ϵ, take k such that $\frac{1}{2^k} < \epsilon$. From $\lim d_i = \infty$, there is n_0 such that $d_i > k$ for all $i \geq n_0$. Then for all $i, j \geq n_0$, we have $d(M_i, M_j) < \frac{1}{2^k} < \epsilon$. □

Our Λ^∞ and $\to^\infty$ are equivalent to Λ^{111} and the strongly convergent reduction sequence of length ω in [6]. If we add usual finite β-reduction to our infinite reduction, they will become equivalent to the infinite terms and the infinite β-reduction $\to^\infty$ in [2].

Definition 8.4. We say that $M \in \Lambda^\infty$ has a redex when M is $(M_0, M_1, \ldots)$ and we have u and n such that $M_i|u = (\lambda x.P_i)Q_i$ for all $i \geq n$. $M \in \Lambda^\infty$ is a normal form if M does not have any redex. We say that $M \in \Lambda$ is normalizing by infinite reduction when there is a normal form $N \in \Lambda^\infty$ such that $M \to^\infty N$. NF_∞ is defined to be the set of normalizing λ-terms by infinite reduction.

Theorem 8.5. *The set NF_∞ of λ-terms normalizing by infinite reduction is not recursively enumerable.*

We will prove this theorem in this section after some preparation.

The next lemma is a standard result in λ-calculus.

Lemma 8.6. *If $M \to_\beta^* N$ includes n steps of head reduction, then there is L such that $M \to_h^* L \to_i^* N$ and $M \to_h^* L$ has $\geq n$ steps.*

Definition 8.7. We will use $\mathrm{S}, Y_0, Y, \overline{n}, u, U$ and Δ defined in Section 3. Fix variables a and w. T is defined by

$$T = Y(\lambda txy.Uxy(\lambda w.tx(\mathrm{S}y)a)(\Delta\Delta)).$$

We will write $M\vec{a}^{(n)}$ for $Ma\ldots a \quad$ (n times of a).

$T\overline{en}$ computes $\{e\}^{pr}(n), \{e\}^{pr}(n+1), \ldots$ and produces extra arguments a's if they are positive, and stops with $\Delta\Delta$ when it encounters some m such that $\{e\}^{pr}(m) = 0$.

Proposition 8.8. *If e is not in* PPR*, then $T\overline{e}\overline{0}$ is not normalizing by infinite reduction.*

Proof. Assume $T\overline{e}\overline{0}$ is normalizing by infinite reduction. We will show contradiction.

Suppose $T\overline{e}\overline{0} = M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \ldots$, and M_∞ is $(M_0, M_1, \ldots)$ and normal. Let d_i be the depth of the redex for $M_i \to_\beta M_{i+1}$.

Since e is not in PPR, there is m_0 such that $\{e\}^{pr}(m_0) = 0$ and $\{e\}^{pr}(m) > 0$ for all $m < m_0$.

Choose a fresh variable z and let T' be $Y(\lambda txy.Uxy(\lambda w.tx(\mathrm{S}y)a)z)$. $T'\overline{en} \to_\beta^* T'\overline{en+1}a$ holds when $\{e\}^{pr}(n) > 0$, and $T'\overline{en} \to_\beta^* z$ holds when $\{e\}^{pr}(n) = 0$. Hence $T'\overline{e}\overline{0} \to_\beta^* z\vec{a}^{(m_0)}$. By Lemma 8.6, we have L such that $T'\overline{e}\overline{0} \to_h^* L \to_i^* z\vec{a}^{(m_0)}$. Then $L = z\vec{P}$ for some $\vec{P}$. By substituting $\Delta\Delta$ for z, we have $T\overline{e}\overline{0} \to_h^* \Delta\Delta\vec{P}'$ where $\vec{P}' = \vec{P}[z := \Delta\Delta]$.

Case 1 when $M_0 \to_\beta M_1 \to_\beta \ldots$ includes only finitely many head reduction steps. We have some n such that M_n is a head normal form since M_∞ is normal. Then we have $\Delta\Delta\vec{P}' =_\beta M_n$, which leads to contradiction.

Case 2 when $M_0 \to_\beta M_1 \to_\beta \ldots$ includes infinitely many head reduction steps. Suppose $T\overline{e}\overline{0} \to_h^{n_0} \Delta\Delta\vec{P}'$ for some n_0. We have some n_1 such that $M_0 \to_\beta^* M_{n_1}$ includes at least n_0 head reduction steps. By Lemma 8.6, we have Q such that $M_0 \to_h^{n_0\le} Q \to_i^* M_{n_1}$. Hence $Q = \Delta\Delta\vec{P}'$. Therefore $M_{n_1} = \Delta\Delta\vec{R}$ for some $\vec{R}$, which contradicts to the normality of M_∞. □

Proposition 8.9. *e is in* PPR *if and only if $T\overline{e}\overline{0} \in \mathrm{NF}_\infty$.*

Proof. From the left-hand side to the right-hand side. We have $T\overline{e}\overline{0} \to_\beta^* T\overline{e}\overline{1}a \to_\beta^* T\overline{e}\overline{2}aa \to_\beta^* T\overline{e}\overline{3}aaa \to_\beta^* \ldots$. Let this reduction sequence be $M_0 \to_\beta M_1 \to_\beta \ldots$ and M_∞ be the sequence $(M_0, M_1, \ldots)$. Then $T\overline{e}\overline{0} \to^\infty M_\infty$ holds and M_∞ is a normal form. Hence $T\overline{e}\overline{0}$ is in NF_∞.

From the right-hand side to the left-hand side. The claim immediately follows from Proposition 8.8. □

Proof of Theorem 8.5. If NF_∞ were recursively enumerable, the set $\{e|T\overline{e}\overline{0} \in \mathrm{NF}_\infty\}$ would be also recursively enumerable, but this set is the same as PPR from Proposition 8.9, so PPR would be also recursively enumerable, which would contradict to Proposition 3.5. Therefore NF_∞ is not recursively enumerable. □

The same technique will prove that $\mathrm{NF}_\infty \cup \mathrm{NF}_{<\infty}$ is not recursively enumerable, where $\mathrm{NF}_{<\infty}$ is the set of weakly normalizing terms by β-reduction.

9 Concluding Remarks

Future work will be extending our technique developed in this paper to general infinite λ-calculi Λ^{abc} with $\to^\infty$ for $a, b, c = 0, 1$ in [6], in order to (1) prove the set NF_∞^{abc} of λ-terms having normal forms by $\to^\infty$ in Λ^{abc} is not recursively enumerable and (2) provide an intersection type system with a countably infinite set of types that characterizes the set NF_∞^{abc}.

Acknowledgments

We would like to thank Professor Jan Willem Klop, Professor Mariangiola Dezani-Ciancaglini, and Professor Henk Barendregt for references. We would also like to thank Professor Elio Giovannetti for suggesting Klop's Problem. We would also like to thank Professor Kazushige Terui and Professor Makoto Kanazawa for discussions at NII Logic Seminar. We would also thank the anonymous referees for valuable comments for improving this paper.

References

1. Barendregt, H., Coppo, M., Dezani-Ciancaglini, M.: A Filter Lambda Model and the Completeness of Type Assignment. Journal of Symbolic Logic 48(4), 931–940 (1983)
2. Berarducci, A., Dezani-Ciancaglini, M.: Infinite lambda-calculus and Types. Theoretical Computer Science 212, 29–75 (1999)
3. Dezani-Ciancaglini, M., Honsell, F., Motohama, Y.: Compositional Characterization of λ-terms using Intersection Types. Theoretical Computer Science 340(3), 459–495 (2005)
4. Dezani-Ciancaglini, M.: Private communication (2007)
5. Hindley, J.R., Seldin, J.P.: Introduction to Combinators and λ-Calculus. Cambridge University Press, Cambridge (1986)
6. Kennaway, J.R., Klop, J.W., Sleep, M.R., de Vries, F.J.: Infinitary lambda calculus. Theoretical Computer Science 175(1), 93–125 (1997)
7. Kurata, T.: A Type Theoretical View of Böhm-Trees. In: de Groote, P., Hindley, J.R. (eds.) TLCA 1997. LNCS, vol. 1210, pp. 231–247. Springer, Heidelberg (1997)
8. Raffalli, C.: Data Types, Infinity and Equality in System AF2. In: Meinke, K., Börger, E., Gurevich, Y. (eds.) CSL 1993. LNCS, vol. 832, pp. 280–294. Springer, Heidelberg (1994)
9. Tatsuta, M., Dezani-Ciancaglini, M.: Normalisation is Insensible to lambda-term Identity or Difference. In: Proceedings of Twenty First Annual IEEE Symposium on Logic in Computer Science, pp. 327–336 (2006)
10. Terese (Bezem, M., Klop, J.W., de Vrijer, R. (eds.)), Term Rewriting Systems, Cambridge University Press (2003)