数理解析研究所講究録
第 1635 巻 2009 年 33-50

On General Methods for Proving Reduction Properties of Typed Lambda Terms

Kentaro Kikuchi

RIEC, Tohoku University
Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan
kentaro@nue.riec.tohoku.ac.jp

Abstract. A general method for proving properties of typed lambda terms is developed by adapting proof of strong normalization without the reducibility method. It is applicable to not only normalization but also other reduction properties such as confluence and standardization. A comparison is made between known general conditions for applying the reducibility method and our approach. The method is extended to systems for intersection types with and without the type constant ω.

1 Introduction

Strong normalization for the simply typed lambda calculus is usually proved by the reducibility method [21], which is not formalizable in first order arithmetic. Some methods of proving strong normalization without reducibility have been studied in the literature (see, e.g. [19, Section 5] for a review of those methods for proving strong normalization). Some of them use an inductive characterization of strongly normalizing terms given by van Raamsdonk and Severi [18]. It was also pointed out in [18] that there is a similarity between the inductive characterization and the notion of saturated sets which are used in proofs by reducibility.

On the other hand, general methods for proving properties of typed lambda terms have been developed in the field of reducibility or logical relations. Following Statman's work [20], Mitchell [15, 16] derived sufficient conditions for applying the reducibility method to show various properties apart from strong normalization. In [7], Ghilezan et al. gave more general conditions; in particular, they distinguished two different kinds of conditions which the property to be shown should satisfy. Applications of their method include confluence and strong normalization for both β and $\beta\eta$-reduction, standardization for β-reduction, and some other reduction properties of typed lambda terms.

In this paper, we develop a general method for proving properties of typed lambda terms without using the reducibility method. Instead of the inductive characterization of strongly normalizing terms mentioned above, we introduce a new type assignment system, which is a modification of Valentini's system [23]. Using the new type system, we can prove various properties of typed terms by simple inductions on the typing derivation. The method is purely syntactic and powerful enough to show all the reduction properties treated in [15, 16, 7].

In the latter part of the paper, we extend the method to systems for intersection types. In a similar way to the simply typed case, we prove strong normalization and other reduction properties of terms typable in the system for intersection types without the type constant ω. The method is also applicable to the system with ω, yielding uniform proofs of weak normalization, head normalization and some other reduction properties for terms typable with certain kinds of types.

The main contribution of this paper is the application of proof method for strong normalization without reducibility to other reduction properties of typed lambda terms. This kind of technique was not developed in [18, 23]. We compare this approach with general conditions for applying the reducibility method, and find that our system corresponds to one of the two kinds of conditions explored in [7], while the system in [23] corresponds to the conditions in [15, 16]. As a consequence, our system turns out to provide more general conditions than the system in [23]. It should also be noted that our proof of strong normalization for terms typable with intersection types is simpler than those in [18, 23], using a similar technique to the second proof in [10] for simply typed terms. Moreover our method works well for the system with the type constant ω, which was not studied in [18, 23]. Since type information is essential to proving weak normalization and head normalization for terms typable with certain kinds of types, it is not possible to establish these properties using only inductive characterizations of the sets of weakly and head normalizing terms.

The organization of the paper is as follows. In Section 2 we introduce simply typed lambda calculus. In Section 3 we give a proof of strong normalization for typed terms. In Section 4 we consider application to other properties of typed terms. In Section 5 we compare related work and our method. In Section 6 we extend the method to systems for intersection types without the type constant ω. In Section 7 we apply the method to systems with ω. In Section 8 we discuss related work on intersection types.

2 Simply Typed Lambda Calculus

In this section we introduce two type assignment systems for the simply typed lambda calculus à la Curry. One is in the ordinary natural deduction style and the other in sequent calculus style. The latter is used for proving properties of terms typable in the former.

First we introduce some basic notions on the lambda calculus [1]. The set of terms of the lambda calculus is defined by the grammar: $M ::= x \mid MM \mid \lambda x.M$ where x ranges over a denumerable set of variables. We use letters $x, y, z, \ldots$ for variables and $M, N, P, Q, \ldots$ for terms. The notions of free and bound variables are defined as usual, and the set of free variables occurring in M is denoted by $FV(M)$. We identify α-convertible terms.

The β-rule is stated as $(\lambda x.M)N \to_\beta M[x := N]$ where the expression $M[x := N]$ denotes the term resulting from substituting N for every free occurrence of

Table 1. The system $\lambda_{\rightarrow}$

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ (Ax) \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}\ (\rightarrow I)$$

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}\ (\rightarrow E)$$

x in M. β-reduction is the contextual closure of the β-rule. We use $\rightarrow_\beta$ for one-step reduction, and $\stackrel{*}{\rightarrow}_\beta$ for its reflexive transitive closure. β-equality $=_\beta$ is the symmetric closure of $\stackrel{*}{\rightarrow}_\beta$. A term M is said to be *strongly normalizing* if all β-reduction sequences starting from M terminate. The set of strongly normalizing terms is denoted by $\mathcal{SN}_\beta$.

The set of simple types is defined by the grammar: $\sigma ::= \varphi \mid \sigma \rightarrow \sigma$ where φ ranges over a denumerable set of type atoms. We use letters $\sigma, \tau, \rho, \ldots$ for arbitrary types. The type assignment system $\lambda_{\rightarrow}$ is defined by the rules in Table 1. A *basis* in the system $\lambda_{\rightarrow}$ is defined as a finite set of pairs $\{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ where the variables are pairwise distinct. The basis $\Gamma, x : \sigma$ denotes the union $\Gamma \cup \{x : \sigma\}$ where $x \notin \Gamma$ ($x \notin \Gamma$ means that x does not appear in Γ, i.e., for no type τ, $x : \tau \in \Gamma$).

Besides the usual natural deduction style system $\lambda_{\rightarrow}$, we introduce a sequent calculus style system $\lambda^s_{\rightarrow}$ defined by the rules in Table 2. In the system $\lambda^s_{\rightarrow}$, $x : \sigma$ is allowed to appear in Γ of the basis $\Gamma, x : \sigma$. Note in particular that the premises of the rule $(L \rightarrow)$ may have $x : \sigma \rightarrow \tau$ in Γ. The $(Beta)^s$ rule is directly inspired by the reduction relation. The $(Beta)^s$-free part of the system $\lambda^s_{\rightarrow}$ derives judgements for normal lambda terms, whose derivations correspond to the so-called normal cut-free proofs (cf. [22, p. 193]).

Table 2. The system $\lambda^s_{\rightarrow}$

$$\frac{}{\Gamma, x : \sigma \vdash_s x : \sigma}\ (Ax) \qquad \frac{\Gamma \vdash_s M[x := N]N_1 \ldots N_n : \sigma \quad \Gamma \vdash_s N : \tau}{\Gamma \vdash_s (\lambda x.M)NN_1 \ldots N_n : \sigma}\ (Beta)^s$$

$$\frac{\Gamma \vdash_s N : \sigma \quad \Gamma, y : \tau \vdash_s yN_1 \ldots N_n : \rho}{\Gamma, x : \sigma \rightarrow \tau \vdash_s xNN_1 \ldots N_n : \rho}\ (L \rightarrow) \qquad \frac{\Gamma, x : \sigma \vdash_s M : \tau}{\Gamma \vdash_s \lambda x.M : \sigma \rightarrow \tau}\ (R \rightarrow)$$

where $y \notin FV(N_1) \cup \cdots \cup FV(N_n)$ and $y \notin \Gamma$ where $x \notin \Gamma$

3 A Proof of Strong Normalization

If we try to prove strong normalization for the terms typed in the system $\lambda_{\rightarrow}$ directly by induction on derivations, we will find difficulty in the case $(\rightarrow E)$. One way of overcoming this difficulty is to use reducibility predicates [21]. Here we use the sequent calculus style system $\lambda^s_{\rightarrow}$ instead. For the system $\lambda^s_{\rightarrow}$, we can prove strong normalization for typed terms directly by induction on derivations.

Theorem 1. *If $\Gamma \vdash_s M : \sigma$ then $M \in \mathcal{SN}_\beta$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$ in $\lambda^s_{\rightarrow}$. The only problematic case is where the last rule applied is $(Beta)^s$. In that case, by the induction hypothesis, we have $M[x := N]N_1 \ldots N_n \in \mathcal{SN}_\beta$ and $N \in \mathcal{SN}_\beta$. From the former we have $M, N_1, \ldots, N_n \in \mathcal{SN}_\beta$. Then any infinite reduction sequence starting from $(\lambda x.M)NN_1 \ldots N_n$ must have the form

$$\begin{aligned}(\lambda x.M)NN_1 \ldots N_n &\overset{*}{\rightarrow}_\beta (\lambda x.M')N'N_1' \ldots N_n' \\ &\rightarrow_\beta M'[x := N']N_1' \ldots N_n' \\ &\rightarrow_\beta \ \cdots\end{aligned}$$

where $M \overset{*}{\rightarrow}_\beta M'$, $N \overset{*}{\rightarrow}_\beta N'$ and $N_i \overset{*}{\rightarrow}_\beta N_i'$ for $1 \leq i \leq n$. But then there is an infinite reduction sequence

$$\begin{aligned}M[x := N]N_1 \ldots N_n &\overset{*}{\rightarrow}_\beta M'[x := N']N_1' \ldots N_n' \\ &\rightarrow_\beta \ \cdots\end{aligned}$$

contradicting the hypothesis. Hence $(\lambda x.M)NN_1 \ldots N_n \in \mathcal{SN}_\beta$. □

To complete a proof of strong normalization for terms typed in $\lambda_{\rightarrow}$, what remains to be shown is that if M is typable in $\lambda_{\rightarrow}$ then it is typable in $\lambda^s_{\rightarrow}$. In the following we show that $\lambda^s_{\rightarrow}$ is indeed closed under the rules of $\lambda_{\rightarrow}$. For this we first show that $\lambda^s_{\rightarrow}$ is closed under the weakening rule.

Lemma 1. *If $\Gamma \vdash_s M : \tau$ and $x \notin \Gamma$ then $\Gamma, x : \sigma \vdash_s M : \tau$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$. □

Next we prove a useful lemma, which is needed in the proofs of subsequent lemmas. This technique resembles the second proof of strong normalization for the simply typed lambda calculus in [10].

Lemma 2. *If $\Gamma \vdash_s M : \sigma \rightarrow \tau$ and $x \notin \Gamma$ then $\Gamma, x : \sigma \vdash_s Mx : \tau$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \sigma \rightarrow \tau$. Let us consider here some cases.

- $$\frac{}{\Gamma, y : \sigma \to \tau \vdash_s y : \sigma \to \tau}\ (Ax)$$
  In this case we take two axioms $\Gamma, x : \sigma \vdash_s x : \sigma$ and $\Gamma, x : \sigma, z : \tau \vdash_s z : \tau$, and obtain $\Gamma, x : \sigma, y : \sigma \to \tau \vdash_s yx : \tau$ by an instance of the $(L \to)$ rule.
- $$\frac{\Gamma \vdash_s M[y := N]N_1 \dots N_n : \sigma \to \tau \quad \Gamma \vdash_s N : \rho}{\Gamma \vdash_s (\lambda y.M)NN_1 \dots N_n : \sigma \to \tau}\ (Beta)^s$$
  By the induction hypothesis, we have $\Gamma, x : \sigma \vdash_s M[y := N]N_1 \dots N_n x : \tau$, and by Lemma 1, we have $\Gamma, x : \sigma \vdash_s N : \rho$. From these, we obtain $\Gamma, x : \sigma \vdash_s (\lambda y.M)NN_1 \dots N_n x : \tau$ by an instance of the $(Beta)^s$ rule.
- $$\frac{\Gamma, y : \sigma \vdash_s M : \tau}{\Gamma \vdash_s \lambda y.M : \sigma \to \tau}\ (R \to)$$
  where $y \notin \Gamma$. From the judgement $\Gamma, y : \sigma \vdash_s M : \tau$, we have $\Gamma, x : \sigma \vdash_s M[y := x] : \tau$. From this and $\Gamma, x : \sigma \vdash_s x : \sigma$, which is an axiom, we obtain $\Gamma, x : \sigma \vdash_s (\lambda y.M)x : \tau$ by an instance of the $(Beta)^s$ rule. □

Now we are in a position to show that $\lambda^s_{\to}$ is closed under substitution. This is proved by a technique similar to Gentzen's cut-elimination procedure, where Lemma 2 plays a role of the inversion lemma.

Lemma 3. *If $\Gamma \vdash_s N : \sigma$ and $\Gamma, x : \sigma \vdash_s P : \tau$, where $x \notin \Gamma$, then $\Gamma \vdash_s P[x := N] : \tau$.*

Proof. The proof is by main induction on the size of σ and subinduction on the length of the derivation of $\Gamma, x : \sigma \vdash_s P : \tau$. Let us consider here some cases according to the last rule used in the derivation of $\Gamma, x : \sigma \vdash_s P : \tau$.

- $$\frac{}{\Gamma, x : \sigma, y : \tau \vdash_s y : \tau}\ (Ax) \quad (y \neq x)$$
  In this case we just have to take $\Gamma, y : \tau \vdash_s y : \tau$, which is an axiom.
- $$\frac{}{\Gamma, x : \sigma \vdash_s x : \sigma}\ (Ax)$$
  In this case we have to prove $\Gamma \vdash_s N : \sigma$. But it is one of the assumptions.
- $$\frac{\Gamma, x : \sigma \vdash_s M[y := Q]N_1 \dots N_n : \tau \quad \Gamma, x : \sigma \vdash_s Q : \rho}{\Gamma, x : \sigma \vdash_s (\lambda y.M)QN_1 \dots N_n : \tau}\ (Beta)^s$$
  By the subinduction hypothesis, we obtain both
  $$\Gamma \vdash_s M[y := Q][x := N]N_1[x := N] \dots N_n[x := N] : \tau$$
  and
  $$\Gamma \vdash_s Q[x := N] : \rho.$$
  Since y is an abstracted variable, we can assume that it does not appear in N. Hence the first judgement is
  $$\Gamma \vdash_s M[x := N][y := Q[x := N]]N_1[x := N] \dots N_n[x := N] : \tau.$$

Thus we obtain

$$\Gamma \vdash_s (\lambda y.M[x := N])Q[x := N]N_1[x := N] \dots N_n[x := N] : \tau$$

by an instance of the $(Beta)^s$ rule.

- $$\frac{\Gamma, x:\sigma \vdash_s M : \rho_1 \quad \Gamma, x:\sigma, y:\rho_2 \vdash_s yN_1 \dots N_n : \tau}{\Gamma, x:\sigma, z:\rho_1 \to \rho_2 \vdash_s zMN_1 \dots N_n : \tau}\ (L \to) \quad (z \neq x)$$
where $y \notin FV(N_1) \cup \dots \cup FV(N_n)$ and $y \notin \Gamma, x : \sigma$. By the subinduction hypothesis, we obtain both

$$\Gamma, z : \rho_1 \to \rho_2 \vdash_s M[x := N] : \rho_1$$

and

$$\Gamma, z : \rho_1 \to \rho_2, y : \rho_2 \vdash_s (yN_1 \dots N_n)[x := N] : \tau.$$

Since $y \neq x$, we have $(yN_1 \dots N_n)[x := N] \equiv yN_1[x := N] \dots N_n[x := N]$. Hence we conclude by an instance of the $(L \to)$ rule.

- $$\frac{\Gamma' \vdash_s M : \sigma_1 \quad \Gamma', y : \sigma_2 \vdash_s yN_1 \dots N_n : \tau}{\Gamma', x : \sigma_1 \to \sigma_2 \vdash_s xMN_1 \dots N_n : \tau}\ (L \to)$$
where $y \notin FV(N_1) \cup \dots \cup FV(N_n)$, $y \notin \Gamma'$ and $\Gamma' \setminus \{x : \sigma_1 \to \sigma_2\} = \Gamma$. By the subinduction hypothesis, we obtain both

$$\Gamma \vdash_s M[x := N] : \sigma_1 \tag{1}$$

and

$$\Gamma, y : \sigma_2 \vdash_s (yN_1 \dots N_n)[x := N] : \tau \tag{2}$$

and, since we can assume $y \neq x$, we have $(yN_1 \dots N_n)[x := N] \equiv yN_1[x := N] \dots N_n[x := N]$. Now consider the assumption $\Gamma \vdash_s N : \sigma_1 \to \sigma_2$ and a fresh variable z which does not appear in Γ. Then by Lemma 2, we have $\Gamma, z : \sigma_1 \vdash_s Nz : \sigma_2$. Hence, by the main induction hypothesis, we obtain $\Gamma \vdash_s NM[x := N] : \sigma_2$ by substituting the term $M[x := N]$ in (1) for z. Then, again by the main induction hypothesis, we obtain

$$\Gamma \vdash_s NM[x := N]N_1[x := N] \dots N_n[x := N] : \tau$$

by substituting the term $NM[x := N]$ for y in (2). □

Now we can prove that the system $\lambda^s_\to$ is closed under the $(\to E)$ rule.

Lemma 4. *If $\Gamma \vdash_s M : \sigma \to \tau$ and $\Gamma \vdash_s N : \sigma$ then $\Gamma \vdash_s MN : \tau$.*

Proof. By Lemma 2, we have $\Gamma, x : \sigma \vdash_s Mx : \tau$ for any fresh variable x. Hence by the previous lemma, we obtain $\Gamma \vdash_s (Mx)[x := N] \equiv MN : \tau$. □

Now we prove the announced theorem.

Theorem 2. *If $\Gamma \vdash M : \sigma$ then $\Gamma \vdash_s M : \sigma$.*

Proof. By induction on the derivation of $\Gamma \vdash M : \sigma$ in $\lambda_{\rightarrow}$, using Lemma 4. □

Corollary 1. *If $\Gamma \vdash M : \sigma$ then $M \in \mathcal{SN}_\beta$.*

Proof. By Theorems 1 and 2. □

4 Application to Other Properties

Since the proof of Theorem 2 in the previous section is independent of strong normalization, we may prove other properties of typed terms by induction on derivations in the system $\lambda^s_{\rightarrow}$. In the present section we illustrate this by showing confluence and standardization for β-reduction and strong normalization for $\beta\eta$-reduction on typed terms.

First we define the set $\mathcal{CR}_\beta$ as the set of terms M that satisfy the following:

$$\forall M_1, M_2[M \stackrel{*}{\rightarrow}_\beta M_1 \wedge M \stackrel{*}{\rightarrow}_\beta M_2 \Rightarrow \exists N[M_1 \stackrel{*}{\rightarrow}_\beta N \wedge M_2 \stackrel{*}{\rightarrow}_\beta N]].$$

Theorem 3. *If $\Gamma \vdash_s M : \sigma$ then $M \in \mathcal{CR}_\beta$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$ in $\lambda^s_{\rightarrow}$. The only problematic case is where the last rule applied is $(Beta)^s$. In that case, by the induction hypothesis, we have $M[x := N]N_1 \dots N_n \in \mathcal{CR}_\beta$. Our aim is to prove $(\lambda x.M)NN_1 \dots N_n \in \mathcal{CR}_\beta$. For this, it suffices to show that if P is obtained from $(\lambda x.M)NN_1 \dots N_n$ by some steps of β-reduction then we can reduce P to some term that is obtained from $M[x := N]N_1 \dots N_n$.

There are two cases to consider. First, suppose

$$(\lambda x.M)NN_1 \dots N_n \stackrel{*}{\rightarrow}_\beta (\lambda x.M')N'N'_1 \dots N'_n \equiv P$$

where $M \stackrel{*}{\rightarrow}_\beta M'$, $N \stackrel{*}{\rightarrow}_\beta N'$ and $N_i \stackrel{*}{\rightarrow}_\beta N'_i$ for $1 \leq i \leq n$. Then we can reduce P to $M'[x := N']N'_1 \dots N'_n$, which is obtained also from $M[x := N]N_1 \dots N_n$. Next, suppose

$$\begin{aligned}(\lambda x.M)NN_1 \dots N_n &\stackrel{*}{\rightarrow}_\beta (\lambda x.M')N'N'_1 \dots N'_n \\ &\rightarrow_\beta M'[x := N']N'_1 \dots N'_n \\ &\stackrel{*}{\rightarrow}_\beta P\end{aligned}$$

where $M \stackrel{*}{\rightarrow}_\beta M'$, $N \stackrel{*}{\rightarrow}_\beta N'$ and $N_i \stackrel{*}{\rightarrow}_\beta N'_i$ for $1 \leq i \leq n$. Then we have

$$\begin{aligned}M[x := N]N_1 \dots N_n &\stackrel{*}{\rightarrow}_\beta M'[x := N']N'_1 \dots N'_n \\ &\stackrel{*}{\rightarrow}_\beta P.\end{aligned}$$

This completes the proof. □

Corollary 2. *If $\Gamma \vdash M : \sigma$ then $M \in \mathcal{CR}_\beta$.*

Proof. By Theorems 2 and 3. □

Next we consider standardization for β-reduction. For this we need to introduce some notions. If $M \equiv \lambda x_1 \ldots \lambda x_m.(\lambda x.P)QN_1 \ldots N_n$ $(m, n \geq 0)$ then $(\lambda x.P)Q$ is called the *head redex* of M. We write $M \to_h M'$ if M' is obtained from M by reducing the head redex of M (head reduction). We write $M \to_i M'$ if M' is obtained from M by reducing a redex that is not a head redex (internal reduction). The set $\mathcal{ST}_\beta$ is then defined as the set of terms M that satisfy the following:

$$\forall P[M \xrightarrow{*}_\beta P \Rightarrow \exists N[M \xrightarrow{*}_h N \wedge N \xrightarrow{*}_i P]].$$

Theorem 4. *If $\Gamma \vdash_s M : \sigma$ then $M \in \mathcal{ST}_\beta$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$ in $\lambda^s_\to$. The only problematic case is where the last rule applied is $(Beta)^s$. In that case, by the induction hypothesis, we have $M[x := N]N_1 \ldots N_n \in \mathcal{ST}_\beta$. Our aim is to prove $(\lambda x.M)NN_1 \ldots N_n \in \mathcal{ST}_\beta$. Suppose that P is obtained from $(\lambda x.M)NN_1 \ldots N_n$ by some steps of β-reduction. There are two cases to consider. First, if

$$(\lambda x.M)NN_1 \ldots N_n \xrightarrow{*}_\beta (\lambda x.M')N'N'_1 \ldots N'_n \equiv P$$

where $M \xrightarrow{*}_\beta M'$, $N \xrightarrow{*}_\beta N'$ and $N_i \xrightarrow{*}_\beta N'_i$ for $1 \leq i \leq n$, then the reduction is internal, so we are done. Secondly, if

$$\begin{aligned}(\lambda x.M)NN_1 \ldots N_n &\xrightarrow{*}_\beta (\lambda x.M')N'N'_1 \ldots N'_n \\ &\to_\beta M'[x := N']N'_1 \ldots N'_n \\ &\xrightarrow{*}_\beta P\end{aligned}$$

where $M \xrightarrow{*}_\beta M'$, $N \xrightarrow{*}_\beta N'$ and $N_i \xrightarrow{*}_\beta N'_i$ for $1 \leq i \leq n$, then we have

$$\begin{aligned}M[x := N]N_1 \ldots N_n &\xrightarrow{*}_\beta M'[x := N']N'_1 \ldots N'_n \\ &\xrightarrow{*}_\beta P.\end{aligned}$$

Since $(\lambda x.M)NN_1 \ldots N_n \to_h M[x := N]N_1 \ldots N_n \in \mathcal{ST}_\beta$, we can conclude that $(\lambda x.M)NN_1 \ldots N_n \in \mathcal{ST}_\beta$. □

Corollary 3. *If $\Gamma \vdash M : \sigma$ then $M \in \mathcal{ST}_\beta$.*

Proof. By Theorems 2 and 4. □

The standardization theorem in the usual sense (cf. [1, p. 300]) follows immediately from the above corollary.

Next we consider strong normalization for $\beta\eta$-reduction. The η-rule is stated as $\lambda x.Mx \to_\eta M$ where $x \notin FV(M)$. η-reduction is the contextual closure of the η-rule. The set of terms that are strongly normalizing with respect to $\beta\eta$-reduction is denoted by $\mathcal{SN}_{\beta\eta}$.

Theorem 5. *If* $\Gamma \vdash_s M : \sigma$ *then* $M \in \mathcal{SN}_{\beta\eta}$.

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$ in $\lambda^s_{\rightarrow}$. The case where the last rule applied is $(Beta)^s$ is solved in a similar way to the proof of Theorem 1, except that an infinite reduction sequence starting from $(\lambda x.M)NN_1 \ldots N_n$ might have the form

$$\begin{aligned}(\lambda x.M)NN_1 \ldots N_n &\xrightarrow{*}_{\beta\eta} (\lambda x.M'x)N'N'_1 \ldots N'_n \\ &\rightarrow_\eta M'N'N'_1 \ldots N'_n \\ &\rightarrow_{\beta\eta} \cdots\end{aligned}$$

where $M \xrightarrow{*}_{\beta\eta} M'x$, $x \notin FV(M')$, $N \xrightarrow{*}_{\beta\eta} N'$ and $N_i \xrightarrow{*}_{\beta\eta} N'_i$ for $1 \leq i \leq n$. But since the reduction step $(\lambda x.M'x)N' \rightarrow_\eta M'N'$ can also be carried out by β-reduction, this case reduces to the case we treated in the proof of Theorem 1. Next, if the last rule applied is $(R \rightarrow)$, we have to show that $M \in \mathcal{SN}_{\beta\eta}$ implies $\lambda x.M \in \mathcal{SN}_{\beta\eta}$. For this, suppose $M \in \mathcal{SN}_{\beta\eta}$. Then we have $M[x := x] \in \mathcal{SN}_{\beta\eta}$ and $x \in \mathcal{SN}_{\beta\eta}$, and hence we have $(\lambda x.M)x \in \mathcal{SN}_{\beta\eta}$ by a similar reasoning to the case of $(Beta)^s$ where $N \equiv x$ and $n = 0$. Thus we have $\lambda x.M \in \mathcal{SN}_{\beta\eta}$. □

Corollary 4. *If* $\Gamma \vdash M : \sigma$ *then* $M \in \mathcal{SN}_{\beta\eta}$.

Proof. By Theorems 2 and 5. □

5 Comparison with Related Work

In this section we make a comparison among several methods, with and without reducibility, for proving properties of typed lambda terms. We distinguish methods with two different kinds of conditions, and discuss which one provides more general conditions.

Our sequent calculus style system $\lambda^s_{\rightarrow}$ is a modification of Valentini's system [23] which was introduced to prove strong normalization without using the reducibility method. The difference between Valentini's system and ours is as follows. First, the rule $(R \rightarrow)$ of Valentini's system has the form

$$\frac{\Gamma, x : \sigma \vdash_s Mx : \tau}{\Gamma \vdash_s M : \sigma \rightarrow \tau}$$

where $x \notin \Gamma$ and $x \notin FV(M)$, while in our system the rule $(R \rightarrow)$ is the same as the usual abstraction rule. Also, Valentini's system has restriction on types in some rules to type atoms. With this restriction and the lack of our Lemma 2, his proof of strong normalization is more complicated than ours.

In [23], Valentini treated normalization properties only, and did not point out that his system can be used for proving other properties than normalization. However, the system has a close relation to the conditions that Mitchell [15, 16] derived for applying the reducibility method to show various properties apart from normalization. To be precise, let $\mathcal{P}$ be a property (a set) of lambda terms. $\mathcal{P}$ is said to be *type-closed* if the following three conditions[1] are satisfied.

1. $Mx \in \mathcal{P} \Rightarrow M \in \mathcal{P}$ where $x \notin FV(M)$,
2. $M_1 \in \mathcal{P} \wedge \cdots \wedge M_n \in \mathcal{P} \Rightarrow xM_1 \ldots M_n \in \mathcal{P}$,
3. $M[x := N]N_1 \ldots N_n \in \mathcal{P} \wedge N \in \mathcal{P} \Rightarrow (\lambda x.M)NN_1 \ldots N_n \in \mathcal{P}$.

Mitchell showed that if M is typable in the system $\lambda_{\rightarrow}$ then $M \in \mathcal{P}$ for any type-closed $\mathcal{P}$, using reducibility predicates as a special case of logical relations. Now there is a similarity between the above conditions and the typing rules of Valentini's system; the condition 1 corresponds to the $(R \rightarrow)$ rule, the condition 2 to the $(L \rightarrow)$ rule, and the condition 3 to the $(Beta)^s$ rule. Verifying that $\mathcal{P}$ satisfies the three conditions is similar to proving by induction on derivations that every term typed in Valentini's system has the property $\mathcal{P}$. For example, one can show that every term typed in Valentini's system has the property $\mathcal{CR}_{\beta\eta}$ in the same way as explained in [16, pp. 557–558].

Now our system $\lambda^s_{\rightarrow}$ is different from Valentini's in the $(R \rightarrow)$ rule, so it is expected that there is a corresponding version of type-closed set and the system $\lambda^s_{\rightarrow}$ can also be used for proving properties of typed terms. Such a version of type-closed set is found in Ghilezan et al. [7], where the condition 1 is replaced by the following condition 1′.

1′. $M \in \mathcal{P} \Rightarrow \lambda x.M \in \mathcal{P}$.

Here we say that $\mathcal{P}$ is *type-closed′* if the conditions 1′, 2 and 3 are satisfied. For these conditions, one can also prove that if M is typable in the system $\lambda_{\rightarrow}$ then $M \in \mathcal{P}$ for any type-closed′ $\mathcal{P}$, using an alternative version of the reducibility method [7]. Since the conditions of type-closed′ set correspond to the typing rules in our system $\lambda^s_{\rightarrow}$, verifying that $\mathcal{P}$ is type-closed′ is similar to proving by induction on derivations in $\lambda^s_{\rightarrow}$ that every typed term has the property $\mathcal{P}$. Thus we can prove all properties of typed terms treated in [7] without using the reducibility method, in such a way as we demonstrated in the previous section.

The conditions of type-closed′ set also appear in van Raamsdonk and Severi [18] as the clauses of an inductive characterization of strongly normalizing terms. They showed that the smallest type-closed′ set coincides with the set of strongly normalizing terms and that if M is typable in the system $\lambda_{\rightarrow}$ then M belongs to the smallest type-closed′ set.

Table 3 summarizes the general methods mentioned above (though in [23] and [18] it was not explicitly pointed out that their systems can be used for proving other properties than normalization).

[1] Strictly speaking, the condition 1 of type-closed set in [15, 16] is slightly more general, but in verifying that a particular set $\mathcal{P}$ satisfies the condition 1, we generally assume $Mx \in \mathcal{P}$ for $x \notin FV(M)$ and show $M \in \mathcal{P}$.

Table 3. General methods for typed lambda terms

	1. $Mx \in \mathcal{P} \Rightarrow M \in \mathcal{P}$	1$'$. $M \in \mathcal{P} \Rightarrow \lambda x.M \in \mathcal{P}$
Reducibility	Mitchell [15, 16]	Ghilezan, Kunčak, and Likavec [7]
Without reducibility	Valentini [23]	van Raamsdonk and Severi [18] This work

Now we are interested in which method provides the most general conditions. To see this, it is useful to note the following fact, which was observed in Koletsos and Stavrinos [12].

Proposition 1. *Let $\mathcal{P}$ be a set of lambda terms that satisfies the conditions 1, 2 and 3 of type-closed set. Then $\mathcal{P}$ also satisfies the condition 1$'$, hence $\mathcal{P}$ is a type-closed$'$ set.*

Proof. Let $\mathcal{P}$ satisfy the conditions 1, 2 and 3, and let $M \in \mathcal{P}$. Then $M[x := x] \in \mathcal{P}$ and $x \in \mathcal{P}$ by the condition 2, and so $(\lambda x.M)x \in \mathcal{P}$ by the condition 3. Hence $\lambda x.M \in \mathcal{P}$ by the condition 1. □

This proposition means that if we can verify that $\mathcal{P}$ satisfies 1, 2 and 3, then we can always verify that $\mathcal{P}$ also satisfies 1$'$. In general, the condition 1 is suitable to prove properties of $\beta\eta$-reduction, and once we establish that $\mathcal{P}$ satisfies 1, 2 and 3, we can obtain that $\mathcal{P}$ also satisfies 1$'$ (cf. the proof of Theorem 5).

It seems difficult to show the converse of Proposition 1 directly. One of the examples of properties for which the condition 1 is difficult to establish is confluence for β-reduction, as remarked in [16, p. 559]. In contrast, it is quite easy to show that the condition 1$'$ holds for confluence for β-reduction as we saw in the proof of Theorem 3.

Thus we conclude that the methods using the condition 1$'$ are more general than those using the condition 1. Now, in the remaining three, this work differs from the others in that we use a type system in sequent calculus style. In the next sections we consider systems for intersection types to illustrate that type information is useful.

6 Extension to Intersection Types

In the remainder of the paper we are concerned with systems for intersection types. (For background information about intersection types, see, e.g. [5].) In this section we introduce type systems for intersection types without the type constant ω. We prove strong normalization for typable terms in a similar way to that in Section 3, as well as other reduction properties discussed in the previous sections.

Like in the simply typed case, we introduce two kinds of type assignment systems for intersection types. The ordinary natural deduction style system $\lambda_\cap$ is obtained from $\lambda_\rightarrow$ by adding the following rules for $\cap$:

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \cap \tau}\,(\cap I) \qquad \frac{\Gamma \vdash M : \sigma \cap \tau}{\Gamma \vdash M : \sigma}\,(\cap E) \qquad \frac{\Gamma \vdash M : \sigma \cap \tau}{\Gamma \vdash M : \tau}\,(\cap E)$$

The sequent calculus style system $\lambda^s_\cap$ is obtained from the system $\lambda^s_\rightarrow$ by adding the following rules:

$$\frac{\Gamma, x : \sigma_1, x : \sigma_2 \vdash_s xN_1 \ldots N_n : \tau}{\Gamma, x : \sigma_1 \cap \sigma_2 \vdash_s xN_1 \ldots N_n : \tau}\,(L\cap) \qquad \frac{\Gamma \vdash_s M : \sigma \quad \Gamma \vdash_s M : \tau}{\Gamma \vdash_s M : \sigma \cap \tau}\,(R\cap)$$

where a variable may have different types in a basis. Such a variable is intended to have the type of intersection of all the different types.

Example 1. Self-application can now be typed naturally in $\lambda^s_\cap$, as follows (cf. [23, pp. 478–479]).

$$\cfrac{\cfrac{\cfrac{x : \sigma \vdash_s x : \sigma \quad x : \sigma, y : \tau \vdash_s y : \tau}{x : \sigma, x : \sigma \to \tau \vdash_s xx : \tau}\,(L \to)}{x : \sigma \cap (\sigma \to \tau) \vdash_s xx : \tau}\,(L\cap)}{\vdash_s \lambda x.xx : (\sigma \cap (\sigma \to \tau)) \to \tau}\,(R \to)$$

As in the system $\lambda^s_\rightarrow$, we can prove strong normalization for terms typable in $\lambda^s_\cap$ directly by induction on derivations.

Theorem 6. *If $\Gamma \vdash_s M : \sigma$ then $M \in \mathcal{SN}_\beta$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$ in $\lambda^s_\cap$, similarly to the proof of Theorem 1. □

To prove strong normalization for terms typable in the system $\lambda_\cap$, we show below that if M is typable in the system $\lambda_\cap$ then it is typable in $\lambda^s_\cap$. First we prove the following lemmas on properties of $\lambda^s_\cap$.

Lemma 5. *If $\Gamma \vdash_s M : \tau$ and $x \notin \Gamma$ then $\Gamma, x : \sigma \vdash_s M : \tau$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \tau$. □

Lemma 6. *If $\Gamma \vdash_s M : \sigma \to \tau$ and $x \notin \Gamma$ then $\Gamma, x : \sigma \vdash_s Mx : \tau$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \sigma \to \tau$, similarly to the proof of Lemma 2. □

Lemma 7. *If $\Gamma \vdash_s M : \sigma \cap \tau$ then $\Gamma \vdash_s M : \sigma$ and $\Gamma \vdash_s M : \tau$.*

Proof. By induction on the derivation of $\Gamma \vdash_s M : \sigma \cap \tau$. $\square$

Now we show that $\lambda_\cap^s$ is closed under substitution, which is stated as follows.

Lemma 8. *If $\Gamma, x : \sigma_1, \ldots, x : \sigma_m \vdash_s P : \tau$, where $x \notin \Gamma$, and $\Gamma \vdash_s N : \sigma_i$ for any $1 \leq i \leq m$, then $\Gamma \vdash_s P[x := N] : \tau$.*

Proof. The proof is by main induction on the complexity $\mu(\sigma_1, \ldots, \sigma_m)$ of the sequence $\sigma_1, \ldots, \sigma_m$ and subinduction on the length of the derivation of $\Gamma, x : \sigma_1, \ldots, x : \sigma_m \vdash_s P : \tau$. The inductive definition of the complexity measure μ is the following:

$$\mu(\Phi) = \begin{cases} 1 & \text{if } \Phi \text{ is a type atom} \\ \mu(\sigma) + \mu(\tau) + 1 & \text{if } \Phi = \sigma \to \tau \\ \mu(\sigma) + \mu(\tau) + 1 & \text{if } \Phi = \sigma \cap \tau \\ \mu(\sigma_1 \cap \cdots \cap \sigma_m) & \text{if } \Phi = \sigma_1, \ldots, \sigma_m \end{cases}$$

Most of the cases proceed in a similar way to those in the proof of Lemma 3. Here we only consider a few cases. (We write $\overline{x} : \overline{\sigma}$ as a shorthand for $x : \sigma_1, \ldots, x : \sigma_m$.)

- $$\frac{\Gamma, \overline{x} : \overline{\sigma} \vdash_s M : \rho_1 \quad \Gamma, \overline{x} : \overline{\sigma}, y : \rho_2 \vdash_s yN_1 \ldots N_n : \tau}{\Gamma, \overline{x} : \overline{\sigma}, x : \rho_1 \to \rho_2 \vdash_s xMN_1 \ldots N_n : \tau} \ (L \to)$$

  where $y \notin FV(N_1) \cup \cdots \cup FV(N_n)$ and $y \notin \Gamma, \overline{x} : \overline{\sigma}$. By the subinduction hypothesis, we obtain both

  $$\Gamma \vdash_s M[x := N] : \rho_1 \tag{3}$$

  and

  $$\Gamma, y : \rho_2 \vdash_s (yN_1 \ldots N_n)[x := N] : \tau \tag{4}$$

  and, since $y \neq x$, we get $(yN_1 \ldots N_n)[x := N] \equiv yN_1[x := N] \ldots N_n[x := N]$. Now consider the assumption $\Gamma \vdash_s N : \rho_1 \to \rho_2$ and a fresh variable z which does not appear in Γ. Then by Lemma 6, we have $\Gamma, z : \rho_1 \vdash_s Nz : \rho_2$. Hence, by the main induction hypothesis, we obtain $\Gamma \vdash_s NM[x := N] : \rho_2$ by substituting the term $M[x := N]$ in (3) for z. Then, again by the main induction hypothesis, we obtain

  $$\Gamma \vdash_s NM[x := N]N_1[x := N] \ldots N_n[x := N] : \tau$$

  by substituting the term $NM[x := N]$ for y in (4).

- $$\frac{\Gamma, \overline{x} : \overline{\sigma}, x : \rho_1, x : \rho_2 \vdash_s xN_1 \ldots N_n : \tau}{\Gamma, \overline{x} : \overline{\sigma}, x : \rho_1 \cap \rho_2 \vdash_s xN_1 \ldots N_n : \tau} \ (L\cap)$$

  Let us consider the assumption $\Gamma \vdash_s N : \rho_1 \cap \rho_2$. Then, by Lemma 7, we have $\Gamma \vdash_s N : \rho_1$ and $\Gamma \vdash_s N : \rho_2$. Hence, by the subinduction hypothesis, we obtain $\Gamma \vdash_s NN_1[x := N] \ldots N_n[x := N] : \tau$ by substituting the term N for x in $\Gamma, \overline{x} : \overline{\sigma}, x : \rho_1, x : \rho_2 \vdash_s xN_1 \ldots N_n : \tau$. $\square$

Now we can prove that the system $\lambda^s_\cap$ is closed under the $(\to E)$ rule.

Lemma 9. *If $\Gamma \vdash_s M : \sigma \to \tau$ and $\Gamma \vdash_s N : \sigma$ then $\Gamma \vdash_s MN : \tau$.*

Proof. By Lemma 6, we have $\Gamma, x : \sigma \vdash_s Mx : \tau$ for any fresh variable x. Hence by the previous lemma, we obtain $\Gamma \vdash_s (Mx)[x := N] \equiv MN : \tau$. □

Now we prove the announced theorem.

Theorem 7. *If $\Gamma \vdash M : \sigma$ then $\Gamma \vdash_s M : \sigma$.*

Proof. By induction on the derivation of $\Gamma \vdash M : \sigma$ in $\lambda_\cap$, using Lemmas 7 and 9. □

Corollary 5. *If $\Gamma \vdash M : \sigma$ then $M \in \mathcal{SN}_\beta$.*

Proof. By Theorems 6 and 7. □

The properties discussed in the previous sections can also be obtained. For instance, Corollaries 2, 3 and 4 in Section 4 hold for $\lambda_\cap$ instead of $\lambda_\to$ because Theorems 3, 4 and 5 can be proved by induction on derivations in $\lambda^s_\cap$ similarly to the case of $\lambda^s_\to$.

7 Type Systems with ω

The method presented in the previous sections works also for intersection type systems with the type constant ω. In this section we introduce some systems extending $\lambda_\cap$ and $\lambda^s_\cap$ with ω, and prove in a uniform way weak normalization and head normalization for terms typable with certain kinds of types. We also discuss applicability of the method to other reduction properties.

The extended systems are listed in Table 4. The systems $\lambda_{\cap\omega}$ and $\lambda^s_{\cap\omega}$ are obtained from $\lambda_\cap$ and $\lambda^s_\cap$, respectively, by adding the type constant ω and the (ω) rule. The system $\lambda^l_{\cap\omega}$ is obtained from $\lambda^s_{\cap\omega}$ by replacing the $(Beta)^s$ rule by the $(Beta)^l$ rule which is a general form of the rule considered in [23] (σ is restricted to type atoms in [23]). In order to distinguish the judgements of the systems, we use the symbols $\vdash_\omega$, $\vdash_{s\omega}$ and $\vdash_{l\omega}$.

To prove properties of terms typable in the extended systems, it is necessary to clarify the relationship among them. First we show that the terms typable in the ordinary natural deduction style system $\lambda_{\cap\omega}$ are typable in $\lambda^s_{\cap\omega}$, in almost the same way as in the previous section.

Theorem 8. *If $\Gamma \vdash_\omega M : \sigma$ then $\Gamma \vdash_{s\omega} M : \sigma$.*

Proof. It is easy to see that Lemmas 5 through 9 hold for $\lambda^s_{\cap\omega}$ instead of $\lambda^s_\cap$. Then the theorem follows by induction on the derivation of $\Gamma \vdash_\omega M : \sigma$ in $\lambda_{\cap\omega}$. □

Next we relate the systems $\lambda^s_{\cap\omega}$ and $\lambda^l_{\cap\omega}$.

Lemma 10. *$\Gamma \vdash_{s\omega} M : \sigma$ if and only if $\Gamma \vdash_{l\omega} M : \sigma$.*

Proof. The implication from left to right is immediate by forgetting the right premise of $(Beta)^s$. For the converse, observe that the $(Beta)^l$ rule is derivable in $\lambda^s_{\cap\omega}$ using the rules $(Beta)^s$ and (ω). □

Note that the argument so far is independent of weak or head normalization.

Now we are ready to show various properties of terms typable in the type systems with ω. These properties are proved by induction on derivations in the system $\lambda^l_{\cap\omega}$.

First, remember that a term M is *weakly normalizing* if some β-reduction sequence starting from M terminates. The set of weakly normalizing terms is denoted by $\mathcal{WN}_\beta$. We say that a type σ is *ω-free* if ω does not occur in σ. For the system $\lambda^l_{\cap\omega}$ we have the following theorem.

Theorem 9. *Let $\Gamma \vdash_{l\omega} M : \sigma$ where σ and all types in Γ are ω-free. Then $M \in \mathcal{WN}_\beta$.*

Proof. By induction on the derivation of $\Gamma \vdash_{l\omega} M : \sigma$, noting that if σ and all types in Γ are ω-free then so are the types in the premises of the last applied rule in the derivation. □

Corollary 6. *Let $\Gamma \vdash_{\omega} M : \sigma$ where σ and all types in Γ are ω-free. Then $M \in \mathcal{WN}_\beta$.*

Proof. By Theorem 8, Lemma 10 and Theorem 9. □

Similarly, some of the properties discussed in the previous sections can also be obtained. For instance, we have the following.

Theorem 10. *Let $\Gamma \vdash_{l\omega} M : \sigma$ where σ and all types in Γ are ω-free. Then*

1. *$M \in \mathcal{CR}_\beta$,*
2. *$M \in \mathcal{ST}_\beta$.*

Table 4. Systems extended with ω

$$\frac{}{\Gamma \vdash M : \omega}\ (\omega) \qquad \frac{\Gamma \vdash M[x := N]N_1 \ldots N_n : \sigma}{\Gamma \vdash (\lambda x.M)NN_1 \ldots N_n : \sigma}\ (Beta)^l$$

	Notation
$\lambda_{\cap\omega} := \lambda_\cap + (\omega)$	$\Gamma \vdash_\omega M : \sigma$
$\lambda^s_{\cap\omega} := \lambda^s_\cap + (\omega)$	$\Gamma \vdash_{s\omega} M : \sigma$
$\lambda^l_{\cap\omega} := \lambda^s_{\cap\omega} - (Beta)^s + (Beta)^l$	$\Gamma \vdash_{l\omega} M : \sigma$

Proof. By induction on the derivation of $\Gamma \vdash_{l\omega} M : \sigma$, similarly to the proofs of Theorems 3 and 4. Note that in the proofs of Theorems 3 and 4, the induction hypothesis of the right premise is not used in the case where the last applied rule is $(Beta)^s$, so the proofs work also for the rule $(Beta)^l$. $\square$

Corollary 7. *Let $\Gamma \vdash_\omega M : \sigma$ where σ and all types in Γ are ω-free. Then*

1. $M \in \mathcal{CR}_\beta$,
2. $M \in \mathcal{ST}_\beta$.

Proof. By Theorem 8, Lemma 10 and Theorem 10. $\square$

Next we consider head normalization. A term M is said to be *head normalizing* if the head reduction sequence starting from M terminates. (The notion of head reduction is defined in Section 4.) The set of head normalizing terms is denoted by $\mathcal{HN}$. Also, the *non-trivial* types are defined as follows: type atoms are non-trivial, $\sigma \cap \tau$ is non-trivial if one of σ or τ is non-trivial, and $\sigma \to \tau$ is non-trivial if τ is non-trivial. For terms typable with non-trivial types in the system $\lambda^l_{\cap\omega}$, we have the following theorem.

Theorem 11. *Let $\Gamma \vdash_{l\omega} M : \sigma$ where σ is non-trivial. Then $M \in \mathcal{HN}$.*

Proof. By induction on the derivation of $\Gamma \vdash_{l\omega} M : \sigma$. $\square$

Corollary 8. *Let $\Gamma \vdash_\omega M : \sigma$ where σ is non-trivial. Then $M \in \mathcal{HN}$.*

Proof. By Theorem 8, Lemma 10 and Theorem 11. $\square$

The properties discussed in the previous sections can not in general be obtained for terms typable with non-trivial types, because we can not say anything about terms typed with ω.

8 Related Work on Intersection Types

As we mentioned earlier, our system $\lambda^s_\cap$ is a modification of Valentini's system [23]. The main difference between Valentini's system and ours was discussed in Section 5. Valentini also proposed a system that characterizes weakly normalizing terms without the type constant ω, but it was not related to the original natural deduction style system. In consequence, he could not show any properties of the original system, such as our Corollaries 6 through 8.

In [18], van Raamsdonk and Severi proved strong normalization for terms typable in the system $\lambda_\cap$ using an inductive characterization of strongly normalizing terms (cf. Section 5). Their proof is more complicated than ours, since they use a Generation Lemma, which is usually used for proving the converse of the theorem, i.e., that all strongly normalizing terms are typable. A similar proof of strong normalization for intersection types is found in [14] which discusses a lambda calculus with generalized applications. Our proof in Section 6 avoids the use of a Generation Lemma by substituting Lemmas 6 and 7. In [18, 14], strong normalization was the only property treated, and they did not consider any system with the type constant ω.

In [4], there is another attempt to prove strong normalization for terms typable with intersection types without using the reducibility method. However, the proof has a gap, because unlike in the simply typed case the set of typable and strongly normalizing terms is not closed under substitution (hence, Lemma 18(1) of [4] is not correct). Since a variable may have two different types in a term, it is necessary to specify the types of variables in the basis of the judgement that is derived with the term.

Other syntactic proofs of strong normalization for terms typable with intersection types are found in [11, 2], where the problem is reduced to that of weak normalization with respect to a new calculus or to a new notion of reduction. The proofs in [18, 23] and ours are different from those in [11, 2] in that strong normalization is proved directly rather than inferring it from weak normalization. Yet another syntactic proof [3] uses a translation from terms typable with intersection types into simply typed terms.

On the other hand, the (semantic) reducibility method has been used to prove strong normalization for terms typable with intersection types [17, 13, 6, 7]. Krivine [13] and Gallier [6] also applied the reducibility method to prove weak normalization and head normalization for terms typable with ω-free and nontrivial types, respectively, in the system with ω. Ghilezan et al. [7] gave sufficient conditions for applying the reducibility method to prove various reduction properties of terms typable in the system without ω. Compared with proofs in [13, 6, 7], our proofs are more uniform in that the key theorems (Theorems 7 and 8) for both the systems with and without ω are proved in almost the same way, and the proofs of weak and head normalization are both by induction on derivations in the same system $\lambda^{l}_{\cap\omega}$.

9 Conclusion

We have presented a general method for proving properties of typed lambda terms, and compared in detail the method and other methods with and without reducibility. Our method turned out to provide the most general conditions among them as well as uniform proofs for various properties of terms typable in simple and intersection type systems.

A similar approach to the present work was taken by Goguen [9] for a logical framework using dependent types. His presentation is based on typed operational semantics [8]. He adapted the proof of strong normalization without reducibility in [10] to prove some properties for the logical framework. Although his approach is limited to type systems with strongly normalizing terms, our method works as well for type systems with non-normalizing terms, such as the intersection type system with ω.

References

1. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, Amsterdam, revised edition, 1984.
2. G. Boudol. On strong normalization in the intersection type discipline. In *Proceedings of TLCA '03*, Lecture Notes in Computer Science 2701, pages 60–74. Springer-Verlag, 2003.
3. A. Bucciarelli, A. Piperno, and I. Salvo. Intersection types and λ-definability. *Mathematical Structures in Computer Science*, 13:15–53, 2003.
4. R. David. Normalization without reducibility. *Annals of Pure and Applied Logic*, 107:121–130, 2001.
5. M. Dezani-Ciancaglini, E. Giovannetti, and U. de'Liguoro. Intersection types, λ-models, and Böhm trees. In *Theories of Types and Proofs*, volume 2 of *MSJ Memoirs*, pages 45-97. Mathematical Society of Japan, Tokyo, 1998.
6. J. Gallier. Typing untyped λ-terms, or reducibility strikes again! *Annals of Pure and Applied Logic*, 91:231–270, 1998.
7. S. Ghilezan, V. Kunčak, and S. Likavec. Reducibility method for termination properties of typed lambda terms. In *Proceedings of 5th International Workshop on Termination*, pages 14–16, 2001.
8. H. Goguen. Typed operational semantics. In *Proceedings of TLCA '95*, Lecture Notes in Computer Science 902, pages 186–200. Springer-Verlag, 1995.
9. H. Goguen. Soundness of the logical framework for its typed operational semantics. In *Proceedings of TLCA '99*, Lecture Notes in Computer Science 1581, pages 177–197. Springer-Verlag, 1999.
10. F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed λ-calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic*, 42:59–87, 2003.
11. A. J. Kfoury and J. B. Wells. New notions of reduction and non-semantic proofs of strong β-normalization in typed λ-calculi. In *Proceedings of LICS'95*, pages 311–321. IEEE Computer Society Press, 1995.
12. G. Koletsos and G. Stavrinos. The structure of reducibility proofs. In *Proceedings of the Second Panhellenic Logic Symposium*, pages 138–144, 1999.
13. J.-L. Krivine. *Lambda-Calculus, Types and Models.* Ellis Horwood, Hemel Hempstead, 1993.
14. R. Matthes. Characterizing strongly normalizing terms of a λ-calculus with generalized applications via intersection types. In *Proceedings of ICALP Satellite Workshops 2000*, pages 339–354. Carleton Scientific, 2000.
15. J. C. Mitchell. Type systems for programming languages. In *Handbook of Theoretical Computer Science*, volume B, pages 365–458. Elsevier, Amsterdam, 1990.
16. J. C. Mitchell. *Foundations for Programming Languages.* MIT Press, Boston, 1996.
17. G. Pottinger. A type assignment for the strongly normalizable λ-terms. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 561–577. Academic Press, London, 1980.
18. F. van Raamsdonk and P. Severi. On normalisation. Technical Report CS-R9545, CWI, 1995.
19. F. van Raamsdonk, P. Severi, M. H. B. Sørensen, and H. Xi. Perpetual reductions in λ-calculus. *Information and Computation*, 149:173–225, 1999.
20. R. Statman. Logical relations and the typed λ-calculus. *Information and Control*, 65:85–97, 1985.
21. W. W. Tait. Intensional interpretations of functionals of finite type I. *The Journal of Symbolic Logic*, 32:198–212, 1967.
22. A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge Tracts in Theoretical Computer Science 43. Cambridge University Press, 2nd edition, 2000.
23. S. Valentini. An elementary proof of strong normalization for intersection types. *Archive for Mathematical Logic*, 40:475–488, 2001.