Uniform Proofs of Normalisation and Approximation for Intersection Types

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

We present intersection type systems in the style of sequent calculus, modifying the systems that Valentini introduced to prove normalisation properties without using the reducibility method. Our systems are more natural than Valentini's ones and equivalent to the usual natural deduction style systems. We prove the characterisation theorems of strong and weak normalisation through the proposed systems, and, moreover, the approximation theorem by means of direct inductive arguments. This provides in a uniform way proofs of the normalisation and approximation theorems via type systems in sequent calculus style.

1 Introduction

A traditional way of proving strong normalisation for typed λ-terms is the reducibility method [20], which uses set-theoretic comprehension. Other methods without using reducibility have also been studied in the literature (see, e.g. Section 5 of [19] for a review of those methods). Some of them use an inductive characterisation of strongly normalising λ-terms given by van Raamsdonk and Severi [18]. In [21], Valentini introduced, instead of using the inductive characterisation, an intersection type system that is closed under the rules of the original system, and proved strong normalisation by a simple induction on the typing derivation.

In this paper we develop Valentini's approach further providing an improvement on his system and its extensions with an axiom for the type constant ω. These systems are in the style of sequent calculus and equivalent to the original intersection type systems in natural deduction style. Using the new systems, we prove the characterisation theorems of strong and weak normalisation, which are well-known properties of intersection type systems [17, 8].

Another important point in our approach is that we design new systems that derive the same sequents as the original natural deduction style systems do, so that we can prove various other properties than normalisation by simple inductions on the typing derivation (cf. [15]). In the present paper we illustrate that by showing the approximation theorem for the type system with ω, which is usually proved using reducibility predicates over a typing context and a type (see, e.g. [11, 4]).

The difference between the systems in [21] and ours is the following. First, some rules of the systems in [21] have restrictions on types to be type variables. Also, the rule for abstraction takes a form that implies the η-rule. On the other hand, our systems do not have the restrictions on types, and our rule for abstraction is the usual one. In this natural setting, we show that our system is closed under the rules of the original natural deduction style system. This part of the proof of strong normalisation is much shorter than that in [21]. Secondly, the system characterising weakly normalising λ-terms in [21] does not have the type constant ω, and is not related to the original natural deduction style system. In this paper, we introduce new systems with an axiom for the type constant ω, and prove weak normalisation

Jakob Rehof (Ed.): Intersection Types and Related Systems (ITRS)
EPTCS 177, 2015, pp. 10–23, doi:10.4204/EPTCS.177.2

of λ-terms that are typable with ω-free types in the original system. The closure under the rules of the original system is shown by almost the same argument as that in the case of the system without ω.

In [21], only normalisation properties are discussed, and other properties than normalisation are not proved using the sequent calculus style systems. Some other papers [18, 16, 9, 1] have studied strong normalisation for terms typable with intersection types without using reducibility. Each of them uses an inductive characterisation of strongly normalising terms, but any other properties than normalisation have not been treated. So the present paper seems to be the first to apply a proof method for normalisation without reducibility to other properties of intersection type systems.

There is also an attempt in [3] to give uniform proofs of the characterisation theorems of normalisation and the approximation theorem. The method is through strong normalisation for reduction on typing derivations. However, it uses reducibility predicates to prove the strong normalisation, and the proof seems more complicated than ours.

The organisation of the paper is as follows. In Section 2 we introduce two kinds of intersection type systems. In Section 3 we prove the characterisation theorem of strong normalisation through the new type system. In Section 4 we introduce type systems with ω, and prove the characterisation theorem of weak normalisation. In Section 5 we prove the approximation theorem using one of the new systems with ω.

2 Intersection type systems

In this section we introduce two intersection type systems: one is in the ordinary natural deduction style and the other in sequent calculus style. They prove to be equivalent, and both characterise strongly normalising λ-terms.

First we introduce some basic notions on the λ-calculus [5]. The set Λ of λ-terms 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 $M,N,P,\ldots$ for λ-terms. The notions of free and bound variables are defined as usual. The set of free variables occurring in a λ-term M is denoted by $\mathsf{FV}(M)$. We identify α-convertible λ-terms, and use $\equiv$ to denote syntactic equality modulo α-conversion. $\_[\_ := \_]$ is used for usual capture-free substitution.

The β-rule is stated as $(\lambda x.M)N \to M[x := N]$, and β-reduction is the contextual closure of the β-rule. We use $\longrightarrow_\beta$ for one-step reduction, and $\longrightarrow_\beta^*$ for its reflexive transitive closure. A λ-term M is said to be *strongly* (*weakly*) *normalising* if all (some, respectively) β-reduction sequences starting from M terminate. The set of strongly (weakly) normalising λ-terms is denoted by SN^β (WN^β, respectively).

The set of types is defined by the grammar: $\sigma ::= \varphi \mid \sigma \to \sigma \mid \sigma \cap \sigma$ where φ ranges over a denumerable set of type variables. We use letters $\sigma,\tau,\rho,\ldots$ for arbitrary types. The type assignment systems $\lambda_\cap$

$$\frac{}{\Gamma, x:\sigma \vdash x:\sigma}\,(\mathsf{Ax}) \qquad \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau}\,(\to \mathsf{I}) \quad \text{where } x \notin \Gamma \qquad \frac{\Gamma \vdash M:\sigma \quad \Gamma \vdash M:\tau}{\Gamma \vdash M : \sigma \cap \tau}\,(\cap \mathsf{I})$$

$$\frac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash MN : \tau}\,(\to \mathsf{E}) \qquad \frac{\Gamma \vdash M:\sigma \cap \tau}{\Gamma \vdash M:\sigma}\,(\cap \mathsf{E}) \qquad \frac{\Gamma \vdash M:\sigma \cap \tau}{\Gamma \vdash M:\tau}\,(\cap \mathsf{E})$$

Figure 1: Natural deduction style system $\lambda_\cap$

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

$$\frac{\Gamma \vdash_s N:\sigma_1 \quad \Gamma, y:\sigma_2 \vdash_s yN_1\ldots N_n:\tau}{\Gamma, x:\sigma_1 \to \sigma_2 \vdash_s xNN_1\ldots N_n:\tau}\ (\mathsf{L}\to) \qquad \frac{\Gamma, x:\sigma \vdash_s M:\tau}{\Gamma \vdash_s \lambda x.M:\sigma \to \tau}\ (\mathsf{R}\to)$$

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

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

Figure 2: Sequent calculus style system $\lambda_\cap^s$

and $\lambda_\cap^s$ are defined by the rules in Figures 1 and 2, respectively. A *typing context* is defined as a finite set of pairs $\{x_1:\sigma_1,\ldots,x_n:\sigma_n\}$ where the variables are pairwise distinct in the system $\lambda_\cap$ while they may be the same in the system $\lambda_\cap^s$. A variable with different types is intended to have the type of intersection of all of them. The typing context $\Gamma, x:\sigma$ denotes the union $\Gamma \cup \{x:\sigma\}$, and $x \notin \Gamma$ means that x does not appear in Γ, i.e., for no type σ, $x:\sigma \in \Gamma$. Note that $x:\sigma \in \Gamma$ is possible in the typing context $\Gamma, x:\sigma$. In particular, the premisses of the rule $(\mathsf{L}\to)$ may have $x:\sigma_1 \to \sigma_2$ in Γ. In that case, $x:\sigma_1 \to \sigma_2$ is introduced by the rule $(\mathsf{L}\to)$ with implicit contraction.

The system in [21] has the restriction in $\lambda_\cap^s$ that the type σ in the rules (Ax) and $(\mathsf{Beta})^s$ and the type τ in the rules $(\mathsf{L}\to)$ and $(\mathsf{L}\cap)$ must be type variables. Also, the rule $(\mathsf{R}\to)$ takes the following form:

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

where $x \notin \Gamma$ and $x \notin \mathsf{FV}(M)$, so that the system includes the η-rule and is not equivalent to the system $\lambda_\cap$. (For example, $\vdash_s \lambda x.x:(\sigma \to \tau) \to ((\rho \cap \sigma) \to \tau)$ is derivable in the system of [21], but $\vdash \lambda x.x:(\sigma \to \tau) \to ((\rho \cap \sigma) \to \tau)$ is not derivable in $\lambda_\cap$.)

Example 2.1. Self-application can now be typed naturally in $\lambda_\cap^s$, as follows (cf. [21, pp. 478–479]).

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

The $(\mathsf{Beta})^s$-free part of the system $\lambda_\cap^s$ types exactly the terms in β-normal form, and any β-redex in a typed term must be constructed through the rule $(\mathsf{Beta})^s$. So it is immediately seen that the terms that are not head-normalising (e.g. $(\lambda x.xx)(\lambda x.xx)$) can not be typed in the system $\lambda_\cap^s$.

Proposition 2.2. *$\Gamma, x:\sigma_1 \cap \sigma_2 \vdash_s M:\tau$ if and only if $\Gamma, x:\sigma_1, x:\sigma_2 \vdash_s M:\tau$.*

Proof. By induction on the derivations. □

Henceforth we write $\Gamma_\cap$ for the typing context in which each variable has the type of intersection of all the types that the variable has in Γ.

3 Characterisation of strongly normalising λ-terms

If one tries to prove strong normalisation for terms typed in the system $\lambda_\cap$ directly by induction on derivations, a difficulty arises in the case of the rule $(\rightarrow \mathsf{E})$. One way of overcoming this difficulty is to use reducibility predicates [20]. Here we use the sequent calculus style system $\lambda_\cap^s$ instead. For the system $\lambda_\cap^s$, we can prove strong normalisation for typed terms directly by induction on derivations.

Theorem 3.1. *If $\Gamma \vdash_s M : \sigma$ then $M \in \mathsf{SN}^\beta$.*

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

$$\begin{array}{rcl}(\lambda x.M)NN_1\dots N_n & \longrightarrow_\beta^* & (\lambda x.M')N'N_1'\dots N_n' \\ & \longrightarrow_\beta & M'[x := N']N_1'\dots N_n' \\ & \longrightarrow_\beta & \dots\end{array}$$

where $M \longrightarrow_\beta^* M'$, $N \longrightarrow_\beta^* N'$ and $N_i \longrightarrow_\beta^* N_i'$ for $i \in \{1, \dots, n\}$. But then there is an infinite reduction sequence

$$\begin{array}{rcl}M[x := N]N_1\dots N_n & \longrightarrow_\beta^* & M'[x := N']N_1'\dots N_n' \\ & \longrightarrow_\beta & \dots\end{array}$$

contradicting the hypothesis. Hence $(\lambda x.M)NN_1 \dots N_n \in \mathsf{SN}^\beta$. □

To complete a proof of strong normalisation for terms typed in the system $\lambda_\cap$, what remains to be shown is that if M is typable in $\lambda_\cap$ then it is typable in $\lambda_\cap^s$. This is proved using several lemmas below. First we show that $\lambda_\cap^s$ is closed under the weakening rule.

Lemma 3.2. *If $\Gamma \vdash_s M : \tau$ then $\Gamma, x : \sigma \vdash_s M : \tau$.*

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

The next two lemmas are the essential difference from the proof of [21]. These are used in the proof of Lemma 3.5 below. The simply typed counterpart of Lemma 3.3 is found in the second proof of strong normalisation for the simply typed λ-calculus in [12].

Lemma 3.3. *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 $\Gamma \vdash_s M : \sigma \rightarrow \tau$. Here we show a few cases.

- $\dfrac{}{\Gamma, y : \sigma \rightarrow \tau \vdash_s y : \sigma \rightarrow \tau}\ (\mathsf{Ax})$

  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 $\Gamma, x : \sigma, y : \sigma \rightarrow \tau \vdash_s yx : \tau$ by an instance of the $(\mathsf{L} \rightarrow)$ rule.

- $\dfrac{\Gamma \vdash_s M[y := N]N_1 \dots N_n : \sigma \rightarrow \tau \quad \Gamma \vdash_s N : \rho}{\Gamma \vdash_s (\lambda y.M)NN_1 \dots N_n : \sigma \rightarrow \tau}\ (\mathsf{Beta})^s$

  By the induction hypothesis, we have $\Gamma, x : \sigma \vdash_s M[y := N]N_1 \dots N_n x : \tau$, and by Lemma 3.2, 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 $(\mathsf{Beta})^s$ rule.

- $\dfrac{\Gamma, y:\sigma \vdash_s M:\tau}{\Gamma \vdash_s \lambda y.M:\sigma\to\tau}\ (\mathsf{R}\to)$

  where $y \notin \Gamma$. From $\Gamma, y:\sigma \vdash_s M:\tau$, we have $\Gamma, x:\sigma \vdash_s M[y:=x]:\tau$. From this and the axiom $\Gamma, x:\sigma \vdash_s x:\sigma$, we obtain $\Gamma, x:\sigma \vdash_s (\lambda y.M)x:\tau$ by an instance of the $(\mathsf{Beta})^s$ rule. □

Lemma 3.4. *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 $\Gamma \vdash_s M:\sigma\cap\tau$. □

Now we are in a position to prove the following important lemma.

Lemma 3.5. *$\lambda^s_\cap$ is closed under substitution, i.e., if $\Gamma, x:\sigma_1,\dots,x:\sigma_m \vdash_s P:\tau$ where $x\notin\Gamma$, $m\geq 0$ and $\sigma_i\neq\sigma_j$ for $i\neq j$, and, for any $i\in\{1,\dots,m\}$, $\Gamma\vdash_s N:\sigma_i$, then $\Gamma\vdash_s P[x:=N]:\tau$.*

Proof. The proof is by main induction on the number of '$\to$' and '$\cap$' occurring in $\sigma_1,\dots,\sigma_m$ and subinduction on the length of the derivation of $\Gamma, x:\sigma_1,\dots,x:\sigma_m \vdash_s P:\tau$. We proceed by case analysis according to the last rule used in the derivation of $\Gamma, x:\sigma_1,\dots,x:\sigma_m \vdash_s P:\tau$. Here we consider a few cases.

- Suppose the last rule in the derivation is

  $$\dfrac{\Gamma,\overline{x}:\overline{\sigma}\vdash_s M[y:=Q]N_1\dots N_n:\tau \quad \Gamma,\overline{x}:\overline{\sigma}\vdash_s Q:\rho}{\Gamma,\overline{x}:\overline{\sigma}\vdash_s (\lambda y.M)QN_1\dots N_n:\tau}\ (\mathsf{Beta})^s$$

  where $\overline{x}:\overline{\sigma} = x:\sigma_1,\dots,x:\sigma_m$. 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 a bound variable, we can assume that it does not occur in N. Hence the first judgement is

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

  From this and $\Gamma\vdash_s Q[x:=N]:\rho$, 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 $(\mathsf{Beta})^s$ rule.

- Suppose the last rule in the derivation is

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

  where $\{\overline{x}:\overline{\sigma},x:\rho_1\to\rho_2\}=\{x:\sigma_1,\dots,x:\sigma_m\}$, $y\notin\mathsf{FV}(N_1)\cup\dots\cup\mathsf{FV}(N_n)$ and $y\notin\Gamma,\overline{x}:\overline{\sigma}$. By the subinduction hypothesis, we obtain both

  $$\Gamma\vdash_s M[x:=N]:\rho_1 \tag{1}$$

  and

  $$\Gamma, y:\rho_2\vdash_s (yN_1\dots N_n)[x:=N]:\tau \tag{2}$$

Now consider the assumption $\Gamma \vdash_s N : \rho_1 \to \rho_2$ and a fresh variable z. Then by Lemma 3.3, we have $\Gamma, z : \rho_1 \vdash_s Nz : \rho_2$. From this and (1), we have $\Gamma \vdash_s NM[x := N] : \rho_2$ by the main induction hypothesis. Then, again by the main induction hypothesis, we obtain

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

from (2) and $\Gamma \vdash_s NM[x := N] : \rho_2$.

- Suppose the last rule in the derivation is

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

where $\{\overline{x} : \overline{\sigma}, x : \rho_1 \cap \rho_2\} = \{x : \sigma_1, \ldots, x : \sigma_m\}$. Then, applying Proposition 2.2 to the conclusion, we have $\Gamma, (\overline{x} : \overline{\sigma})', x : \rho_1, x : \rho_2 \vdash_s xN_1 \ldots N_n : \tau$ where $(\overline{x} : \overline{\sigma})' = \overline{x} : \overline{\sigma} \setminus \{x : \rho_1 \cap \rho_2\}$. Now, from the assumption $\Gamma \vdash_s N : \rho_1 \cap \rho_2$, we have $\Gamma \vdash_s N : \rho_1$ and $\Gamma \vdash_s N : \rho_2$ by Lemma 3.4. Hence, by the main induction hypothesis, we obtain $\Gamma \vdash_s NN_1[x := N] \ldots N_n[x := N] : \tau$. □

Now we can show that the system $\lambda_\cap^s$ is closed under the $(\to \mathsf{E})$ rule.

Lemma 3.6. *If $\Gamma \vdash_s M : \sigma \to \tau$ and $\Gamma \vdash_s N : \sigma$ then $\Gamma \vdash_s MN : \tau$.*

Proof. By Lemma 3.3, 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 can prove the announced theorem.

Theorem 3.7. *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 3.4 and 3.6. □

The converse of this theorem also holds when typing contexts are restricted to those of $\lambda_\cap$. To prove it, we need some lemmas on properties of the system $\lambda_\cap$.

Lemma 3.8. *If $\Gamma \vdash M : \tau$ and $z \notin \Gamma$ then $\Gamma, z : \sigma \vdash M : \tau$.*

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

Lemma 3.9. *$\lambda_\cap$ is closed under substitution, i.e., if $\Gamma, x : \sigma \vdash P : \tau$ where $x \notin \Gamma$ and $\Gamma \vdash N : \sigma$ then $\Gamma \vdash P[x := N] : \tau$.*

Proof. By induction on the derivation of $\Gamma, x : \sigma \vdash P : \tau$. □

Next we prove a Generation Lemma. For its statement we define a preorder on types.

Definition 3.10. *The relation $\leq$ on types is defined by the following axioms and rules:*

1. $\sigma \leq \sigma$
2. $\sigma \cap \tau \leq \sigma,\ \sigma \cap \tau \leq \tau$
3. $\sigma \leq \tau,\ \tau \leq \rho \Rightarrow \sigma \leq \rho$
4. $\sigma \leq \tau,\ \sigma \leq \rho \Rightarrow \sigma \leq \tau \cap \rho$

Lemma 3.11. *If $\Gamma \vdash M : \sigma$ and $\sigma \leq \tau$ then $\Gamma \vdash M : \tau$.*

Proof. By induction on the definition of $\sigma \leq \tau$. □

Lemma 3.12 (Generation Lemma)**.**

1. *$\Gamma \vdash MN : \sigma$ if and only if there exist $\sigma_1, \dots, \sigma_n, \tau_1, \dots, \tau_n$ $(n \geq 1)$ such that $\sigma_1 \cap \dots \cap \sigma_n \leq \sigma$ and, for all $i \in \{1, \dots, n\}$, $\Gamma \vdash M : \tau_i \to \sigma_i$ and $\Gamma \vdash N : \tau_i$.*
2. *$\Gamma \vdash \lambda x.M : \sigma$ if and only if there exist $\tau_1, \dots, \tau_n, \rho_1, \dots, \rho_n$ $(n \geq 1)$ such that $(\tau_1 \to \rho_1) \cap \dots \cap (\tau_n \to \rho_n) \leq \sigma$ and, for all $i \in \{1, \dots, n\}$, $\Gamma, x : \tau_i \vdash M : \rho_i$.*

Proof. The implications from right to left are immediate by the typing rules and Lemma 3.11. The converses are shown by induction on the derivations. □

Now we can prove a crucial lemma about type-checking in the system $\lambda_\cap$.

Lemma 3.13. *If $\Gamma \vdash M[x := N] : \sigma$ and $\Gamma \vdash N : \tau$ where $x \notin \Gamma$ then there exists a type ρ such that $\Gamma, x : \rho \vdash M : \sigma$ and $\Gamma \vdash N : \rho$.*

Proof. By induction on the structure of M, using Lemma 3.12. □

We are now ready to prove the equivalence between the systems $\lambda^s_\cap$ and $\lambda_\cap$.

Theorem 3.14. *$\Gamma \vdash_s M : \sigma$ if and only if $\Gamma_\cap \vdash M : \sigma$.*

Proof. The implication from right to left follows from Theorem 3.7 and Proposition 2.2. The converse is shown by induction on the derivation of $\Gamma \vdash_s M : \sigma$. If the last applied rule is $(\mathsf{Beta})^s$, we use Lemmas 3.12 and 3.13. □

Finally we show that all strongly normalising terms are typable in $\lambda^s_\cap$.

Theorem 3.15. *If $M \in \mathsf{SN}^\beta$ then there exist a typing context Γ and a type σ such that $\Gamma \vdash_s M : \sigma$.*

Proof. The proof is by main induction on the maximal length of all β-reduction sequences starting from M and subinduction on the structure of M. We analyse the possible cases according to the shape of the term M.

- $M \equiv x$ for some variable x. In this case we just have to take $x : \sigma \vdash_s x : \sigma$, which is an axiom.
- $M \equiv xN_1 \dots N_n$. By the subinduction hypothesis, for any $i \in \{1, \dots, n\}$, there exist a typing context Γ_i and a type σ_i such that $\Gamma_i \vdash_s N_i : \sigma_i$. Then consider the following derivation (recall that $\lambda^s_\cap$ is closed under the weakening rule):

$$\dfrac{\cup\Gamma_i \vdash_s N_n : \sigma_n \quad \cup\Gamma_i, y_n : \tau \vdash_s y_n : \tau}{\cup\Gamma_i, y_{n-1} : \sigma_n \to \tau \vdash_s y_{n-1}N_n : \tau}\ (\mathsf{L}\to)$$

$$\vdots$$

$$\dfrac{\cup\Gamma_i \vdash_s N_1 : \sigma_1 \qquad \dfrac{\cup\Gamma_i \vdash_s N_2 : \sigma_2 \quad \cup\Gamma_i, y_2 : \sigma_3 \to \dots \to \sigma_n \to \tau \vdash_s y_2N_3 \dots N_n : \tau}{\cup\Gamma_i, y_1 : \sigma_2 \to \dots \to \sigma_n \to \tau \vdash_s y_1N_2 \dots N_n : \tau}\ (\mathsf{L}\to)}{\cup\Gamma_i, x : \sigma_1 \to \dots \to \sigma_n \to \tau \vdash_s xN_1 \dots N_n : \tau}\ (\mathsf{L}\to)$$

- $M \equiv \lambda x.P$. By the subinduction hypothesis, there exist a typing context Γ and a type σ such that $\Gamma, x : \sigma_1, \dots, x : \sigma_n \vdash_s P : \sigma$ where $x \notin \Gamma$ and $n \geq 0$. Then we have $\Gamma \vdash_s \lambda x.P : \sigma_1 \cap \dots \cap \sigma_n \to \sigma$ by the $(\mathsf{L}\cap)$ and $(\mathsf{R}\to)$ rules. (We use a weakening rule instead of $(\mathsf{L}\cap)$ when $n = 0$.)
- $M \equiv (\lambda x.P)NN_1 \dots N_n$. By the main induction hypothesis, there exist a typing context Γ_1 and a type σ_1 such that $\Gamma_1 \vdash_s P[x := N]N_1 \dots N_n : \sigma_1$, and, by the subinduction hypothesis, there exist a typing context Γ_2 and a type σ_2 such that $\Gamma_2 \vdash_s N : \sigma_2$. Then, by the weakening and $(\mathsf{Beta})^s$ rules, we obtain $\Gamma_1, \Gamma_2 \vdash_s (\lambda x.P)NN_1 \dots N_n : \sigma_1$. □

It is interesting to note that in the above proof we do not use the $(\mathsf{R}\cap)$ rule at all, so it is redundant for characterising the strongly normalising λ-terms. The absence of the $(\mathsf{R}\cap)$ rule leads to a restriction on types that is similar to those investigated in [2].

The results in this section are summarised as follows.

Corollary 3.16. *For any λ-term M, the following are equivalent.*

1. *M is typable in $\lambda_\cap$.*
2. *M is typable in $\lambda_\cap^s$.*
3. *M is strongly normalising.*
4. *M is typable in $\lambda_\cap^s$ without using the $(\mathsf{R}\cap)$ rule.*

Proof. ($1 \Rightarrow 2$) This follows from Theorem 3.7.
($2 \Rightarrow 3$) This follows from Theorem 3.1.
($3 \Rightarrow 4$) This follows from the proof of Theorem 3.15.
($4 \Rightarrow 2$) This is trivial.
($2 \Rightarrow 1$) This follows from Theorem 3.14. □

4 Characterisation of weakly normalising λ-terms

In this section we are concerned with weak normalisation and some type systems obtained by extending the systems $\lambda_\cap$ and $\lambda_\cap^s$. The main goal of this section is to prove the characterisation theorem of weak normalisation in a similar way to that of strong normalisation in the previous section.

The extended systems are listed in Figure 3. First we introduce a new rule $(\mathsf{Beta})^l$, which is a general form of the rule considered in [21] (σ is restricted to type variables in [21]). Then the system $\lambda_\cap^l$ is obtained from $\lambda_\cap^s$ by replacing the $(\mathsf{Beta})^s$ rule by the $(\mathsf{Beta})^l$ rule. The systems $\lambda_{\cap\omega}$, $\lambda_{\cap\omega}^s$ and $\lambda_{\cap\omega}^l$ are obtained from $\lambda_\cap$, $\lambda_\cap^s$ and $\lambda_\cap^l$, respectively, by adding the type constant ω and the (ω) rule. In order to distinguish the judgements of the systems, we use the symbols $\vdash_l$, $\vdash_\omega$, $\vdash_{s\omega}$ and $\vdash_{l\omega}$.

For the system $\lambda_\cap^l$, we have the following theorem.

Theorem 4.1. *If $\Gamma \vdash_l M : \sigma$ then $M \in \mathsf{WN}^\beta$.*

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

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

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

Figure 3: Systems extended with ω

For characterisation of weak normalisation in terms of typability 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 4.2. *If $\Gamma \vdash_\omega M : \sigma$ then $\Gamma \vdash_{s\omega} M : \sigma$.*

Proof. It is easy to see that Lemmas 3.2 through 3.6 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 $\lambda^s_{\cap\omega}$, $\lambda^l_{\cap\omega}$ and $\lambda^l_\cap$. This completes one direction of the characterisation theorem of weak normalisation.

Lemma 4.3. *$\Gamma \vdash_{s\omega} M : \sigma$ if and only if $\Gamma \vdash_{l\omega} M : \sigma$.*

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

Lemma 4.4. *Suppose σ and all types in Γ are ω-free. Then $\Gamma \vdash_{l\omega} M : \sigma$ if and only if $\Gamma \vdash_l M : \sigma$.*

Proof. The implication from right to left is trivial. For the converse, observe that every type occurring in the derivation of $\Gamma \vdash_{l\omega} M : \sigma$ also occurs in Γ or σ. □

Corollary 4.5. *If $\Gamma \vdash_\omega M : \sigma$ where σ and all types in Γ are ω-free, then $M \in \mathsf{WN}^\beta$.*

Proof. By Theorem 4.2, Lemmas 4.3 and 4.4, and Theorem 4.1. □

Conversely, if a λ-term M is weakly normalising, then there exist a typing context Γ and a type σ, both ω-free, such that $\Gamma \vdash_\omega M : \sigma$. To prove this, we need the following lemmas on properties of the system $\lambda_{\cap\omega}$. These are shown in similar ways to the proofs of Lemmas 3.8 through 3.12.

Lemma 4.6. *If $\Gamma \vdash_\omega M : \tau$ and $z \notin \Gamma$ then $\Gamma, z : \sigma \vdash_\omega M : \tau$.*

Lemma 4.7. *$\lambda_\cap$ is closed under substitution, i.e., if $\Gamma, x : \sigma \vdash_\omega P : \tau$ where $x \notin \Gamma$ and $\Gamma \vdash_\omega N : \sigma$ then $\Gamma \vdash_\omega P[x := N] : \tau$.*

Definition 4.8. *The relation $\leq_\omega$ on types is defined by the axioms and rules in Definition 3.10 together with the axiom $\sigma \leq_\omega \omega$.*

Lemma 4.9. *If $\Gamma \vdash_\omega M : \sigma$ and $\sigma \leq_\omega \tau$ then $\Gamma \vdash_\omega M : \tau$.*

Lemma 4.10 (Generation Lemma)**.** *Let σ be any type with $\omega \not\leq_\omega \sigma$. Then*

1. *$\Gamma \vdash_\omega MN : \sigma$ if and only if there exist $\sigma_1, \dots, \sigma_n, \tau_1, \dots, \tau_n$ $(n \geq 1)$ such that $\sigma_1 \cap \cdots \cap \sigma_n \leq_\omega \sigma$ and, for all $i \in \{1, \dots, n\}$, $\Gamma \vdash_\omega M : \tau_i \to \sigma_i$ and $\Gamma \vdash_\omega N : \tau_i$.*

2. *$\Gamma \vdash_\omega \lambda x.M : \sigma$ if and only if there exist $\tau_1, \dots, \tau_n, \rho_1, \dots, \rho_n$ $(n \geq 1)$ such that $(\tau_1 \to \rho_1) \cap \cdots \cap (\tau_n \to \rho_n) \leq_\omega \sigma$ and, for all $i \in \{1, \dots, n\}$, $\Gamma, x : \tau_i \vdash_\omega M : \rho_i$.*

Now we can prove a crucial lemma about type-checking in the system $\lambda_{\cap\omega}$.

Lemma 4.11. *If $\Gamma \vdash_\omega M[x := N] : \sigma$ where $x \notin \Gamma$ then there exists a type ρ such that $\Gamma, x : \rho \vdash_\omega M : \sigma$ and $\Gamma \vdash_\omega N : \rho$.*

Proof. By induction on the structure of M, using Lemma 4.10. If $M \equiv y (\not\equiv x)$ or $\omega \leq_\omega \sigma$, then we take $\rho = \omega$. □

We can now prove that in the system $\lambda_{\cap\omega}$, types are preserved under the inverse of β-reduction.

Lemma 4.12. *If $\Gamma \vdash_\omega N : \sigma$ and $M \longrightarrow_\beta N$ then $\Gamma \vdash_\omega M : \sigma$.*

Proof. By induction on the structure of M, using Lemma 4.10. If M is the β-redex then we use Lemma 4.11. □

Now we can prove the announced theorem.

Theorem 4.13. *If $M \in \mathsf{WN}^\beta$ then there exist a typing context Γ and a type σ such that $\Gamma \vdash_\omega M : \sigma$ and both Γ and σ are ω-free.*

Proof. Let M' be a normal form of M. By Theorem 3.15, every normal form is typable in $\lambda^s_\cap$, so there exist a typing context Γ and a type σ, both ω-free, such that $\Gamma \vdash_\omega M' : \sigma$. Hence, by Lemma 4.12, we have $\Gamma \vdash_\omega M : \sigma$. □

We can also prove the equivalence of the systems $\lambda_{\cap\omega}$, $\lambda^s_{\cap\omega}$ and $\lambda^l_{\cap\omega}$.

Theorem 4.14. *For any typing context Γ, any λ-term M and any type σ, the following are equivalent.*

1. $\Gamma_\cap \vdash_\omega M : \sigma$.
2. $\Gamma \vdash_{s\omega} M : \sigma$.
3. $\Gamma \vdash_{l\omega} M : \sigma$.

Proof. ($1 \Rightarrow 2$) This follows from Theorem 4.2 and Proposition 2.2 with $\vdash_{s\omega}$ instead of $\vdash_s$.
($2 \Rightarrow 3$) This follows from Lemma 4.3.
($3 \Rightarrow 1$) This follows by induction on the length of the derivation of $\Gamma \vdash_{l\omega} M : \sigma$. If the last applied rule is $(\mathsf{Beta})^l$, we use Lemmas 4.10 and 4.11. □

The results in this section are summarised as follows.

Corollary 4.15. *For any λ-term M, the following are equivalent.*

1. *$\Gamma \vdash_\omega M : \sigma$ for some typing context Γ and type σ, both ω-free.*
2. *$\Gamma \vdash_{s\omega} M : \sigma$ for some typing context Γ and type σ, both ω-free.*
3. *$\Gamma \vdash_{l\omega} M : \sigma$ for some typing context Γ and type σ, both ω-free.*
4. *$\Gamma \vdash_l M : \sigma$ for some typing context Γ and type σ.*
5. *M is weakly normalising.*

Proof. ($1 \Rightarrow 2$) This follows from Theorem 4.2.
($2 \Rightarrow 3$) This follows from Lemma 4.3.
($3 \Rightarrow 4$) This follows from Lemma 4.4.
($4 \Rightarrow 5$) This follows from Theorem 4.1.
($5 \Rightarrow 1$) This follows from Theorem 4.13. □

5 Application to other properties

The sequent calculus style systems we introduced in the previous sections are very useful for proving properties of intersection type systems. In this section we illustrate that by giving a simple proof of the (logical) approximation theorem, a property that is usually proved using reducibility predicates parametrised by typing contexts (see, e.g. [11, 4]). Proofs of some other properties through the sequent calculus style systems are found in [15], which also makes a comparison between general conditions for applying the reducibility method and our approach.

For the statement of the approximation theorem, we introduce some preliminary definitions. The set of $\lambda\bot$-terms [5] is obtained by adding the constant $\bot$ to the formation rules of λ-terms. The type systems in the previous section are extended to those for $\lambda\bot$-terms, where any $\lambda\bot$-term containing $\bot$ is typable by the (ω) rule.

Definition 5.1. *The* approximation mapping α *from λ-terms to $\lambda\bot$-terms is defined inductively by*

$$\begin{aligned}\alpha(\lambda x_1 \ldots x_n.xN_1 \ldots N_m) &:= \lambda x_1 \ldots x_n.x\alpha(N_1)\ldots\alpha(N_m)\\ \alpha(\lambda x_1 \ldots x_n.(\lambda x.M)NN_1 \ldots N_m) &:= \lambda x_1 \ldots x_n.\bot\end{aligned}$$

where $n, m \geq 0$.

Lemma 5.2.

1. *If $\Gamma \vdash_{l\omega} \alpha(M) : \sigma$ and $M \longrightarrow_\beta^* N$ then $\Gamma \vdash_{l\omega} \alpha(N) : \sigma$.*
2. *Let $M \longrightarrow_\beta^* N$, $M \longrightarrow_\beta^* N'$, $\Gamma \vdash_{l\omega} \alpha(N) : \sigma$ and $\Gamma \vdash_{l\omega} \alpha(N') : \tau$. Then there exists N'' such that $M \longrightarrow_\beta^* N''$ and $\Gamma \vdash_{l\omega} \alpha(N'') : \sigma \cap \tau$.*

Proof. The first part is proved by induction on the derivation of $\Gamma \vdash_{l\omega} \alpha(M) : \sigma$. For the second part, we use confluence of β-reduction. □

Now the logical approximation theorem can be formulated as follows.

Theorem 5.3. *$\Gamma \vdash_\omega M : \sigma$ if and only if there exists M' such that $M \longrightarrow_\beta^* M'$ and $\Gamma \vdash_\omega \alpha(M') : \sigma$.*

Proof. ($\Rightarrow$) By Theorem 4.14, it suffices to show that if $\Gamma \vdash_{l\omega} M : \sigma$ then there exists M' such that $M \longrightarrow_\beta^* M'$ and $\Gamma \vdash_{l\omega} \alpha(M') : \sigma$. The proof is by induction on the derivation of $\Gamma \vdash_{l\omega} M : \sigma$. Here we consider some cases.

- $\dfrac{\Gamma \vdash_{l\omega} M[x := N]N_1 \ldots N_n : \sigma}{\Gamma \vdash_{l\omega} (\lambda x.M)NN_1 \ldots N_n : \sigma}\ (\mathsf{Beta})^l$

  By the induction hypothesis, there exists M' such that $M[x := N]N_1 \ldots N_n \longrightarrow_\beta^* M'$ and $\Gamma \vdash_{l\omega} \alpha(M') : \sigma$. This M' also satisfies $(\lambda x.M)NN_1 \ldots N_n \longrightarrow_\beta^* M'$.

- $\dfrac{\Gamma \vdash_{l\omega} N : \sigma_1 \quad \Gamma, y : \sigma_2 \vdash_{l\omega} yN_1 \ldots N_n : \tau}{\Gamma, x : \sigma_1 \to \sigma_2 \vdash_{l\omega} xNN_1 \ldots N_n : \tau}\ (\mathsf{L} \to)$

  where $y \notin \mathsf{FV}(N_1) \cup \cdots \cup \mathsf{FV}(N_n)$ and $y \notin \Gamma$. By the induction hypothesis, there exist $N', N_1', \ldots, N_n'$ such that $N \longrightarrow_\beta^* N'$, $N_i \longrightarrow_\beta^* N_i'$, $\Gamma \vdash_{l\omega} \alpha(N') : \sigma_1$ and $\Gamma, y : \sigma_2 \vdash_{l\omega} y\alpha(N_1') \ldots \alpha(N_n') : \tau$. Hence, by an instance of the $(\mathsf{L} \to)$ rule, we obtain $\Gamma, x : \sigma_1 \to \sigma_2 \vdash_{l\omega} x\alpha(N')\alpha(N_1') \ldots \alpha(N_n') : \tau$. So we take $xN'N_1' \ldots N_n'$ as M'.

- $\dfrac{\Gamma, x:\sigma \vdash_{l\omega} N:\tau}{\Gamma \vdash_{l\omega} \lambda x.N : \sigma \to \tau}$ $(\mathsf{R}\to)$

  where $x \notin \Gamma$. By the induction hypothesis, there exists N' such that $N \longrightarrow_{\beta}^{*} N'$ and $\Gamma, x:\sigma \vdash_{l\omega} \alpha(N'):\tau$. By an instance of the $(\mathsf{R}\to)$ rule, we obtain $\Gamma \vdash_{l\omega} \lambda x.\alpha(N') : \sigma \to \tau$. Since $\alpha(\lambda x.N') \equiv \lambda x.\alpha(N')$, we take $\lambda x.N'$ as M'.

- $\dfrac{\Gamma \vdash_{l\omega} M:\sigma \quad \Gamma \vdash_{l\omega} M:\tau}{\Gamma \vdash_{l\omega} M:\sigma\cap\tau}$ $(\mathsf{R}\cap)$

  By the induction hypothesis, there exist M_1, M_2 such that $M \longrightarrow_{\beta}^{*} M_1$, $M \longrightarrow_{\beta}^{*} M_2$, $\Gamma \vdash_{l\omega} \alpha(M_1):\sigma$ and $\Gamma \vdash_{l\omega} \alpha(M_2):\tau$. Then by Lemma 5.2(2), there exists M' such that $M \longrightarrow_{\beta}^{*} M'$ and $\Gamma \vdash_{l\omega} \alpha(M'):\sigma\cap\tau$.

($\Leftarrow$) We can show by induction on the derivation that if $\Gamma \vdash_{\omega} \alpha(M'):\sigma$ then $\Gamma \vdash_{\omega} M':\sigma$. Hence, by Lemma 4.12, we have $\Gamma \vdash_{\omega} M:\sigma$. □

Thus our method has been successfully applied to proving the approximation theorem for the mapping α and the system $\lambda_{\cap\omega}$. It is work in progress to give similar proofs of the approximation theorems for the η-approximation mapping α_{η}, which maps $\lambda x.\bot$ directly to $\bot$, and type systems with various preorders as discussed in [10, 11, 4].

6 Conclusion

We have presented uniform proofs of the characterisation theorems of normalisation properties and the approximation theorem. The proofs have been given via intersection type systems in sequent calculus style. As investigated in [15], our method can be considered to have embedded certain conditions for applying reducibility directly into the typing rules of the sequent calculus style systems. (See [13] for a recent survey of general conditions for applying the reducibility method.)

As mentioned in the introduction, there are some proofs [18, 16, 9, 1] of strong normalisation for terms typable with intersection types without using reducibility, but they have not considered any other properties than normalisation. Other syntactic proofs of strong normalisation for terms typable with intersection types are found in [14, 6], where the problem is reduced to that of weak normalisation with respect to another calculus or to another notion of reduction. The proofs of [18, 21] and ours are different from those of [14, 6] in that strong normalisation is proved directly rather than inferring it from weak normalisation. Yet another syntactic proof [7] uses a translation from terms typable with intersection types into simply typed λ-terms.

There are many directions for future work. In addition to the one indicated at the last paragraph of Section 5, it would be worth investigating the type inference and the inhabitation problems for intersection types by means of our sequent calculus style systems.

Acknowledgements I would like to thank Katsumasa Ishii for drawing my attention to Valentini's paper and pointing out that the system includes the η-rule. I also thank the anonymous reviewers of ITRS 2014 workshop for valuable comments. The figures of the derivations have been produced with Makoto Tatsuta's `proof.sty` macros.

References

[1] Andreas Abel (2007): *Syntactical strong normalization for intersection types with term rewriting rules*. In: *Proceedings of HOR'07*, pp. 5–12.

[2] Steffen van Bakel (1992): *Complete restrictions of the intersection type discipline*. *Theoretical Computer Science* 102, pp. 135–163, doi:10.1016/0304-3975(92)90297-S.

[3] Steffen van Bakel (2004): *Cut-elimination in the strict intersection type assignment system is strongly normalizing*. *Notre Dame Journal of Formal Logic* 45, pp. 35–63, doi:10.1305/ndjfl/1094155278.

[4] Henk Barendregt, Wil Dekkers & Richard Statman (2013): *Lambda Calculus with Types*. Cambridge University Press, doi:10.1017/CBO9781139032636.

[5] Henk P. Barendregt (1984): *The Lambda Calculus: Its Syntax and Semantics*, revised edition. North-Holland, Amsterdam.

[6] Gerard Boudol (2003): *On strong normalization in the intersection type discipline*. In: *Proceedings of TLCA'03*, Lecture Notes in Computer Science 2701, Springer-Verlag, pp. 60–74, doi:10.1007/3-540-44904-3_5.

[7] Antonio Bucciarelli, Adolfo Piperno & Ivano Salvo (2003): *Intersection types and λ-definability*. *Mathematical Structures in Computer Science* 13, pp. 15–53, doi:10.1017/S0960129502003833.

[8] Mario Coppo, Mariangiola Dezani-Ciancaglini & Betti Venneri (1981): *Functional characters of solvable terms*. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 27, pp. 45–58, doi:10.1002/malq.19810270205.

[9] René David (2001): *Normalization without reducibility*. *Annals of Pure and Applied Logic* 107, pp. 121–130, doi:10.1016/S0168-0072(00)00030-0.

[10] Mariangiola Dezani-Ciancaglini, Elio Giovannetti & Ugo de'Liguoro (1998): *Intersection types, λ-models, and Böhm trees*. In: *Theories of Types and Proofs*, *MSJ Memoirs* 2, Mathematical Society of Japan, Tokyo, pp. 45–97.

[11] Mariangiola Dezani-Ciancaglini, Furio Honsell & Yoko Motohama (2001): *Approximation theorems for intersection type systems*. *Journal of Logic and Computation* 11, pp. 395–417, doi:10.1093/logcom/11.3.395.

[12] Felix Joachimski & Ralph Matthes (2003): *Short proofs of normalization for the simply-typed λ-calculus, permutative conversions and Gödel's T*. *Archive for Mathematical Logic* 42, pp. 59–87, doi:10.1007/s00153-002-0156-9.

[13] Fairouz Kamareddine, Vincent Rahli & Joe B. Wells (2012): *Reducibility proofs in the λ-calculus*. *Fundamenta Informaticae* 121, pp. 121–152, doi:10.3233/FI-2012-773.

[14] Assaf J. Kfoury & Joe B. Wells (1995): *New notions of reduction and non-semantic proofs of strong β-normalization in typed λ-calculi*. In: *Proceedings of LICS'95*, IEEE Computer Society Press, pp. 311–321, doi:10.1109/LICS.1995.523266.

[15] Kentaro Kikuchi (2009): *On general methods for proving reduction properties of typed lambda terms*. In: *Proof theoretical study of the structure of logic and computation*, RIMS Kôkyûroku 1635, pp. 33–50. Available at http://hdl.handle.net/2433/140464. (Unrefereed proceedings).

[16] Ralph Matthes (2000): *Characterizing strongly normalizing terms of a λ-calculus with generalized applications via intersection types*. In: *Proceedings of ICALP Satellite Workshops 2000*, Carleton Scientific, pp. 339–354.

[17] Garrel Pottinger (1980): *A type assignment for the strongly normalizable λ-terms*. In: *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, London, pp. 561–577.

[18] Femke van Raamsdonk & Paula Severi (1995): *On normalisation*. Technical Report CS-R9545, CWI.

[19] Femke van Raamsdonk, Paula Severi, Morten Heine B. Sørensen & Hongwei Xi (1999): *Perpetual reductions in λ-calculus*. *Information and Computation* 149, pp. 173–225, doi:10.1006/inco.1998.2750.

[20] William W. Tait (1967): *Intensional interpretations of functionals of finite type I*. *The Journal of Symbolic Logic* 32, pp. 198–212, doi:`10.2307/2271658`.

[21] Silvio Valentini (2001): *An elementary proof of strong normalization for intersection types*. *Archive for Mathematical Logic* 40, pp. 475–488, doi:`10.1007/s001530000070`.