Electronic Notes in Theoretical Computer Science 70 No. 1 (2002)
URL: `http://www.elsevier.nl/locate/entcs/volume70.html` 18 pages

# Reducibility: a ubiquitous method in lambda calculus with intersection types

Silvia Ghilezan [1,3]

*École Normale Supérieure de Lyon, France,*
*Faculty of Engineering, University of Novi Sad,*
*Trg Dositeja Obradovića 6, Yugoslavia*

Silvia Likavec [1,2,4]

*Dipartimento di Informatica*
*Università di Torino*
*Corso Svizzera 185, 10149 Torino, Italy*

**Abstract**

A general reducibility method is developed for proving reduction properties of lambda terms typeable in intersection type systems with and without the universal type $\Omega$. Sufficient conditions for its application are derived. This method leads to uniform proofs of confluence, standardization, and weak head normalization of terms typeable in the system with the type $\Omega$. The method extends Tait's reducibility method for the proof of strong normalization of the simply typed lambda calculus, Krivine's extension of the same method for the strong normalization of intersection type system without $\Omega$, and Statman-Mitchell's logical relation method for the proof of confluence of $\beta\eta$-reduction on the simply typed lambda terms. As a consequence, the confluence and the standardization of all (untyped) lambda terms is obtained.

*Key words:* Lambda calculus, intersection types, reducibility method, confluence, standardization.

[1] Partially supported by grant 1630 "Representation of proofs with applications, classification of structures and infinite combinatorics" (of the Ministry of Science, Technology, and Development of Serbia).
[2] Partially supported by EU within the FET - Global Computing initiative, project DART ST-2001-33477 , and by MURST Cofin'01 project COMETA. The funding bodies are not responsible for any use that might be made of the results presented here.
[3] Email: `gsilvia@uns.ns.ac.yu`
[4] Email: `likavec@di.unito.it`

# 1 Introduction

The substantial idea of the reducibility method is to interpret types by suitable sets of lambda terms which satisfy certain realizability properties. The reducibility method, based on realizability interpretations, was introduced in Tait [20] for proving the strong normalization property for the simply typed lambda calculus and further developed in Girard [11] and Tait [21] for proving the strong normalization property for polymorphic (second order) lambda calculus. There is an overview of these proofs in Barendregt [3].

In Mitchell [16] and [15] this method is referred to as logical relations and it is discussed that apart from the strong normalization this method can be used for the proof of the confluence (Church-Rosser property) of $\beta\eta$-reduction and other basic results of the simply typed lambda calculus. The original proof of the Church-Rosser property of the simply typed lambda calculus using logical relations and the reducibility method is due to Statman [19] and Koletsos [12].

In Krivine [14] and later in Ghilezan [10] the reducibility method is applied in order to characterize all and only the strongly normalizing lambda terms in lambda calculus with intersection types without the universal type. The reducibility method is also used in Gallier [9] for characterizing some special classes of lambda terms such as strongly normalizing terms, normalizing terms, head normalizing terms, and weak head normalizing terms by their typeability in various intersection type systems. In Dezani et al. [7] and Dezani and Ghilezan [6] the reducibility method is applied to characterize both the mentioned terms and their persistent versions. The strong normalization of an intersection type system with explicit substitution is proved in Dougherty and Lescanne [8] using reducibility method.

This work presents the reducibility method as a general framework for proving reduction properties of lambda terms typeable in the intersection type system $\lambda\cap^\Omega$ with the type $\Omega$ and in the system $\lambda\cap$ without it. We distinguish two different kinds of type interpretation with respect to a given set $\mathcal{P} \subseteq \Lambda$. Also, we distinguish two different closure conditions which a given set $\mathcal{P} \subseteq \Lambda$ has to satisfy. By combining different type interpretations with appropriate closure conditions on $\mathcal{P} \subseteq \Lambda$ we prove the soundness of the type assignment in both cases. In this way a method for proving properties of lambda terms typeable with intersection types is obtained. We generalize the case of $\lambda\cap^\Omega$ obtaining a reducibility method for this system, which leads to uniform proofs of reduction properties of terms typeable in $\lambda\cap^\Omega$.

Our focus is on the system $\lambda\cap^\Omega$ since a consequence of the reducibility method applied on is the development of a proof methodology for untyped lambda calculus which could be used to prove properties with suitable invariance.

The paper is organized as follows. Section 2 is an overview of some basic notions regarding lambda terms, intersection types, and type assignment systems considered. In Section 3 we prove soundness of type assignment with respect to both type interpretations and develop the reducibility method for $\lambda\cap^\Omega$. As a consequence, we establish a method for proving reduction properties of untyped lambda terms. We show in Section 4 that the reducibility method represents a uniform way for proving the confluence of $\beta$-reduction, standardization, and weak head normalization property of terms typeable in $\lambda\cap^\Omega$. Also, we obtain the confluence of $\beta$-reduction and standardization on all lambda terms by the method for untyped lambda terms.

## 2 Terms, Types, and Type Systems

First, we present some preliminary notions of reductions on lambda terms, such as $\beta$-reduction, head reduction, and internal reduction. These notions can be found in Barendregt [2].

**Definition 2.1** The set $\Lambda$ of (untyped) lambda terms is defined by the following abstract syntax.

$$\boxed{\begin{array}{ccl} \Lambda & ::= & \mathtt{var} \mid \Lambda\Lambda \mid \lambda\mathtt{var}.\Lambda \\ \mathtt{var} & ::= & x \mid \mathtt{var}' \end{array}}$$

We use $x, y, z, \ldots$ as meta variables that range over term variables and $M, N, P, Q, \ldots$ for arbitrary terms.

$FV(M)$ denotes the set of free variables of a term $M$. By $M[x := N]$ we denote the term obtained by substituting the term $N$ for all the free occurrences of the variable $x$ in $M$, taking into account that free variables of $N$ remain free in the term obtained. The syntactical equality between terms is denoted by $\equiv$.

The main axiom of *$\beta$-reduction* is

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

where $(\lambda x.M)N$ is a $\beta$-redex. The transitive reflexive contextual closure of $\to_\beta$ is denoted by $\twoheadrightarrow_\beta$. The $\beta$-equality $=_{(\beta)}$ ($\beta$-conversion) is the symmetric transitive closure of $\twoheadrightarrow_\beta$.

If $M \equiv \lambda x_1 \ldots x_n.(\lambda x.M_0)M_1 \ldots M_m$, $n \geq 0$, $m \geq 1$, then $(\lambda x.M_0)M_1$ is called the *head-redex* of $M$ (Barendregt [2], p.173). We write $M \to_h M'$ if $M'$ is obtained from $M$ by reducing the head redex of $M$ (head reduction). We write $M \to_i N$ if $M'$ is obtained from $M$ by reducing a redex other than the head redex (internal reduction). We also use the transitive closures of these relations, notation $\twoheadrightarrow_h$ and $\twoheadrightarrow_i$, respectively.

A term is a *weak head normal form* if it starts with an abstraction, or with a variable. A term is *weakly head normalizing* if it reduces to a weak head normal form. Let $\mathcal{W}$ denote the set of all lambda terms that have a weak head normal form.

$$\mathcal{W} = \{M \in \Lambda \mid (\exists P, P_1, \ldots, P_n \in \Lambda)\ M \underset{\beta}{\twoheadrightarrow} \lambda x.P \text{ or } M \underset{\beta}{\twoheadrightarrow} xP_1 \ldots P_n\}.$$

Let $\mathcal{SN}$ denote the set of strongly normalizing terms, i.e.

$$\mathcal{SN} = \{M \in \Lambda \mid \neg(\exists M_1, M_2, \ldots \in \Lambda)\ M \underset{\beta}{\rightarrow} M_1 \underset{\beta}{\rightarrow} M_2 \underset{\beta}{\rightarrow} \ldots\}.$$

Next we present the *intersection type assignment systems* $\lambda\cap$ and $\lambda\cap^\Omega$ that are originated in Coppo and Dezani [4], [5], Pottinger [17], Sallé [18], and Barendregt et al. [1].

**Definition 2.2** The sets $\mathtt{type}$ and $\mathtt{type}^\Omega$ of types are defined as follows.

$$\begin{array}{|rcl|}\hline \mathtt{type} & ::= & \mathtt{atom} \mid \mathtt{type} \rightarrow \mathtt{type} \mid \mathtt{type} \cap \mathtt{type} \\ \mathtt{type}^\Omega & ::= & \mathtt{atom} \mid \Omega \mid \mathtt{type}^\Omega \rightarrow \mathtt{type}^\Omega \mid \mathtt{type}^\Omega \cap \mathtt{type}^\Omega \\ \mathtt{atom} & ::= & \alpha \mid \mathtt{atom}' \\ \hline \end{array}$$

We use $\alpha, \beta, \ldots$ as meta variables for arbitrary atoms and $\varphi, \sigma, \tau, \ldots$ for arbitrary types.

A *type assignment* is an expression of the form $M : \varphi$, where $M \in \Lambda$ and $\varphi \in \mathtt{type}$ or $\varphi \in \mathtt{type}^\Omega$. A *context* $\Gamma$ is a set $\{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ of type assignments with different term variables. Then $Dom\,\Gamma = \{x_1, \ldots, x_n\}$. A context extension $\Gamma, x : \sigma$ denotes the set $\Gamma \cup \{x : \sigma\}$, where $x \notin Dom\Gamma$.

**Definition 2.3** [Preorder on $\mathtt{type}$ and $\mathtt{type}^\Omega$ ]

(i) The relation $\leq$ is defined on $\mathtt{type}$ by the following axioms and rules:

1. $\sigma \leq \sigma$
2. $\sigma \leq \tau,\ \tau \leq \rho\ \Rightarrow \sigma \leq \rho$
3. $\sigma \cap \tau \leq \sigma,\ \sigma \cap \tau \leq \tau$
4. $(\sigma \rightarrow \rho) \cap (\sigma \rightarrow \tau) \leq \sigma \rightarrow \rho \cap \tau$
5. $\sigma \leq \tau,\ \sigma \leq \rho \Rightarrow \sigma \leq \tau \cap \rho$
6. $\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma \cap \tau \leq \sigma' \cap \tau'$
7. $\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma' \rightarrow \tau \leq \sigma \rightarrow \tau'$

(ii) The relation $\leq^\Omega$ is defined on $\mathtt{type}^\Omega$ by 1.-7. and the axioms:

8. $\sigma \leq \Omega$  9. $\sigma \rightarrow \Omega \leq \Omega \rightarrow \Omega$

(iii) The induced equivalence relations are defined by:

1. $\sigma \sim \tau \Leftrightarrow \sigma \leq \tau \ \&\ \tau \leq \sigma$  2. $\sigma \sim^\Omega \tau \Leftrightarrow \sigma \leq^\Omega \tau \ \&\ \tau \leq^\Omega \sigma$

The usual axiom of the preorder on intersection types is $\Omega \leq \Omega \rightarrow \Omega$ (Barendregt et al. [1]). Having this axiom one can distinguish head normalizing terms from unsolvable terms by their typeability, but cannot distinguish weakly head normalizing terms from unsolvable terms. Instead we adopt the axiom 8. $\sigma \rightarrow \Omega \leq \Omega \rightarrow \Omega$, which allows us to distinguish weakly head normalizing from unsolvable terms (Dezani et al. [7]).

**Definition 2.4** [Type assignment systems $\lambda\cap$ and $\lambda\cap^\Omega$]

(i) The pure intersection type system $\lambda\cap$ is generated on the set $\mathtt{type}$ by $(ax)$, $(\rightarrow E)$, $(\rightarrow I)$, $(\cap E)$, $(\cap I)$, and $(\leq)$ given in Figure 1. The deriveability in $\lambda\cap$ is denoted by $\Gamma \vdash P : \varphi$.

(ii) The system $\lambda\cap^{\Omega}$ is generated on the set $\mathtt{type}^{\Omega}$ by $(ax)$, $(\to E)$, $(\to I)$, $(\cap E)$, $(\cap I)$, $(\leq^{\Omega})$, and $(\Omega)$ given in Figure 1. The deriveability in the system $\lambda\cap^{\Omega}$ is denoted by $\Gamma \vdash^{\Omega} P : \varphi$.

$$(ax) \qquad \Gamma, x : \sigma \vdash x : \sigma$$

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

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

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

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

$$(\leq^{(\Omega)}) \qquad \frac{\Gamma \vdash M : \sigma, \quad \sigma \leq^{(\Omega)} \tau}{\Gamma \vdash M : \tau}$$

$$(\Omega) \qquad \overline{\Gamma \vdash M : \Omega}$$

Fig. 1. Axiom and rules

## 3 Reducibility Method

The reducibility method is a generally accepted way for proving the *strong normalization property* of various type systems such as the simply typed lambda calculus in Tait [20], the polymorphic lambda calculus in Tait [21] and Girard [11], and the pure intersection type assignment system in Krivine [14]. This method was applied to the proof of the Church-Rosser property (confluence) of $\beta\eta$-reduction in the simply typed lambda calculus in Statman [19], Koletsos [12], and Mitchell [16] and [15].

The general idea of the reducibility method is to provide a link between terms typeable in a type system and terms satisfying certain reduction properties (e.g. strong normalization, confluence). For that reason types are interpreted by suitable sets of lambda terms: saturated and stable sets in Tait [20] and Krivine [14] and admissible relations in Mitchell [16] and [15]. These interpretations are based on the sets of terms considered (e.g. strong normalization, confluence). Then the soundness of type assignment with respect to these interpretations is obtained. A consequence of soundness is that every term typeable in the type system belongs to the interpretation of its type. This is an intermediate step between the terms typeable in a type system and terms satisfying the reduction property considered.

We present the reducibility method as a general framework which leads to uniform proofs of the basic reduction properties of lambda terms typeable with intersection types. As a consequence of the reducibility method for $\lambda\cap^\Omega$ we establish a method for proving reduction properties of untyped lambda terms.

In order to develop the reducibility method we consider $\Lambda$ as the *applicative structure* whose domain are lambda terms and where the application is just the application of terms. Let us distinguish the following *type interpretations* with respect to a fixed subset $\mathcal{P} \subseteq \Lambda$.

**Definition 3.1** Let $\mathcal{P} \subseteq \Lambda$.

(i) The type interpretation $[\![-]\!] : \mathtt{type} \to 2^\Lambda$ is defined by:
 $(I1)$ $[\![\alpha]\!] = \mathcal{P}$, $\alpha$ is an atom;
 $(I2)$ $[\![\sigma \cap \tau]\!] = [\![\sigma]\!] \cap [\![\tau]\!]$;
 $(I3)$ $[\![\sigma \to \tau]\!] = [\![\sigma]\!] \Rightarrow [\![\tau]\!] = \{M \in \Lambda \mid \forall N \in [\![\sigma]\!] \quad MN \in [\![\tau]\!]\}$.

(ii) The $\Omega$-type interpretation $[\![-]\!]^\Omega : \mathtt{type}^\Omega \to 2^\Lambda$ is defined by
 $(\Omega 1)$ $[\![\alpha]\!]^\Omega = \mathcal{P}$, $\alpha$ is an atom;
 $(\Omega 2)$ $[\![\sigma \cap \tau]\!]^\Omega = [\![\sigma]\!]^\Omega \cap [\![\tau]\!]^\Omega$;
 $(\Omega 3)$ $[\![\sigma \to \tau]\!]^\Omega = [\![\sigma]\!]^\Omega \Rightarrow_\Omega [\![\tau]\!]^\Omega = ([\![\sigma]\!]^\Omega \Rightarrow [\![\tau]\!]^\Omega) \cap \mathcal{P} =$
 $= \{M \in \mathcal{W} \mid \forall N \in [\![\sigma]\!]^\Omega \quad MN \in [\![\tau]\!]^\Omega\}$;
 $(\Omega 4)$ $[\![\Omega]\!]^\Omega = \Lambda$.

On the set $\mathtt{type}$ we can define another type interpretation $[\![-]\!]_\mathcal{P} : \mathtt{type} \to 2^\Lambda$ by $(I1)$, $(I2)$, and a modified $(\Omega 3)$ restricted on $\mathtt{type}$, $[\![\sigma \to \tau]\!]_\mathcal{P} = \{M \in \mathcal{SN} \mid \forall N \in [\![\sigma]\!]_\mathcal{P} \quad MN \in [\![\tau]\!]_\mathcal{P}\}$. With the type interpretation $[\![-]\!]_\mathcal{P}$ one can prove, under slightly different conditions, the same properties as with the type interpretation $[\![-]\!]$, which is less restrictive, namely does not require the intersection with $\mathcal{SN}$ in the interpretation of $\to$. For this reason we decided to work with $[\![-]\!]$. On the other hand the $\Omega$-type interpretation cannot be changed if we want to obtain the required properties for the system $\lambda\cap^\Omega$. This means that the condition $(\Omega 3)$ cannot be replaced by the condition $(I3)$ in the definition of type interpretation on $\mathtt{type}^\Omega$.

The following property of $\Omega$-type interpretation is due to the condition $(\Omega 3)$ and axiom 8. of Definition 2.3 and is easy to verify. On the other hand this property is not true for type interpretation in general ( 3.1(i)), but only under some conditions which will be presented in Lemma 3.8.

**Lemma 3.2** *For every type $\varphi \in \mathtt{type}^{\Omega}$ such that $\varphi \not\sim \Omega$ we have that $[\![\varphi]\!]^{\Omega} \subseteq \mathcal{P}$.*

Let us further define a *valuation of terms* $[\![-]\!]_{\rho} : \Lambda \to \Lambda$ and the *semantic satisfiability relations* $\models$ and $\models^{\Omega}$ which connect two different kinds of type interpretations with term valuations.

In the remainder of this paper, all assertions of the form $[\![-]\!]^{(\Omega)}$ in a statement are to be interpreted either all as $[\![-]\!]$ or all as $[\![-]\!]^{\Omega}$. Similarly for $\models^{(\Omega)}$, $(\Omega)$-type interpretation, $\leq^{(\Omega)}$, and $\mathtt{type}^{(\Omega)}$.

**Definition 3.3** Let $\rho : \mathtt{var} \to \Lambda$ be a valuation of term variables in $\Lambda$. Then $[\![-]\!]_{\rho} : \Lambda \to \Lambda$ is defined as follows

$$[\![M]\!]_{\rho} = M[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n)], \;\text{ where }\; FV(M) = \{x_1, \ldots, x_n\}.$$

An alternation $\rho(x := N)$ of a valuation $\rho$ means that $\rho(x := N)(x) = N$ and $\rho(x := N)(y) = \rho(y)$, where $y \not\equiv x$. The following properties of term valuation will be needed in some proofs.

**Lemma 3.4***(i)* $[\![M]\!]_{\rho(x := N)} \equiv [\![M]\!]_{\rho(x := x)}[x := N]$.

*(ii)* $[\![MN]\!]_{\rho} \equiv [\![M]\!]_{\rho}[\![N]\!]_{\rho}$.

*(iii)* $[\![\lambda x.M]\!]_{\rho} \equiv \lambda x.[\![M]\!]_{\rho(x := x)}$.

**Definition 3.5**(i) $\rho \models^{(\Omega)} M : \varphi \quad$ *iff* $\quad [\![M]\!]_{\rho} \in [\![\varphi]\!]^{(\Omega)}$;

(ii) $\rho \models^{(\Omega)} \Gamma \quad$ *iff* $\quad (\forall (x : \varphi) \in \Gamma) \quad \rho(x) \in [\![\varphi]\!]^{(\Omega)}$;

(iii) $\Gamma \models^{(\Omega)} M : \varphi \quad$ *iff* $\quad (\forall \rho \models^{(\Omega)} \Gamma) \quad \rho \models^{(\Omega)} M : \varphi$.

Let us impose some conditions on $\mathcal{P} \subseteq \Lambda$.

**Definition 3.6** Let $\mathcal{P} \subseteq \Lambda$ be given. We say that:

($VAR$) $\mathcal{P}$ satisfies the variable property, notation $VAR(\mathcal{P})$, if

$$(\forall \varphi \in \mathtt{type})(\forall x \in \mathtt{var}) \;\; x \in [\![\varphi]\!];$$

($VAR^{\Omega}$) $\mathcal{P}$ satisfies the $\Omega$-variable property, notation $VAR^{\Omega}(\mathcal{P})$, if

$$(\forall \varphi \in \mathtt{type}^{\Omega})(\forall x \in \mathtt{var}) \;\; x \in [\![\varphi]\!]^{\Omega};$$

($SAT$) $\mathcal{P}$ is saturated, notation $SAT(\mathcal{P})$, if

$$(\forall M \in \Lambda)(\forall \varphi \in \mathtt{type})(\forall N \in \mathcal{P}) \; M[x := N] \in [\![\varphi]\!] \Rightarrow (\lambda x.M)N \in [\![\varphi]\!];$$

($SAT^{\Omega}$) $\mathcal{P}$ is $\Omega$-saturated, notation $SAT^{\Omega}(\mathcal{P})$, if

$$(\forall M, N \in \Lambda)(\forall \varphi \in \mathtt{type}^{\Omega}) M[x := N] \in [\![\varphi]\!]^{\Omega} \Rightarrow (\lambda x.M)N \in [\![\varphi]\!]^{\Omega};$$

($CLO$) $\mathcal{P}$ is closed by variable application, notation $CLO(\mathcal{P})$, if

$$(\forall \varphi \in \mathtt{type}) Mx \in [\![\varphi]\!] \quad \Rightarrow \quad M \in \mathcal{P};$$

($CLO^{\Omega}$) $\mathcal{P}$ is closed by abstraction, notation $CLO^{\Omega}(\mathcal{P})$, if

$$(\forall \varphi \in \mathtt{type}^{\Omega}) M \in [\![\varphi]\!]^{\Omega} \quad \Rightarrow \quad \lambda x.M \in \mathcal{P}.$$

We will show that these properties are sufficient to develop the reducibility method (Proposition 3.11 and 3.13).

The preorder on types is interpreted as the set theoretic inclusion.

**Lemma 3.7** *If* $\sigma \leq^{(\Omega)} \tau$*, then* $[\![\sigma]\!]^{(\Omega)} \subseteq [\![\tau]\!]^{(\Omega)}$.

**Proof.** By induction on the length of the derivation of $\sigma \leq^{(\Omega)} \tau$. □

As noticed above Lemma 3.2 cannot be proved for type interpretation ( 3.1(i)), but the presence of the conditions $\mathit{VAR}(\mathcal{P})$ and $\mathit{CLO}(\mathcal{P})$ provides a similar property.

**Lemma 3.8** *If* $\mathit{VAR}(\mathcal{P})$ *and* $\mathit{CLO}(\mathcal{P})$ *are satisfied, then* $[\![\varphi]\!] \subseteq \mathcal{P}$ *for all types* $\varphi \in \mathtt{type}$.

**Proof.** If $\varphi$ is an atom, then the statement follows immediately. The interesting case is when $\varphi \equiv \sigma \to \tau$. Take $M \in [\![\sigma \to \tau]\!]$ and show that $M \in \mathcal{P}$. By the definition of the type interpretation $[\![-]\!]$ we have that $M \in [\![\sigma]\!] \Rightarrow [\![\tau]\!]$. By $\mathit{VAR}(\mathcal{P})$ let us take $x \in [\![\sigma]\!]$. Then $Mx \in [\![\tau]\!]$. Hence $M \in \mathcal{P}$ by $\mathit{CLO}(\mathcal{P})$. □

**Remark 3.9** Let us notice that the conditions $\mathit{VAR}(\mathcal{P})$, $\mathit{SAT}(\mathcal{P})$, and $\mathit{CLO}(\mathcal{P})$ provide that $\mathcal{SN} \subseteq \mathcal{P}$. This fact is never used in the proofs but is necessary for the justification of the correctness of the method, since it is well known that the system $\lambda\cap$ completely characterizes all strongly normalizing terms.

Now we can prove the following realizability property, which is referred to as the *soundness property* or the *adequacy property* . More precisely, we prove soundness with respect to type interpretation in Proposition 3.10 and $\Omega$-type interpretation in Proposition 3.12.

**Proposition 3.10 (Soundness with respect to type interpretation)** *If* $\mathit{VAR}(\mathcal{P})$, $\mathit{SAT}(\mathcal{P})$, *and* $\mathit{CLO}(\mathcal{P})$ *hold, then*

$$\Gamma \vdash Q : \varphi \Rightarrow \Gamma \models Q : \varphi.$$

**Proof.** By induction on the derivation of $\Gamma \vdash Q : \varphi$.

*Case 1.* The last step applied is $(ax)$, i.e. $\Gamma, x : \varphi \vdash x : \varphi$. Then obviously $\Gamma, x : \varphi \models x : \varphi$, by Definitions 3.3 and 3.5.

*Case 2.* The last step applied is $(\to E)$, i.e. $\Gamma \vdash M : \tau \to \varphi, \Gamma \vdash N : \tau \Rightarrow \Gamma \vdash MN : \varphi$. Then by the induction hypothesis $\Gamma \models M : \tau \to \varphi$ and $\Gamma \models N : \tau$. Let $\rho \models \Gamma$, then $[\![M]\!]_\rho \in [\![\tau \to \varphi]\!] = [\![\tau]\!] \Rightarrow [\![\varphi]\!]$ and $[\![N]\!]_\rho \in [\![\tau]\!]$, which means that $[\![M]\!]_\rho [\![N]\!]_\rho \in [\![\varphi]\!]$. Since by Lemma 3.4(ii) $[\![MN]\!]_\rho \equiv [\![M]\!]_\rho [\![N]\!]_\rho$ it follows that $[\![MN]\!]_\rho \in [\![\varphi]\!]$.

*Case 3.* The last step applied is $(\to I)$, i.e. $\Gamma, x : \sigma \vdash M : \tau \Rightarrow \Gamma \vdash \lambda x.M : \sigma \to \tau$. By the induction hypothesis $\Gamma, x : \sigma \models M : \tau$. Let $\rho \models \Gamma$ and let $N \in [\![\sigma]\!]$. Then $\rho(x := N) \models \Gamma$ since $x \notin \mathit{Dom}\,\Gamma$ and $\rho(x := N) \models x : \sigma$ since $N \in [\![\sigma]\!]$. Therefore $\rho(x := N) \models M : \tau$, i.e. $[\![M]\!]_{\rho(x := N)} \in [\![\tau]\!]$. By Lemma 3.4(i) this means that $[\![M]\!]_{\rho(x:=x)}[x := N] \in [\![\tau]\!]$. According to Lemma 3.8 $N \in \mathcal{P}$, hence by applying $\mathit{SAT}(\mathcal{P})$ we get $(\lambda x.[\![M]\!]_{\rho(x:=x)})N \in [\![\tau]\!]$. Now by Lemma 3.4(iii) $\lambda x.[\![M]\!]_{\rho(x:=x)} \equiv [\![\lambda x.M]\!]_\rho$. Thus $[\![\lambda x.M]\!]_\rho N \in [\![\tau]\!]$. We conclude that $[\![\lambda x.M]\!]_\rho \in [\![\sigma]\!] \Rightarrow [\![\tau]\!]$ since $N \in [\![\sigma]\!]$ was arbitrary.

*Case 4.* The last step applied is $(\cap E)$, i.e. $\Gamma \vdash M : \sigma \cap \tau \Rightarrow \Gamma \vdash M : \sigma,\ \Gamma \vdash M : \tau$. By the induction hypothesis $\Gamma \models M : \sigma \cap \tau$. Let $\rho \models \Gamma$, then $[\![M]\!]_\rho \in [\![\sigma \cap \tau]\!] = [\![\sigma]\!] \cap [\![\tau]\!]$. Therefore $[\![M]\!]_\rho \in [\![\sigma]\!]$ and $[\![M]\!]_\rho \in [\![\tau]\!]$, i.e. $\Gamma \models M : \sigma$ and $\Gamma \models M : \tau$.

*Case 5.* The last step applied is $(\cap I)$, i.e. $\Gamma \vdash M : \sigma,\ \Gamma \vdash M : \tau \Rightarrow \Gamma \vdash M : \sigma \cap \tau$. Then by the induction hypothesis $\Gamma \models M : \sigma$ and $\Gamma \models M : \tau$. Let $\rho \models \Gamma$, then $[\![M]\!]_\rho \in [\![\sigma]\!]$ and $[\![M]\!]_\rho \in [\![\tau]\!]$. Therefore $[\![M]\!]_\rho \in [\![\sigma \cap \tau]\!]$, i.e. $\Gamma \models M : \sigma \cap \tau$.

*Case 6.* The last step applied is $(\leq)$, i.e. $\Gamma \vdash M : \sigma,\ \sigma \leq \tau \Rightarrow \Gamma \vdash M : \tau$. By the induction hypothesis $\Gamma \models M : \sigma$. Let $\rho \models \Gamma$, then $[\![M]\!]_\rho \in [\![\sigma]\!]$. According to Lemma 3.7 $[\![\sigma]\!] \subseteq [\![\tau]\!]$, so it follows that $[\![M]\!]_\rho \in [\![\tau]\!]$, i.e. $\Gamma \models M : \tau$. $\square$

An immediate consequence of soundness with respect to type interpretation is the following property.

**Proposition 3.11** *If $VAR(\mathcal{P})$, $SAT(\mathcal{P})$, and $CLO(\mathcal{P})$, then*

$$\Gamma \vdash M : \varphi \Rightarrow M \in \mathcal{P}.$$

**Proof.** Let $\Gamma \vdash M : \varphi$, then $\Gamma \models M : \varphi$ by Proposition 3.10. Let us take a valuation $\rho$ such that $\rho(y) \equiv y$ for all $y \in \mathtt{var}$. For every $(x : \sigma) \in \Gamma$ we have that $\rho \models x : \sigma$ since $x \in [\![\sigma]\!]$ by $VAR(\mathcal{P})$. Therefore $\rho \models \Gamma$ and consequently $\rho \models M : \varphi$, which means that $M \equiv [\![M]\!]_\rho \in [\![\varphi]\!]$. By applying Lemma 3.8 we get $M \in \mathcal{P}$. $\square$

**Proposition 3.12 (Soundness with respect to $\Omega$-type interpretation)** *If $VAR^\Omega(\mathcal{P})$, $SAT^\Omega(\mathcal{P})$, and $CLO^\Omega(\mathcal{P})$, then*

$$\Gamma \vdash^\Omega Q : \varphi \Rightarrow \Gamma \models^\Omega Q : \varphi.$$

**Proof.** The proof is along the lines of the proof of Proposition 3.10. Let us reconsider Case 3.

*Case 3.* The last step applied is $(\rightarrow I)$, i.e. $\Gamma, x : \sigma \vdash M : \tau \Rightarrow \Gamma \vdash \lambda x.M : \sigma \rightarrow \tau$. By the induction hypothesis $\Gamma, x : \sigma \models M : \tau$. Let $\rho \models \Gamma$ and let $N \in [\![\sigma]\!]^\Omega$. Then $\rho(x := N) \models \Gamma$ since $x \notin \mathit{Dom}\,\Gamma$ and $\rho(x := N) \models x : \sigma$ since $N \in [\![\sigma]\!]^\Omega$. Therefore $\rho(x := N) \models M : \tau$, i.e. $[\![M]\!]_{\rho(x := N)} \in [\![\tau]\!]^\Omega$. By Lemma 3.4(i) these means that $[\![M]\!]_{\rho(x:=x)}[x := N] \in [\![\tau]\!]^\Omega$. (Here we cannot apply $SAT(\mathcal{P})$, since $\sigma$ can be $\Omega$ so we cannot claim by Lemma 3.2 that $N \in \mathcal{P}$. This was the reason for introducing $SAT^\Omega(\mathcal{P})$.) Hence, by applying $SAT^\Omega(\mathcal{P})$ we get $(\lambda x.[\![M]\!]_{\rho(x:=x)})N \in [\![\tau]\!]^\Omega$. Now by Lemma 3.4(iii) $\lambda x.[\![M]\!]_{\rho(x:=x)} \equiv [\![\lambda x.M]\!]_\rho$. Thus $[\![\lambda x.M]\!]_\rho N \in [\![\tau]\!]^\Omega$. We conclude that $[\![\lambda x.M]\!]_\rho \in [\![\sigma]\!]^\Omega \Rightarrow [\![\tau]\!]^\Omega$ since $N \in [\![\sigma]\!]^\Omega$ was arbitrary.

What we have to show is that $[\![\lambda x.M]\!]_\rho \in [\![\sigma]\!]^\Omega \Rightarrow_\Omega [\![\tau]\!]^\Omega$. Hence it remains to prove that $[\![\lambda x.M]\!]_\rho \in \mathcal{P}$. Let us take $x \in [\![\sigma]\!]$ by $VAR(\mathcal{P})$. By repeating the above argument where $x \in [\![\sigma]\!]$ is taken instead of $N \in [\![\sigma]\!]$ we obtain

$[\![M]\!]_{\rho(x:=x)} \in [\![\tau]\!]^\Omega$, which means that $\lambda x.[\![M]\!]_{\rho(x:=x)} \in \mathcal{P}$ by $CLO^\Omega(\mathcal{P})$. According to Lemma 3.4(iii) $\lambda x.[\![M]\!]_{\rho(x:=x)} \equiv [\![\lambda x.M]\!]_\rho$, thus $[\![\lambda x.M]\!]_\rho \in \mathcal{P}$. □

An immediate consequence of soundness with respect to Ω-type interpretation is the following property.

**Proposition 3.13** *If $VAR^\Omega(\mathcal{P})$, $SAT^\Omega(\mathcal{P})$, and $CLO^\Omega(\mathcal{P})$, then*

$$(\forall \varphi \in \mathtt{type}^\Omega)\ \varphi \not\sim \Omega\ \wedge\ \Gamma \vdash^\Omega M : \varphi \Rightarrow M \in \mathcal{P}.$$

**Proof.** Let $\Gamma \vdash^\Omega M : \varphi$, then $\Gamma \models^\Omega M : \varphi$ by Proposition 3.12. Let us take such a valuation $\rho$ that $\rho(y) \equiv y$ for all $y \in \mathtt{var}$. For every $(x : \sigma) \in \Gamma$ we have that $\rho \models^\Omega x : \sigma$ since $x \in [\![\sigma]\!]^\Omega$ by $VAR^\Omega(\mathcal{P})$. Therefore $\rho \models^\Omega \Gamma$ and consequently $\rho \models^\Omega M : \varphi$, which means that $M \equiv [\![M]\!]_\rho \in [\![\varphi]\!]^\Omega$. Since $\varphi \not\sim \Omega$ according to Lemma 3.2 we have that $[\![\varphi]\!]^\Omega \subseteq \mathcal{P}$. Thus $M \in \mathcal{P}$. □

**Remark 3.14** Let us notice here that the required property, which states that a typeable term belongs to $\mathcal{P}$, is provided on the one hand in Proposition 3.11 by $VAR)(\mathcal{P})$ and $CLO(\mathcal{P})$ (Lemma 3.8) and on the other hand in Proposition 3.13 it is provided by the condition (Ω3) of Ω-type interpretation (Lemma 3.2).

We showed in Proposition 3.10, 3.11, 3.12, and 3.13 that the properties $VAR(\mathcal{P})$, $SAT(\mathcal{P})$, $CLO(\mathcal{P})$, $VAR^\Omega(\mathcal{P})$, $SAT^\Omega(\mathcal{P})$, and $CLO^\Omega(\mathcal{P})$ are sufficient to develop the reducibility method. Nevertheless in order to prove these properties one has to proceed by induction on the construction of the type $\varphi$, but then one needs stronger induction hypotheses which are easier to prove. These stronger conditions actually unify the conditions for saturated and $\mathcal{P}$-saturated sets which are considered in reducibility methods in Krivine [14], Barendregt [3], Gallier [9], and Koletsos and Stavrinos [13]. In the sequel we focus on the system $\lambda\cap^\Omega$, therefore we generalize $VAR^\Omega(\mathcal{P})$, $SAT^\Omega(\mathcal{P})$, and $CLO^\Omega(\mathcal{P})$.

**Definition 3.15** Let $\mathcal{P} \subseteq \Lambda$ be given. We say that:

($\mathcal{P}$-*VAR*) $\mathcal{X} \subseteq \Lambda$ satisfies the $\mathcal{P}$-variable property, notation $VAR(\mathcal{P}, \mathcal{X})$, if

$$(\forall x \in \mathtt{var})\ (\forall n \geq 0)\ (\forall M_1, \ldots, M_n \in \mathcal{P})\ \ xM_1 \ldots M_n \in \mathcal{X}.$$

($\mathcal{P}$-*SAT*) $\mathcal{X} \subseteq \Lambda$ is $\mathcal{P}$-saturated, notation $SAT(\mathcal{P}, \mathcal{X})$, if

$$(\forall M, N \in \Lambda)\ (\forall n \geq 0)\ (\forall M_1, \ldots, M_n \in \mathcal{P})$$
$$M[x := N]M_1 \ldots M_n \in \mathcal{X}\ \Rightarrow\ (\lambda x.M)NM_1 \ldots M_n \in \mathcal{X}.$$

($\mathcal{P}$-*CLO*) $\mathcal{X} \subseteq \Lambda$ is $\mathcal{P}$-closed, notation $CLO(\mathcal{P}, \mathcal{X})$, if

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

We show that the conditions in Definition 3.15 imply the corresponding conditions in Definition 3.6.

**Lemma 3.16** $\mathit{VAR}(\mathcal{P},\mathcal{P}) \;\Rightarrow\; (\forall\varphi \in \mathtt{type}^{\Omega})\ (\varphi \not\sim \Omega \to \tau)\mathit{VAR}(\mathcal{P}, [\![\varphi]\!]^{\Omega}_{(\mathcal{P})})$.

**Proof.** We prove the statement by induction on the construction of $\varphi$. Let us assume $\mathit{VAR}(\mathcal{P},\mathcal{P})$.

*Case* $\varphi \equiv \alpha$ is an atom. Since $[\![\alpha]\!]^{\Omega} = \mathcal{P}$, the statement holds by assumption.

*Case* $\varphi \equiv \sigma \to \tau$. Let $M_1, \dots, M_n \in \mathcal{P}$. We have to show that $xM_1 \dots M_n \in [\![\sigma \to \tau]\!]^{\Omega}$. First, $xM_1 \dots M_n \in \mathcal{P}$ by assumption. It remains to prove that $xM_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \Rightarrow [\![\tau]\!]^{\Omega}$. Take an arbitrary $M_{n+1} \in [\![\sigma]\!]^{\Omega}$. If $\sigma \not\sim \Omega$, then by Lemma 3.2 $M_{n+1} \in \mathcal{P}$, hence $xM_1 \dots M_n M_{n+1} \in [\![\tau]\!]^{\Omega}$ follows by the induction hypothesis. If $\sigma \sim \Omega$ and $\tau \sim \Omega$, then $[\![\sigma]\!]^{\Omega} \to [\![\tau]\!]^{\Omega} = \Lambda \to \Lambda$, so $xM_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \to [\![\tau]\!]^{\Omega}$, since $\Lambda \to \Lambda = \Lambda$.

*Case* $\varphi \equiv \sigma \cap \tau$. Let $M_1, \dots, M_n \in \mathcal{P}$. By the induction hypothesis $xM_1 \dots M_n \in [\![\sigma]\!]^{\Omega}$ and $xM_1 \dots M_n \in [\![\tau]\!]^{\Omega}$. Obviously, $xM_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \cap [\![\tau]\!]^{\Omega}$.

*Case* $\varphi \equiv \Omega$ is obvious, since $[\![\Omega]\!]^{\Omega} = \Lambda$.

In the other direction: $\mathit{VAR}(\mathcal{P},\mathcal{P})$ holds, since $[\![\alpha]\!] = \mathcal{P}$ for any atom type $\alpha$. □

An immediate consequence of Lemma 3.16 is the following statement.

**Corollary 3.17** $\mathit{VAR}(\mathcal{P},\mathcal{P}) \Rightarrow \mathit{VAR}^{\Omega}(\mathcal{P})$*, except for types* $\varphi \sim \Omega \to \tau$*.*

**Proof.** If $\mathit{VAR}(\mathcal{P},\mathcal{P})$ holds, then according to Lemma 3.16 $\mathit{VAR}(\mathcal{P}, [\![\varphi]\!]^{\Omega})$ holds for every $\varphi \in \mathtt{type}^{\Omega}$. Obviously, $\mathit{VAR}(\mathcal{P}, [\![\varphi]\!]^{\Omega})$ implies that $\mathtt{var} \subseteq [\![\varphi]\!]^{\Omega}$ for every $\Omega$-type $\varphi$. □

We proceed similarly for the conditions $\mathit{SAT}^{\Omega}(\mathcal{P})$ and $\mathit{SAT}(\mathcal{P},\mathcal{P})$.

**Lemma 3.18** $\mathit{SAT}(\mathcal{P},\mathcal{P}) \;\Leftrightarrow\; (\forall\varphi \in \mathtt{type}^{\Omega})(\varphi \not\sim \Omega \to \tau)\mathit{SAT}(\mathcal{P}, [\![\varphi]\!]^{\Omega})$.

**Proof.** By induction on the construction of $\varphi$. We proceed as in the previous lemma. Let us assume $\mathit{SAT}(\mathcal{P},\mathcal{P})$.

*Case* $\varphi \equiv \alpha \in \mathtt{atom}$. Since $[\![\alpha]\!]^{\Omega} = \mathcal{P}$, the property holds by assumption.

*Case* $\varphi \equiv \sigma \to \tau$. Let $M_1, \dots, M_n \in \mathcal{P}$. Suppose

$$M[x := N]M_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \Rightarrow_{\Omega} [\![\tau]\!]^{\Omega} = ([\![\sigma]\!]^{\Omega} \Rightarrow [\![\tau]\!]^{\Omega}) \cap \mathcal{P}.$$

Then $(\lambda x.M)NM_1 \dots M_n \in \mathcal{P}$, since $\mathit{SAT}(\mathcal{P},\mathcal{P})$ What remains to show is that $(\lambda x.M)NM_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \Rightarrow [\![\tau]\!]^{\Omega}$. In case that $\sigma \sim \Omega$ and $\tau \sim \Omega$ the property follows since $[\![\sigma]\!]^{\Omega} \Rightarrow [\![\tau]\!]^{\Omega} = \Lambda \to \Lambda = \Lambda$. Let $\sigma \not\sim \Omega$ and $M_{n+1} \in [\![\sigma]\!]^{\Omega}$, then $M[x := N]M_1 \dots M_n M_{n+1} \in [\![\tau]\!]^{\Omega}$. On the other hand by Lemma 3.2 $M_{n+1} \in \mathcal{P}$, so by the induction hypothesis $(\lambda x.M)NM_1 \dots M_n M_{n+1} \in [\![\tau]\!]^{\Omega}$. Since $M_{n+1}$ was arbitrary, we obtain $(\lambda x.M)NM_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \Rightarrow [\![\tau]\!]^{\Omega}$.

*Case* $\varphi \equiv \sigma \cap \tau$. Let $M, N, M_1, \dots, M_n \in \mathcal{P}$. Suppose

$$M[x := N]M_1 \dots M_n \in [\![\sigma]\!]^{\Omega} \cap [\![\tau]\!]^{\Omega}.$$

Then $M[x:=N]M_1\dots M_n \in [\![\sigma]\!]^\Omega$ and $M[x:=N]M_1\dots M_n \in [\![\tau]\!]^\Omega$. By the induction hypothesis $(\lambda x.M)NM_1\dots M_n \in [\![\sigma]\!]^\Omega$ and $(\lambda x.M)NM_1\dots M_n \in [\![\tau]\!]^\Omega$, therefore $(\lambda x.M)NM_1\dots M_n \in [\![\sigma]\!]^\Omega \cap [\![\tau]\!]^\Omega$.

*Case* $\varphi \equiv \Omega$ is straightforward since $[\![\Omega]\!]^\Omega = \Lambda$.

The other direction follows the argument of Lemma 3.16. □

**Corollary 3.19** *$SAT(\mathcal{P},\mathcal{P}) \Rightarrow SAT^\Omega(\mathcal{P})$, except for types $\varphi \sim \Omega \to \tau$.*

**Proof.** By Lemma 3.18 and by Definition 3.6 of $SAT^\Omega(\mathcal{P})$. □

**Lemma 3.20** *$CLO(\mathcal{P},\mathcal{P}) \Rightarrow CLO^\Omega(\mathcal{P})$, except for types $\varphi \sim \Omega$.*

**Proof.** Straightforward, since by Lemma 3.2 $[\![\varphi]\!]^\Omega \subseteq \mathcal{P}$ for all types $\varphi \not\sim \Omega$.□

Consequently, conditions $VAR(\mathcal{P},\mathcal{P})$, $SAT(\mathcal{P},\mathcal{P})$, and $CLO(\mathcal{P},\mathcal{P})$ are generalizations of $VAR^\Omega(\mathcal{P})$, $SAT^\Omega(\mathcal{P})$, and $CLO^\Omega(\mathcal{P})$, respectively. The following statement presents the general reducibility method which leads to uniform proofs of various reduction properties of the lambda terms typeable in $\lambda\cap^\Omega$ and will be presented in the sequel.

**Proposition 3.21 (Reducibility method for typeable terms)** *Let $VAR(\mathcal{P},\mathcal{P})$, $SAT(\mathcal{P},\mathcal{P})$, and $CLO^\Omega(\mathcal{P},\mathcal{P})$. Then*

$$(\forall\varphi \in \mathtt{type}^\Omega)\ \ \varphi \not\sim \Omega \ \ \wedge\ \ \varphi \not\sim \Omega \to \tau \ \ \wedge\ \ \Gamma \vdash^\Omega M : \varphi \Rightarrow M \in \mathcal{P}.$$

**Proof.** According to Proposition 3.13, Corollary 3.17, 3.19, and Lemma 3.20.□

**Definition 3.22** A set $\mathcal{P} \subseteq \Lambda$ is said to be invariant under abstraction if

$$M \in \mathcal{P} \ \Leftrightarrow\ \lambda x.M \in \mathcal{P}$$

Now we have all necessary conditions to establish a proof method for untyped lambda terms.

**Proposition 3.23 (Proof method for untyped terms)** *If $\mathcal{P}$ is invariant under abstraction, $VAR(\mathcal{P},\mathcal{P})$, and $SAT(\mathcal{P},\mathcal{P})$, then $\mathcal{P} = \Lambda$.*

**Proof.** For all $M \in \Lambda$ we have that $\lambda x.M$ is a weakly head normalizing term, i.e. $\lambda x.M \in \mathcal{W}$. It is easy to verify that there is a context $\Gamma$ such that $\Gamma \vdash^\Omega \lambda x.M : \Omega \to \Omega$. The conditions of Proposition 3.21 are satisfied since $CLO(\mathcal{P},\mathcal{P})$ is one of the implications in the Definition 3.22. Hence, according to Proposition 3.21 it follows that $\lambda x.M \in \mathcal{P}$. Therefore $M \in \mathcal{P}$, since $\mathcal{P}$ is invariant under abstraction. □

# 4 Application of the methods

In this section we show that the method given in Proposition 3.21 is applicable to the system $\lambda\cap^{\Omega}$, when $\mathcal{P}$ is:

4.1 $\mathcal{P} = \mathcal{C} = \{M \in \Lambda \mid \beta\text{-reduction is confluent on } M\}$;

4.2 $\mathcal{P} = \mathcal{S} = \{M \mid \text{every reduction of } M \text{ can be done in a standard way}\}$;

4.3 $\mathcal{P} = \mathcal{W} = \{M \mid M \text{ is weakly head normalizing}\}$.

Moreover we show that the proof method presented in Proposition 3.23 is suitable in Case 4.1 and 4.2.

### *4.1 Confluence of* $\twoheadrightarrow_\beta$ *in* $\Lambda$

The first proof of confluence of simply typed lambda calculus using the reducibility method is due to Statman [19] and Koletsos [12]. Applying the reducibility method we prove the confluence of $\beta$-reduction on lambda terms typeable in $\lambda\cap^{\Omega}$ by appropriate types. As a direct consequence we obtain the confluence of $\beta$-reduction on the set $\Lambda$ of all (untyped) lambda terms.

Let $\mathcal{C}$ be the set of all lambda terms on which $\beta$-reduction is confluent. We shall prove that $VAR(\mathcal{C},\mathcal{C})$, $SAT(\mathcal{C},\mathcal{C})$, and $CLO(\mathcal{C},\mathcal{C})$ hold. Then the confluence of $\beta$-reduction on lambda terms typeable in $\lambda\cap^{\Omega}$ by an appropriate type is a direct consequence of the method presented in the previous section in Proposition 3.21. For the sake of simplicity in this section we write $\rightarrow$ instead of $\rightarrow_\beta$ and $\twoheadrightarrow$ instead of $\twoheadrightarrow_\beta$.

**Definition 4.1** $\mathcal{C} = \{M \in \Lambda \mid M_1 \twoheadleftarrow M \twoheadrightarrow M_2 \Rightarrow (\exists M_3 \in \Lambda)\;\; M_1 \twoheadrightarrow M_3 \twoheadleftarrow M_2\}$.

**Lemma 4.2** $VAR(\mathcal{C},\mathcal{C})$.

**Proof.** Let $xM'_1 \dots M'_n \twoheadleftarrow xM_1 \dots M_n \twoheadrightarrow xM''_1 \dots M''_n$. The only possibility for the reductions is $M'_i \twoheadleftarrow M_i \twoheadrightarrow M''_i$ for $1 \le i \le n$. Since $M_i \in \mathcal{C}$, there is $M'''_i$, for each $i$, $1 \le i \le n$, such that $M'_i \twoheadrightarrow M'''_i \twoheadleftarrow M''_i$. But then

$$xM'_1 \dots M'_n \twoheadrightarrow xM'''_1 \dots M'''_n \twoheadleftarrow xM''_1 \dots M''_n.$$

$\square$

**Lemma 4.3** $SAT(\mathcal{C},\mathcal{C})$.

**Proof.** Let $M, N \in \Lambda$ and $M_1, \dots, M_n \in \mathcal{C}$ and $M[x := N]M_1 \dots M_n \in \mathcal{C}$. Let $P \equiv (\lambda x.M)NM_1 \dots M_n$ and suppose $R \twoheadleftarrow P \twoheadrightarrow S$. Depending on whether the head redex of $P$ is reduced we consider the following cases.

**Case** $(\lambda x.M')N'M'_1 \dots M'_n \twoheadleftarrow P \twoheadrightarrow (\lambda x.M'')N''M''_1 \dots M''_n$ with $M' \twoheadleftarrow M \twoheadrightarrow M''$, $N' \twoheadleftarrow N \twoheadrightarrow N''$, and $M'_i \twoheadleftarrow M_i \twoheadrightarrow M''_i$ for $1 \le i \le n$. Then $M' \twoheadrightarrow M''' \twoheadleftarrow M''$, $N' \twoheadrightarrow N''' \twoheadleftarrow N''$, and $M'_i \twoheadrightarrow M'''_i \twoheadleftarrow M''_i$ for $1 \le i \le n$, so $(\lambda x.M')N'M'_1 \dots M'_n \twoheadrightarrow (\lambda x.M''')N'''M'''_1 \dots M'''_n \twoheadleftarrow (\lambda x.M'')N''M''_1 \dots M''_n$.

**Case** $R \twoheadleftarrow M'[x := N']M'_1 \dots M'_n \twoheadleftarrow P \twoheadrightarrow M''[x := N'']M''_1 \dots M''_n \twoheadrightarrow S$ with $M' \twoheadleftarrow M \twoheadrightarrow M''$, $N' \twoheadleftarrow N \twoheadrightarrow N''$, and $M'_i \twoheadleftarrow M_i \twoheadrightarrow M''_i$, for $1 \le i \le n$. Then

$R \twoheadleftarrow M'[x:=N']M'_1 \ldots M'_n \twoheadleftarrow M[x:=N]M_1 \ldots M_n \twoheadrightarrow M''[x:=N'']M''_1 \ldots M''_n \twoheadrightarrow S$, so the result follows from $P \twoheadrightarrow M[x:=N]M_1 \ldots M_n \in \mathcal{C}$.

**Case** $R \twoheadleftarrow M'[x:=N']M'_1 \ldots M'_n \twoheadleftarrow P \twoheadrightarrow (\lambda x.M'')N''M''_1 \ldots M''_n$ with $M' \twoheadleftarrow M \twoheadrightarrow M''$, $N' \twoheadleftarrow N \twoheadrightarrow N''$, and $M'_i \twoheadleftarrow M_i \twoheadrightarrow M''_i$ for $1 \leq i \leq n$. Let $M' \twoheadrightarrow M''' \twoheadleftarrow M''$, $N' \twoheadrightarrow N''' \twoheadleftarrow N''$ and $M'_i \twoheadrightarrow M'''_i \twoheadleftarrow M''_i$ for $1 \leq i \leq n$. Then $R \twoheadleftarrow M'[x:=N']M'_1 \ldots M'_n \twoheadleftarrow M[x:=N]M_1 \ldots M_n \twoheadrightarrow M'''[x:=N''']M'''_1 \ldots M'''_n$ so from $M[x:=N]M_1 \ldots M_n \in \mathcal{C}$ there is $Z \in \Lambda$ such that $R \twoheadrightarrow Z \twoheadleftarrow M'''[x:=N''']M'''_1 \ldots M'''_n$. But then also

$$R \twoheadrightarrow Z \twoheadleftarrow M'''[x:=N''']M'''_1 \ldots M'''_n \twoheadleftarrow (\lambda x.M'')N''M''_1 \ldots M''_n.$$

□

**Lemma 4.4** (*CLO*$(\mathcal{C},\mathcal{C})$) $M \in \mathcal{C} \Rightarrow \lambda x.M \in \mathcal{C}$.

**Proof.** Let $M \in \mathcal{C}$. Assume $R \twoheadleftarrow \lambda x.M \twoheadrightarrow S$. Then $R \equiv \lambda x.R'$ and $S \equiv \lambda x.S'$ with $R' \twoheadleftarrow M \twoheadrightarrow S'$. Hence, there is $Z \in \Lambda$ such that $R' \twoheadrightarrow Z \twoheadleftarrow S'$. Thus $\lambda x.R' \twoheadrightarrow \lambda x.Z \twoheadleftarrow \lambda x.S'$. □

**Proposition 4.5** *Let $M \in \Lambda$ be given. If $\Gamma \vdash^{\Omega} M : \varphi$ for some context $\Gamma$ and $\varphi \not\sim \Omega$, $\varphi \not\sim \Omega \to \tau$, then $\beta$-reduction is confluent on $M$.*

**Proof.** By Proposition 3.21 and Lemma 4.2, 4.3, and 4.4. □

An important consequence of Proposition 4.5 is the confluence of $\beta$-reduction on the set $\Lambda$ of all (untyped) lambda terms. In order to prove that the conditions of Proposition 3.23 are fulfilled it remains to prove the inverse of Lemma 4.4.

**Lemma 4.6** *Let $M \in \Lambda$. Then:*

$$\lambda x.M \in \mathcal{C} \Rightarrow M \in \mathcal{C}.$$

**Proof.** Assume $\lambda x.M \in \mathcal{C}$. The only way to $\beta$-reduce $\lambda x.M$ is to $\beta$-reduce $M$. Hence, $M \in \mathcal{C}$. □

Notice that the previous property does not hold for $\beta\eta$-reduction.

**Corollary 4.7 (Confluence of $\beta$-reduction)** *If $M \in \Lambda$, then $M \in \mathcal{C}$.*

**Proof.** By Proposition 3.23 and Lemma 4.2, 4.3, 4.4, and 4.6. □

### *4.2 Standardization in $\Lambda$*

The property of lambda terms that each reduction can be decomposed into head reductions followed by internal reductions (these notions are mentioned in Section 2) is referred to as the *standardization* (Barendregt [2]). Here we prove standardization of terms typeable in $\lambda\cap^{\Omega}$ applying the techniques of the reducibility method. As a direct consequence we obtain the standardization of all (untyped) lambda terms.

Let $\mathcal{S}$ denote the set of all lambda terms that satisfy the standardization property. We prove $VAR(\mathcal{S},\mathcal{S})$, $SAT(\mathcal{S},\mathcal{S})$, and $CLO(\mathcal{S},\mathcal{S})$. In this section we write $\rightarrow$ instead of $\rightarrow_\beta$ and $\twoheadrightarrow$ instead of $\twoheadrightarrow_\beta$.

**Definition 4.8** $\mathcal{S} = \{M \in \Lambda \mid M \twoheadrightarrow Z \Rightarrow (\exists N \in \Lambda)\ M \twoheadrightarrow_h N \twoheadrightarrow_i Z\}$.

**Lemma 4.9** *$VAR(\mathcal{S},\mathcal{S})$.*

**Proof.** If $xM_1 \dots M_n \twoheadrightarrow Z$, then $xM_1 \dots M_n \twoheadrightarrow_i Z$ since the term has no head redexes. □

**Lemma 4.10** *$SAT(\mathcal{S},\mathcal{S})$.*

**Proof.** Let $M, N, M_1 \dots M_n \in \mathcal{S}$ and let $M[x := N]M_1 \dots M_n \in \mathcal{S}$. Suppose $P \equiv (\lambda x.M)NM_1 \dots M_n \twoheadrightarrow Z$.

**Case** $Z \equiv (\lambda x.M')N'M'_1 \dots M'_n$, $M \twoheadrightarrow M'$, $N \twoheadrightarrow N'$, $M_i \twoheadrightarrow M'_i$ for $1 \leq i \leq n$. Then the reduction is internal: $P \twoheadrightarrow_h P \twoheadrightarrow_i Z$.

**Case** $P \twoheadrightarrow (\lambda x.M')N'M'_1 \dots M'_n \rightarrow M'[x := N']M'_1 \dots M'_n \twoheadrightarrow Z$, $M \twoheadrightarrow M'$, $N \twoheadrightarrow N'$, and $M_i \twoheadrightarrow M'_i$ for $1 \leq i \leq n$. Then $M[x := N]M_1 \dots M_n \twoheadrightarrow$ $\twoheadrightarrow M'[x := N']M'_1 \dots M'_n \twoheadrightarrow Z$. Since $M[x := N]M_1 \dots M_n \in \mathcal{S}$ we have that $(\lambda x.M)NM_1 \dots M_n \rightarrow_h M[x := N]M_1 \dots M_n \twoheadrightarrow_h Z' \twoheadrightarrow_i Z$, which means that $(\lambda x.M)NM_1 \dots M_n \in \mathcal{S}$. □

**Lemma 4.11** **($CLO(\mathcal{S},\mathcal{S})$)** *$M \in \mathcal{S} \Rightarrow \lambda x.M \in \mathcal{S}$.*

**Proof.** Suppose $M \in \mathcal{S}$ and $\lambda x.M \twoheadrightarrow Z$. Then $Z \equiv \lambda x.M'$ with $M \twoheadrightarrow M'$, so $M \twoheadrightarrow_h N \twoheadrightarrow_i M'$ for some term $N$. But the head redex of $M$ is also a head redex of $\lambda x.M$ and vice versa, so $\lambda x.M \twoheadrightarrow_h \lambda x.N \twoheadrightarrow_i \lambda x.M'$, which means that $\lambda x.M \in \mathcal{S}$. □

**Proposition 4.12** *Let $M \in \Lambda$ be given. If $\Gamma \vdash^\Omega M : \varphi$ for some context $\Gamma$ and $\varphi \not\sim \Omega$, $\varphi \not\sim \Omega \rightarrow \tau$, then $M \in \mathcal{S}$.*

**Proof.** By Proposition 3.21 and Lemma 4.9, 4.10, and 4.11. □

An important consequence of Proposition 4.12 is the standardization property for all lambda terms. First, let us prove the inverse of Lemma 4.11 in order to establish the invariance of $\mathcal{S}$ under abstraction.

**Lemma 4.13** *Let $M \in \Lambda$. Then:*

$$\lambda x.M \in \mathcal{S} \Rightarrow M \in \mathcal{S}.$$

**Proof.** The only way to head reduce $\lambda x.M$ is to head reduce $M$. Hence, the standard reduction of $\lambda x.M$ is the standard reduction of $M$ as well, i.e. $M \in \mathcal{S}$. □

**Corollary 4.14 (Standardization in $\Lambda$)** *If $M \in \Lambda$, then $M \in \mathcal{S}$.*

**Proof.** By Proposition 3.23 and Lemma 4.9, 4.10, 4.11, and 4.13. □

### *4.3 Existence of weak head normal form in $\lambda\cap^\Omega$*

A term is a *weak head normal form* if it starts with an abstraction, or with a variable. A term is weakly head normalizing if it reduces to a weak head normal form. Let $\mathcal{W}$ denote the set of all lambda terms that have a weak head normal form. We shall prove that $VAR(\mathcal{W},\mathcal{W})$, $SAT(\mathcal{W},\mathcal{W})$, and $CLO(\mathcal{W},\mathcal{W})$ are satisfied. Then the existence of a weak head normal form of lambda terms typeable in $\lambda\cap^\Omega$ by appropriate types is a direct consequence of the method presented in the previous section in Proposition 3.21. For the sake of simplicity in this section we write $\to$ instead of $\to_\beta$ and $\twoheadrightarrow$ instead of $\twoheadrightarrow_\beta$.

The set $\mathcal{W}$ of weakly head normalizing terms is already defined in Section 2:

$$\mathcal{W} = \{M \in \Lambda \mid (\exists P, P_1, \dots, P_n \in \Lambda)\ M \underset{\beta}{\twoheadrightarrow} \lambda x.P \text{ or } M \underset{\beta}{\twoheadrightarrow} xP_1 \dots P_n\}.$$

It is easy to verify the required properties in this case since:

- $VAR(\mathcal{W},\mathcal{W})$ holds because all terms of the form $xM_1 \dots M_n$ are weak head normal forms;
- $SAT(\mathcal{W},\mathcal{W})$ holds because the set $\mathcal{W}$ of weakly head normalizing terms is closed under $\beta$-conversion;
- $CLO(\mathcal{W},\mathcal{W})$ holds because all terms of the form $\lambda x.M$ are weak head normal forms.

**Proposition 4.15** *Let $M \in \Lambda$ be given. If $\Gamma \vdash^\Omega M : \varphi$ for some context $\Gamma$ and $\varphi \not\sim \Omega$, $\varphi \not\sim \Omega \to \tau$, then $M$ has a weak head normal form.*

**Proof.** By Proposition 3.21 and the previous discussion. □

## 5 Discussion

*What are the limits of the methods?* In the reducibility method it suffices to consider terms typeable by types satisfying certain conditions in order to change the proposed set $\mathcal{P}$. In this way we can prove that terms typeable in $\lambda\cap^\Omega$ by certain types are head-normalizing and normalizing.

The intersection type system can be changed by changing the preorder on types. If the axiom $\sigma \to \Omega \le \Omega \to \Omega$ is replaced by the usual axiom $\Omega \le \Omega \to \Omega$, then the system obtained is not able to distinguish weakly head normalizing terms from unsolvable terms. In this system solvable terms are known to be typeable by types not equivalent to $\Omega$ (non-trivial types). Hence, in this system the method can be applied to sets containing all solvable terms.

The proof methodology for untyped lambda calculus presented here is suitable for sets of lambda terms $\mathcal{P} \supseteq \mathcal{W}$ that are invariant under abstraction ($M \in \mathcal{P} \Leftrightarrow \lambda x.M \in \mathcal{P}$). Obviously $\mathcal{P} = \mathcal{W}$ cannot be the case, since this would lead to the contradiction $\Lambda = \mathcal{W}$. This is prevented by the invariance under abstraction, which $\mathcal{W}$ does not satisfy.

*Contributions of the presented methods?* The reducibility method presented here is an abstract method which unifies different known formulations discussed in Introduction. It derives necessary conditions which a set $\mathcal{P} \subseteq \Lambda$ has to fulfill in order to be a candidate for the application of the method. The proof methodology for untyped lambda calculus presented as a consequence of reducibility applied to $\lambda\cap^{\Omega}$ seams to be new up to our knowledge.

*Possible applications of the methods?* The reducibility method for $\lambda\cap$ is not completely developed here. One has to define a condition which will generalize condition $SAT(\mathcal{P})$. This can be done similarly to $SAT(\mathcal{P}, \mathcal{P})$. Hence, the method obtained in this way will provide a uniform way to prove various well-known reduction properties of terms typeable in $\lambda\cap$ such as existence of $(\beta\eta)$ normal form, uniqueness of $(\beta\eta)$ normal form, termination of the leftmost-outermost reduction, strong normalization, and others. The method can be applied to the simply typed lambda calculus as well. The applicability of this method to other type systems can be one of the areas of further investigation.

Finiteness of developments in the set $\Lambda$ of all untyped lambda terms can be proved along the lines of proofs presented in Section 4.

## Acknowledgement

The authors thank Viktor Kunčak for participating in the initial stages of this work. The authors are grateful to Mariangiola Dezani-Ciancaglini and Pierre Lescanne for valuable suggestions and remarks. Detailed comments of anonymous referees lead to improvements both in correctness and presentation.

## References

[1] Barendregt, H., M. Coppo and M. Dezani-Ciancaglini, *A filter lambda model and the completeness of type assignment*, Journal of Symbolic Logic **48** (1983), pp. 931–940 (1984).

[2] Barendregt, H. P., "The Lambda Calculus: its Syntax and Semantics," North-Holland, Amsterdam, 1984, revised edition.

[3] Barendregt, H. P., *Lambda calculi with types*, in: S. Abramsky, D. Gabbay and T. Maibaum, editors, *Handbook of Logic in Computer Science*, Volume B, Oxford University Press, Oxford, 1992 pp. 117–309.

[4] Coppo, M. and M. Dezani-Ciancaglini, *A new type-assignment for lambda terms*, Archiv für Mathematische Logik **19** (1978), pp. 139–156.

[5] Coppo, M. and M. Dezani-Ciancaglini, *An extension of the basic functionality theory for the $\lambda$-calculus*, Notre Dame Journal of Formal Logic **21** (1980), pp. 685–693.

[6] Dezani-Ciancaglini, M. and S. Ghilezan, *A lambda model characterizing computational behaviours of terms*, in: Y. Toyama, editor, *Proceedings of the*

*International Workshop Rewriting in Proof and Computation, RPC'01*, 2001, pp. 100–119.

[7] Dezani-Ciancaglini, M., F. Honsell and Y. Motohama, *Compositional characterization of λ-terms using intersection types*, in: M. Nielsen and B. Rovan, editors, *Mathematical Foundations of Computer Science 2000*, Lecture Notes in Computer Science **1893** (2000), pp. 304–314.

[8] Dougherty, D. and P. Lescanne, *Reductions, intersection types, and explicit substitution*, in: S. Abramsky, editor, *Typed Lambda Calculi and Application, 5th International Conference, TLCA 2001*, Lecture Notes in Computer Science **2044** (2001), pp. 121–135.

[9] Gallier, J., *Typing untyped λ-terms, or reducibility strikes again!*, Annals of Pure and Applied Logic **91** (1998), pp. 231–270.

[10] Ghilezan, S., *Strong normalization and typability with intersection types*, Notre Dame Journal of Formal Logic **37** (1996), pp. 44–52.

[11] Girard, J.-Y., *Une extension de l'interprétation de Gödel à l'analyse, et son application à l'elimination des coupures dans l'analyse et la théorie des types*, in: J. Fenstad, editor, *2nd Scandinavian Logic Symposium*, North-Holland, Amsterdam (1971), pp. 63–92.

[12] Koletsos, G., *Church-Rosser theorem for typed functionals*, Journal of Symbolic Logic **50** (1985), pp. 782–790.

[13] Koletsos, G. and G. Stavrinos, *The structure of reducibility proofs*, in: P. Kolaitis and G. Koletsos, editors, *Proceedings of the Second Panhellenic Logic Symposium*, 1999, pp. 138–144.

[14] Krivine, J.-L., "Lambda-calcul types et modèles," Masson, Paris, 1990.

[15] Mitchell, J. C., "Foundation for Programmimg Languages," MIT Press, Boston.

[16] Mitchell, J. C., *Type systems for programming languages*, in: J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Volume B, Elsevier, Amsterdam, 1990 pp. 415–431.

[17] Pottinger, G., *A type assignment for the strongly normalizable λ-terms*, in: J. Seldin and J. Hindley, editors, *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, Academic Press, London, 1980 pp. 561–577.

[18] Sallé, P., *Une extension de la thorie des types en lambda -calcul*, in: G. Ausiello and C. Böhm, editors, *Fifth International Conference on Automata, Languages and Programming*, Lecture Notes in Computer Science **62** (1978), pp. 398–410.

[19] Statman, R., *Logical relations and the typed λ-calculus*, Information and Control **65** (1985), pp. 85–97.

[20] Tait, W. W., *Intensional interpretations of functionals of finite type I*, Journal of Symbolic Logic **32** (1967), pp. 198–212.

[21] Tait, W. W., *A realizability interpretation of the theory of species*, in: *Logic Colloquium*, Lecture Notes in Mathematics **453** (1975), pp. 240–251.