Characterisation of Strongly Normalising $\lambda\mu$-Terms

Steffen van Bakel
Imperial College
London, UK
svb@doc.ic.ac.uk

Franco Barbanera
Università di Catania
Catania, Italy
barba@dmi.unict.it

Ugo de'Liguoro
Università di Torino
Torino, Italy
deliguoro@di.unito.it

A characterisation of strongly normalising terms of the $\lambda\mu$-calculus is provided by means of a typeing system with intersection and product types. The presence of the latter ones (and of the type ω) enables us to represent the particular notion of continuation used in the literature for defining the semantics of the $\lambda\mu$-calculus. This allows to lift to $\lambda\mu$-terms the well-known characterisation property for λ-terms possessed by the intersection types. An interpretation of Parigot's propositional logical system into ours will then provide an alternative proof of strong normalisation for terms typeable in that system.

Introduction

Parigot's $\lambda\mu$-calculus is an extension of the λ-calculus which was introduced in [7] to compute with classical proofs. One of the very first results has been proved in [8], and it states that all $\lambda\mu$-terms that correspond to proofs of second order natural deduction are strongly normalizing. In spite of this motivation, the $\lambda\mu$-calculus is type free. As a consequence there exist more terms than proofs, and among them are also perfectly meaningful ones that do not correspond to any proof. Indeed properties of pure $\lambda\mu$-terms have extensively studied (see e.g. [11, 5, 10]).

The basic idea to turn non-constructive proofs into algorithms is to add a form of continuation by means of names and μ-abstraction to capture control. On the other hand continuations introduce a great deal of complexity to the calculus semantics, which motivates a more abstract approach. Moving from Streicher and Reus denotational semantics of $\lambda\mu$ in [12], we have introduced an intersection type assignment system inducing a filter model in [2]. This is essentially a logical description of the domain theoretic model in [12], with the advantage of providing a formal tool to reason about the meaning of terms. At the end of that work we conjectured that an appropriate subsystem of ours would be able to type exactly all strongly normalizing terms, much as it is the case with Pottinger's theorem in [9] for the pure λ-calculus, where the Coppo-Dezani-Venneri intersection type system was used (for an accurate reconstruction of the proof see [1]; to establish the same result saturated sets are used in [6] chapter 4, and in the survey [3], where a characterisation in terms of a system with subtyping is given).

However in the case of $\lambda\mu$-calculus a straightforward extension of Pottinger's theorem does not hold, at least with respect to the system in [2]. This is due to the fact that in the model a continuation is an infinite tuple of terms, which is typed in the system by (finite intersections of) types $\kappa = \delta_1 \times \cdots \delta_k \times \omega$ for some $k > 0$, where the leading $\delta_1, \ldots, \delta_k$ encode the information about the first k terms in the tuple, while the ending ω represents the lack of information about the remaining infinite part. To solve the problem we first restrict types to those having ω only as the final part of a product type; then we suitably modify the standard interpretation of intersection types adapting Tait's computability argument, in such a way that the semantics of κ is the set of all finite tuples (which we call *stacks*) $\vec{L}$ of strongly normalizing terms that begin with k terms $L_1 \ldots, L_k$ belonging to the interpretations of $\delta_1, \ldots, \delta_k$.

To appear in EPTCS.

We then obtain Parigot's theorem in [8] (for the propositional fragment) by interpreting ordinary types into intersection types and proving that the translation preserves derivability from Parigot's system to ours.

1 The $\lambda\mu$-calculus

Definition 1.1 (TERM SYNTAX [7]) The sets Trm of *terms* and Cmd of *commands* are defined inductively by the following grammar (where $x \in \mathit{Var}$, a set of *term variables*, and $\alpha \in \mathit{Name}$, a set of *names*, that are both denumerable):

$$\begin{array}{rcll} M,N & ::= & x \mid \lambda x.M \mid MN \mid \mu\alpha.Q & \text{(terms)} \\ Q & ::= & [\alpha]M & \text{(commands)} \end{array}$$

Definition 1.2 (SUBSTITUTION [7]) Substitution takes two forms:

term substitution:	$M[N/x]$	(N is substituted for x in M, avoiding capture)
structural substitution:	$T[\alpha \Leftarrow L]$	(every subterm $[\alpha]N$ of M is replaced by $[\alpha]NL$)

where $M,N,L \in \mathsf{Trm}$, $Q \in \mathsf{Cmd}$ and $T \in \mathsf{Trm} \cup \mathsf{Cmd}$. More precisely, $T[\alpha \Leftarrow L]$ is defined by:

$$([\alpha]M)[\alpha \Leftarrow L] \triangleq [\alpha](M[\alpha \Leftarrow L])L$$

whereas in all the other cases it is defined by:

$$\begin{array}{rcl@{\qquad}rcl} x[\alpha \Leftarrow L] & \triangleq & x & & & \\ (\lambda x.M)[\alpha \Leftarrow L] & \triangleq & \lambda x.M[\alpha \Leftarrow L] & (MN)[\alpha \Leftarrow L] & \triangleq & (M[\alpha \Leftarrow L])(N[\alpha \Leftarrow L]) \\ (\mu\beta.Q)[\alpha \Leftarrow L] & \triangleq & \mu\beta.Q[\alpha \Leftarrow L] & ([\beta]M)[\alpha \Leftarrow L] & \triangleq & [\beta]M[\alpha \Leftarrow L] \end{array}$$

Definition 1.3 (REDUCTION [7]) The reduction relation $M \to_\mu N$, where $M,N \in \mathsf{Trm}$, is defined as the compatible closure of the following rules:

$$\begin{array}{rrcl} (\beta): & (\lambda x.M)N & \to & M[N/x] \\ (\mu): & (\mu\beta.Q)N & \to & \mu\beta.Q[\beta \Leftarrow N] \end{array}$$

2 Characterisation of Strong Normalisation

In this section, we will show that we can characterise strong normalisation for pure $\lambda\mu$ terms completely through a notion of intersection typeing.

2.1 The type system

Our characterization can be carried out by means of a purposely tailored version of the type system deviced in [2]. For what concerns the base types, a single base type ν sufficies here for our aims.

Definition 2.1 (TYPES) The sets $\mathcal{T}_D$ of *term types* and $\mathcal{T}_C$ of *continuation-stack types* are defined inductively by the following grammars, where ν is a type constant:

$$\begin{array}{rrcl} \mathcal{T}_D: & \delta & ::= & \nu \mid \omega\to\nu \mid \kappa\to\nu \mid \delta\wedge\delta \\ \mathcal{T}_C: & \kappa & ::= & \delta\times\omega \mid \delta\times\kappa \mid \kappa\wedge\kappa \end{array}$$

We define the set $\mathcal{T}$ of *types* as $\mathcal{T} = \mathcal{T}_D \cup \mathcal{T}_C$.

We shall use σ, τ, ρ, ... to denote generic elements of $\mathcal{T}$.

Notice how a relevant feature of our system, as it is usual also in the strongly normalizing typeings for the λ-calculus [1], is the absence of ω as a proper type (and then of its corresponding typeing rule.) The type ω can occur instead inside a type in order to represent the (unspecified) last part of a continuation stack.

Definition 2.2 The relations $\leq_D$ and $\leq_C$ are the least preorders over $\mathcal{T}_D$ and $\mathcal{T}_C$ respectively, such that:

$$\frac{}{\sigma\wedge\tau \leq_A \sigma} \qquad \frac{}{\sigma\wedge\tau \leq_A \tau} \qquad \frac{}{\nu =_D \omega\to\nu} \qquad \frac{}{\delta_1\times\delta_2\times\omega \leq_C \delta_1\times\omega}$$

$$\frac{}{(\delta_1\times\omega)\wedge(\delta_2\times\kappa) \leq_C (\delta_1\wedge\delta_2)\times\kappa} \qquad \frac{}{(\delta_1\times\kappa_1)\wedge(\delta_2\times\kappa_2) \leq_C (\delta_1\wedge\delta_2)\times(\kappa_1\wedge\kappa_2)}\ (\kappa_1,\kappa_2 \not\equiv \omega)$$

$$\frac{\delta_1 \leq_D \delta_2}{\delta_1\times\omega \leq_C \delta_2\times\omega} \qquad \frac{\delta_1 \leq_D \delta_2 \quad \kappa_1 \leq_C \kappa_2}{\delta_1\times\kappa_1 \leq_C \delta_2\times\kappa_2} \qquad \frac{\sigma \leq_A \tau_1 \quad \sigma \leq_A \tau_2}{\sigma \leq_A \tau_1\wedge\tau_2} \qquad \frac{\kappa_2 \leq_C \kappa_1}{\kappa_1\to\nu \leq_D \kappa_2\to\nu}$$

where $A = D, C$ and, as usual, we define $=_A \triangleq \leq_A \cap \geq_A$.

For convenience of notation, in the following the subscripts D and C on $\leq$ are normally omitted.

The preorders in Definition 2.2 are a restriction to $\mathcal{T}$ of the preorders in [2]. Notice that, in that system, the inequality $\delta_1\times\delta_2\times\omega \leq \delta_1\times\omega$ is derivable: $\delta_1\times\delta_2\times\omega \leq \delta_1\times\omega\times\omega = \delta_1\times\omega$, since in [2] we had $\omega =_C \omega\times\omega$. Here $\omega \notin \mathcal{T}_D$ so that $\delta_1\times\omega\times\omega \notin \mathcal{T}_C$, and therefore the inequality above has to be postulated.

The notions of basis (variable context), denoted by $\Gamma, \Gamma' \ldots$, and name context, denoted by $\Delta, \Delta' \ldots$, are defined in the standard way.
The relation $\leq$ is naturally extended to bases and name contexts as follows:
$\Gamma' \leq \Gamma$ *if and only if* $x : \delta \in \Gamma \Rightarrow [x : \delta' \in \Gamma' \ \&\ \delta' \leq \delta]$. Similarly for name contexts.

Definition 2.3 (TYPEING SYSTEM) We define the notion of *typeing* for pure $\lambda\mu$-terms (Trm) through the following natural deduction system:

$$(Ax):\ \frac{}{\Gamma, x{:}\delta \vdash x{:}\delta \mid \Delta} \qquad (\mu):\ \frac{\Gamma \vdash M{:}\kappa'\to\nu \mid \alpha{:}\kappa, \beta{:}\kappa', \Delta}{\Gamma \vdash \mu\alpha.[\beta]M{:}\kappa\to\nu \mid \beta{:}\kappa', \Delta} \qquad \frac{\Gamma \vdash M{:}\kappa\to\nu \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \kappa\to\nu \mid \Delta}$$

$$(abs):\ \frac{\Gamma, x{:}\delta \vdash M{:}\kappa\to\nu \mid \Delta}{\Gamma \vdash \lambda x.M{:}\delta\times\kappa\to\nu \mid \Delta} \qquad (app):\ \frac{\Gamma \vdash M{:}\delta\times\kappa\to\nu \mid \Delta \quad \Gamma \vdash N{:}\delta \mid \Delta}{\Gamma \vdash MN{:}\kappa\to\nu \mid \Delta}$$

$$(\leq):\ \frac{\Gamma \vdash M{:}\delta \mid \Delta}{\Gamma \vdash M{:}\delta' \mid \Delta}\ (\delta \leq \delta') \qquad (\wedge):\ \frac{\Gamma \vdash M{:}\delta \mid \Delta \quad \Gamma \vdash M{:}\delta' \mid \Delta}{\Gamma \vdash M{:}\delta\wedge\delta' \mid \Delta}$$

where κ in rule (app) [1] and is either a type in $\mathcal{T}_C$ or ω.

As usual, we write $\Gamma \vdash M{:}\delta \mid \Delta$ whenever there exists a derivation built using these rules that has this judgement in the bottom line.

Lemma 2.4 The following Weakening and Strenghtening rules are admissible

$$(W):\ \frac{\Gamma \vdash M{:}\delta \mid \Delta}{\Gamma' \vdash M{:}\delta \mid \Delta'}\ (\Gamma' \leq \Gamma, \Delta' \leq \Delta) \qquad (S):\ \frac{\Gamma \vdash M{:}\delta \mid \Delta}{\Gamma' \vdash M{:}\delta \mid \Delta'}\ \left(\begin{array}{l}\Gamma' = \{x{:}\delta \mid x \in fv(M)\}\\ \Delta' = \{\alpha{:}\kappa \mid \alpha \in fn(M)\}\end{array}\right)$$

[1] (app) and (abs) are used to dub the rules concerning λ-abstraction and application, instead of the usual $(\to I)$ and $(\to E)$, since there is actually no introduction/elimination of the $\to$ type constructor.

Definition 2.5 Given two bases Γ_1 and Γ_2, we define the basis $\Gamma_1 \wedge \Gamma_2$ as follows:

$$\begin{array}{rcl} \Gamma_1 \wedge \Gamma_2 & = & \{x : \delta_1 \wedge \delta_2 \mid x{:}\delta_1 \in \Gamma_1 \ \&\ x{:}\delta_2 \in \Gamma_2\} \cup \\ & & \{x{:}\delta \mid x{:}\delta \in \Gamma_1 \ \&\ x \notin dom(\Gamma_2)\} \cup \{x{:}\delta \mid x{:}\delta \in \Gamma_2 \ \&\ x \notin dom(\Gamma_1)\} \end{array}$$

The name context $\Delta_1 \wedge \Delta_2$ is defined in the same way.

Trivially $dom(\Gamma_1 \wedge \Gamma_2) = dom(\Gamma_1) \cup dom(\Gamma_2)$ and $dom(\Delta_1 \wedge \Delta_2) = dom(\Delta_1) \cup dom(\Delta_2)$. Moreover, it is quite easy to show that $\Gamma_1 \wedge \Gamma_2 \leq \Gamma_i$ and $\Delta_1 \wedge \Delta_2 \leq \Delta_i$ for $i = 1,2$. This then leads immediately to:

Corollary 2.6 *If* $\Gamma_1 \vdash M{:}\delta \mid \Delta_1$ *then for any* Γ_2, Δ_2: $\Gamma_1 \wedge \Gamma_2 \vdash M{:}\delta \mid \Delta_1 \wedge \Delta_2$.

The following substitution results can be proved along the lines followed for the similar ones in [2]:

Lemma 2.7 (SUBSTITUTION LEMMA)

1. $\Gamma \vdash M[N/x]{:}\delta \mid \Delta$ *if and only if there exists* δ' *such that* $\Gamma \vdash N{:}\delta' \mid \Delta$ *and* $\Gamma, x{:}\delta' \vdash M{:}\delta \mid \Delta$.
2. $\Gamma \vdash M[\alpha \Leftarrow L] : \delta \mid \alpha{:}\kappa, \Delta$ *if and only if there exists* δ' *such that* $\Gamma \vdash L{:}\delta' \mid \Delta$, *and* $\Gamma \vdash M{:}\delta \mid \alpha{:}\delta' \times \kappa, \Delta$.

2.2 Typeability implies Strong Normalisation

In this subsection we will show that, as can be expected of a well-defined notion of type assignment that does not type recursion, all typeable terms are strongly normalising. Such a property does not hold for the system in [2] where, in fact, by means of types not allowed in the present system, it is possible to type in a non-trivial way the fixed-point constructor, as shown by the following derivation:

Example 2.8

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f{:}\omega\times\omega\to\nu, x{:}\omega \vdash f : \omega\times\omega\to\nu \mid}\,(Ax) \quad \dfrac{}{f{:}\omega\times\omega\to\nu, x{:}\omega \vdash xx : \omega \mid}\,(\omega)}{f{:}\omega\times\omega\to\nu, x{:}\omega \vdash f(xx) : \omega\to\nu \mid}\,(app)}{f{:}\omega\times\omega\to\nu \vdash \lambda x.f(xx) : \omega\times\omega\to\nu \mid}\,(abs) \quad \dfrac{}{f{:}\omega\times\omega\to\nu \vdash \lambda x.f(xx) : \omega \mid}\,(\omega)}{f{:}\omega\times\omega\to\nu \vdash (\lambda x.f(xx))(\lambda x.f(xx)) : \omega\to\nu \mid}\,(app)}{\vdash \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) : (\omega\times\omega\to\nu)\times\omega\to\nu \mid}\,(abs)$$

Notice that this term does not have a normal form, so is not strongly normalisable.

Definition 2.9 Let $\vec{L} \equiv L_1 : \cdots : L_k$ be a stack of terms; by Trm^* we denote the set of all finite (possibly empty) stacks of terms; we write ϵ for the empty stack. If $M \in \mathsf{Trm}$ and $\vec{L} \equiv L_1 : \cdots : L_k$ then $M{:}\vec{L} \equiv M{:}L_1 : \cdots : L_k \in \mathsf{Trm}^*$, while we define $M(P{:}\vec{L}) \triangleq MP\vec{L}$, so $M\vec{L} \equiv ML_1 \cdots L_k$.
The set $\mathcal{SN}$ of *strongly normalisable* terms is defined as usual as the set of all terms M such that no infinite reduction out of M exists; we use $\mathcal{SN}(M)$ for $M \in \mathcal{SN}$, and $\mathcal{SN}^*$ for the set of finite stacks of terms in $\mathcal{SN}$.

Definition 2.10 (TYPE INTERPRETATION)

1. We define a map $\|\cdot\| : (\mathcal{T}_D \to \wp(\mathsf{Trm})) + (\mathcal{T}_C \to \wp(\mathsf{Trm}^*)$, interpreting term types and continuation-

stack types as, respectively, sets of terms and sets of stacks of terms, as follows:

$$\begin{array}{rcl} \|\omega{\to}\nu\| \triangleq \|\nu\| & \triangleq & \mathcal{SN} \\ \|\kappa{\to}\nu\| & \triangleq & \{M \in \mathsf{Trm} \mid \forall \vec{L} \in \|\kappa\|.\ M\vec{L} \in \|\nu\|\} \\ \|\delta{\times}\omega\| & \triangleq & \{N{:}\vec{L} \mid N \in \|\delta\|, \vec{L} \in \mathcal{SN}^*\} \\ \|\delta{\times}\kappa\| & \triangleq & \{N{:}\vec{L} \mid N \in \|\delta\|, \vec{L} \in \|\kappa\|\} \\ \|\sigma{\wedge}\tau\| & \triangleq & \|\sigma\| \cap \|\tau\| \end{array}$$

2. We define $|\cdot| : \mathcal{T}_C \to \mathbb{N}$ as follows:

$$|\delta{\times}\omega| \triangleq 1 \qquad |\delta{\times}\kappa| \triangleq 1 + |\kappa| \qquad |\kappa_1{\wedge}\kappa_2| \triangleq max\{|\kappa_1|, |\kappa_2|\}$$

By the above interpretation, the elements of $\|\delta_1 \times \cdots \times \delta_n \times \omega\|$ are stacks of strongly normalisable terms having an arbitrary length greater than or equal to n.
It is easy to check that, given $\kappa \in \mathcal{T}_C$, the $|\cdot|$ function returns the minimal length of the stacks in $\|\kappa\|$.

The following technical result is not difficult to show:

Proposition 2.11 *1. If $\mathcal{SN}(x\vec{M})$ and $\mathcal{SN}(\vec{N})$, then $\mathcal{SN}(x\vec{M}\vec{N})$.*
2. If $\mathcal{SN}(M[N/x]\vec{P})$ and $\mathcal{SN}(N)$, then $\mathcal{SN}((\lambda x.M)N\vec{P})$.
3. If $\mathcal{SN}(M)$, then $\mathcal{SN}(\mu\alpha.[\beta]M)$.
4. If $\mathcal{SN}(\mu\alpha.[\beta]M[\alpha \Leftarrow \vec{N}]\vec{L})$ and $\mathcal{SN}(\vec{N})$, then $\mathcal{SN}((\mu\alpha.[\beta]M)\vec{N}\vec{L})$.
5. If $\mathcal{SN}(\mu\alpha.[\alpha]M[\alpha \Leftarrow \vec{N}]\vec{N}\vec{L})$, then $\mathcal{SN}((\mu\alpha.[\alpha]M)\vec{N}\vec{L})$.

Lemma 2.12 For any $\delta \in \mathcal{T}_D$ and $\kappa \in \mathcal{T}_C$:
1. $\|\delta\| \subseteq \mathcal{SN}$ and $\|\kappa\| \subseteq \mathcal{SN}^*$.
2. $x\vec{N} \in \mathcal{SN} \Rightarrow x\vec{N} \in \|\delta\|$, for all δ.
3. $\vec{x} = x_1 : \ldots : x_n \in \|\kappa\|$, for all n and κ such that $n \geq |\kappa|$.

Proof. By simultaneous induction on the structure of types, using Definition 2.10. □

The following result follows immediately from 2.12(2):

Corollary 2.13 For any $x \in Var$ and any $\delta \in \mathcal{T}_D$: $x \in \|\delta\|$.

The following lemma shows that our type interpretation is closed for the type inclusion relation.

Lemma 2.14 For all $\sigma, \tau \in \mathcal{T}$: $\sigma \leq \tau \Rightarrow \|\sigma\| \subseteq \|\tau\|$.

Proof. By induction on the definition of $\leq$. We show one relevant case.

$$\begin{array}{lcl} (\delta_1{\times}\omega){\wedge}(\delta_2{\times}\kappa) \leq (\delta_1{\wedge}\delta_2){\times}\kappa) : \ \|(\delta_1{\times}\omega){\wedge}(\delta_2{\times}\kappa)\| & = & \\ \quad \{M{:}\vec{L} \mid M \in \|\delta_1\|, \vec{L} \in \mathcal{SN}^*\} \cap \{M{:}\vec{L} \mid M \in \|\delta_2\|, \vec{L} \in \|\kappa\|\} & = & (\|\kappa\| \subseteq \mathcal{SN}^* \text{ by } 2.12(1)) \\ \quad \{M{:}\vec{L} \mid M \in \|\delta_1\| \cap \|\delta_2\|, \vec{L} \in \|\kappa\|\} & = & \\ \quad \{M{:}\vec{L} \mid M \in \|\delta_1{\wedge}\delta_2\|, \vec{L} \in \|\kappa\|\} & = & \|(\delta_1{\wedge}\delta_2){\times}\kappa\| \end{array}$$

□

For $\vec{L} = L_1 : \cdots : L_k \in \mathsf{Trm}^*$, we introduce the notation:

$$M[\alpha \Leftarrow \vec{L}] \quad \triangleq \quad M[\alpha \Leftarrow L_1][\alpha \Leftarrow L_2] \cdots [\alpha \Leftarrow L_n]$$

In particular, we set $M[\alpha \Leftarrow \epsilon] = M$. Notice that, by Barendregt's convention, α is not free in any L_i, and

$$[\alpha]M[\alpha \Leftarrow L_1 : \cdots : L_n] \quad \equiv \quad [\alpha]M[\alpha \Leftarrow L_1][\alpha \Leftarrow L_2] \cdots [\alpha \Leftarrow L_n] \quad \equiv \quad [\alpha]ML_1L_2 \cdots L_n$$

Our type interpretation is closed for expansion, both for the logical as for the structural reduction, under proviso.

Lemma 2.15 1. *If* $M[N/x]\vec{P} \in \|\delta\|$ *and* $N \in \|\delta'\|$, *then* $(\lambda x.M)N\vec{P} \in \|\sigma\|$.
2. *If* $\mu\alpha.[\beta]M[\alpha \Leftarrow \vec{N}]\vec{P} \in \|\delta\|$ *and* $\vec{N} \in \|\kappa\|$, *then* $(\mu\alpha.[\beta]M)\vec{N}\vec{P} \in \|\delta\|$.
3. *If* $\mu\alpha.[\alpha]M[\alpha \Leftarrow \vec{N}]\vec{N}\vec{P} \in \|\delta\|$, *then* $(\mu\alpha.[\alpha]M)\vec{N}\vec{P} \in \|\delta\|$.

Proof. By induction on the structure of types, using 2.10, 2.11 and 2.12. □

Definition 2.16 Let $\xi : (\mathit{Var} \to \mathsf{Trm}) + (\mathit{Name} \to \mathsf{Trm}^*)$ be a partial mapping (*substitution*).

1. We define
$$M_\xi = M\,[\xi\, x_1 / x_1, \ldots, \xi\, x_h / x_h, \alpha_1 \Leftarrow \xi\, \alpha_1, \ldots, \alpha_k \Leftarrow \xi\, \alpha_k]$$
By Barendregt's convention we can assume that all x_i are distinct, as are all α_i, and none of these occur in the image of ξ (so it is irrelevant whether all substitutions take place simultaneously or not.)
2. We define $\xi[x \mapsto N]$ and $\xi[\alpha \Leftarrow \vec{L}]$ by, respectively,
$$\xi[x \mapsto M]\, y = \begin{cases} M & \text{if } y = x \\ \xi\, y & \text{otherwise} \end{cases} \qquad \xi[\alpha \Leftarrow \vec{L}]\, \beta = \begin{cases} \vec{L} & \text{if } \alpha = \beta \\ \xi\, \beta & \text{otherwise} \end{cases}$$
3. We will say that ξ *extends* Γ and Δ, if, for all $x{:}\delta \in \Gamma$, we have $\xi\, x \in \|\delta\|$, and, for all $\alpha{:}\kappa \in \Delta$, we have $\xi\, \alpha \in \|\kappa\|$, and no x_i or α_i occurs in the image of ξ.

Lemma 2.17 (REPLACEMENT LEMMA) *Let* ξ *be a substitution that extends* Γ *and* Δ, *then* $\Gamma \vdash M{:}\delta \mid \Delta$ *implies* $M_\xi \in \|\delta\|$.

Proof. By induction on the structure of derivations. We show just the subcase $\alpha \neq \beta$ for rule (μ). In such a case $M = \mu\alpha.[\beta]M'$, and $\delta = \kappa \to \nu$.

Then $\Delta = \beta{:}\kappa', \Delta'$, and $\Gamma \vdash M' : \kappa' \to \nu \mid \alpha{:}\kappa, \beta{:}\kappa', \Delta$. Assume $\vec{L} \in \|\kappa\|$, then $\xi[\alpha \Leftarrow \vec{L}]$ extends Γ and $\alpha{:}\kappa, \beta{:}\kappa', \Delta'$. Then, by induction, $M'_{\xi[\alpha := \vec{L}]} \in \|\kappa' \to \nu\|$. Now let $\vec{Q} \in \|\kappa'\|$, then $M'_{\xi[\alpha := \vec{L}]}\vec{Q} \in \|\nu\|$; then also $(M'\vec{Q})_{\xi[\alpha := \vec{L}]} \in \|\nu\|$. Then by Definition 2.10, $\mathcal{SN}((M'\vec{Q})_{\xi[\alpha := \vec{L}]})$, and by Lemma 2.11(3) also $\mathcal{SN}(\mu\alpha.[\beta](M'\vec{Q})_{\xi[\alpha := \vec{L}]})$, so $\mu\alpha.[\beta](M'\vec{Q})_\xi[\alpha := \vec{L}] \in \|\nu\|$. Then, by Lemma 2.15(2), $(\mu\alpha.[\beta](M'\vec{Q})_\xi)\vec{L} \in \|\nu\|$. Notice that $[\beta]M'_\xi\vec{Q} = [\beta]M'_\xi[\beta \Leftarrow \vec{Q}]$; since $\xi\beta = \vec{Q}$, we can then infer that $[\beta]M'_\xi\vec{Q} = [\beta]M'_\xi$, so $(\mu\alpha.[\beta]M')_\xi\vec{L} \in \|\nu\|$. But then $(\mu\alpha.[\beta]M')_\xi \in \|\kappa \to \nu\|$. □

We now come to the main result of this section, stating that all terms typeable in our system are strongly normalisable.

Theorem 2.18 (TYPEABLE TERMS ARE $\mathcal{SN}$) *If* $\Gamma \vdash M{:}\delta \mid \Delta$ *for some* Γ, Δ *and* δ, *then* $M \in \mathcal{SN}$.

Proof. Let ξ be a substitution such that

$$\begin{array}{lcll} \xi x & = & x & \text{for } x \in dom(\Gamma) \\ \xi\alpha & = & \vec{y}_\alpha & \text{for } \alpha \in dom(\Delta) \end{array}$$

where the length of a vector $\vec{y}_\alpha$ is $|\kappa|$ if $\alpha : \kappa \in \Delta$.
By Lemma 2.12, ξ extends Γ and Δ. Hence, by the Replacement Lemma 2.17, $M_\xi \in \|\delta\|$, and then $M_\xi \in \mathcal{SN}$ by Lemma 2.12(1). Now

$$\begin{array}{rcl} M_\xi & \equiv & M[x_1/x_1,\ldots,x_n/x_n,\alpha_1 \Leftarrow \vec{y}_{\alpha_1},\ldots,\alpha_m \Leftarrow \vec{y}_{\alpha_m}] \\ & \equiv & M[\alpha_1 \Leftarrow \vec{y}_{\alpha_1},\ldots,\alpha_m \Leftarrow \vec{y}_{\alpha_m}] \end{array}$$

Then, by Lemma 2.11, also $(\mu\alpha_1.[\beta_1]\cdots\mu\alpha_m.[\beta_m]M)\vec{y}_{\alpha_1}\cdots\vec{y}_{\alpha_m} \in \mathcal{SN}$, so also $M \in \mathcal{SN}$. □

2.3 Strongly Normalising Terms are Typeable

In this section we will show the counterpart of the previous result, namely that all strongly normalisable terms are typeable in our intersection system.

First we describe the shape of the terms in normal form of the $\lambda\mu$-calculus.

Definition 2.19 (NORMAL FORMS) The set $\mathcal{N} \subseteq \mathsf{Trm}$ of *normal forms* is defined by the grammar:

$$N \quad ::= \quad xN_1\cdots N_k \mid \lambda x.N \mid \mu\alpha.[\beta]N$$

It is straightforward to verify that the terms in $\mathcal{N}$ are precisely the irreducible ones.

We can show that all terms in $\mathcal{N}$ are typeable.

Lemma 2.20 *If $N \in \mathcal{N}$ then there exist Γ,Δ and a type $\kappa\to\nu$ such that $\Gamma \vdash N : \kappa\to\nu \mid \Delta$.*

Proof. By induction on the definition of $\mathcal{N}$. We show one relevant case.

$(N \equiv xN_1\ldots N_k)$: Since $N_1,\ldots,N_k \in \mathcal{N}$, by induction we have that, for all $i \le k$ there exist Γ_i,Δ_i and δ_i such that $\Gamma_i \vdash N_i : \delta_i \mid \Delta_i$ (the structure of each δ_i plays no role in this part). Take

$$\Gamma = \Gamma_1 \wedge \cdots \wedge \Gamma_k \wedge \{x : (\delta_1 \times \cdots \times \delta_k \times \omega) \to \nu\}, \text{and } \Delta = \Delta_1 \wedge \cdots \wedge \Delta_k.$$

Then, by Lemma 2.4, $\Gamma \vdash N_i : \delta_i \mid \Delta$ for all $i \le n$, and $\Gamma \vdash x : (\delta_1 \times \cdots \times \delta_k \times \omega) \to \nu \mid \Delta$. By repeated application of (app) we get $\Gamma \vdash xN_1\cdots N_k : \omega \to \nu \mid \Delta$. □

We will now show that typeing is closed for expansion with respect to both logical and structural reductions, under proviso.

Lemma 2.21 *1. If $\Gamma \vdash M[N/x] : \delta \mid \Delta$ and $\Gamma \vdash N : \delta' \mid \Delta$ then $\Gamma \vdash (\lambda x.M)N : \delta \mid \Delta$.*
2. If $\Gamma \vdash \mu\alpha.[\beta]M[\alpha \Leftarrow N] : \delta \mid \Delta$ and $\Gamma \vdash N : \delta' \mid \Delta$ then $\Gamma \vdash (\mu\alpha.[\beta]M)N : \delta \mid \Delta$.

Proof. The proof of 1. can be carried on along the same lines of similar results for the intersection type systems for the λ-calculus. For 2. we consider just a relevant case, i.e. when $\alpha \in fn([\beta]M)$ with $\alpha = \beta$:

Then $([\beta]M)[\alpha \Leftarrow N] \equiv ([\alpha]M)[\alpha \Leftarrow N] \equiv [\alpha](M[\alpha \Leftarrow N])N$; we can assume, without loss of generality, that $\delta = (\kappa_1 \to \nu) \wedge \cdots \wedge (\kappa_n \to \nu)$, and that for all $i \le n$ there are subderivations ending in:

$$\dfrac{\dfrac{\Gamma \vdash M[\alpha \Leftarrow N] : \delta_i \times \kappa_i \to \nu \mid \alpha{:}\kappa_i, \Delta \qquad \Gamma \vdash N : \delta_i \mid \Delta}{\Gamma \vdash (M[\alpha \Leftarrow N])N : \kappa_i \to \nu \mid \alpha{:}\kappa_i, \Delta}\,(app)}{\Gamma \vdash \mu\alpha.[\alpha](M[\alpha \Leftarrow N])N : \kappa_i \to \nu \mid \Delta}\,(\mu)$$

Then by Lemma 2.7(2), there exists δ'_i such that $\Gamma \vdash N:\delta'_i \mid \Delta$, and $\Gamma \vdash M:\delta_i\times\kappa_i\to\nu \mid \alpha:\delta'_i\times\kappa_i,\Delta$; so we can build the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash M:\delta_i\times\kappa_i\to\nu \mid \alpha:\delta'_i\times\kappa_i,\Delta}{\Gamma \vdash M:\delta_i\wedge\delta'_i\times\kappa_i\to\nu \mid \alpha:\delta'_i\times\kappa_i,\Delta}(\leq)}{\Gamma \vdash M:\delta_i\wedge\delta'_i\times\kappa_i\to\nu \mid \alpha:\delta_i\wedge\delta'_i\times\kappa_i,\Delta}(W)}{\Gamma \vdash \mu\alpha.[\alpha]M:\delta_i\wedge\delta'_i\times\kappa_i\to\nu \mid \Delta}(\mu) \qquad \dfrac{\Gamma \vdash N:\delta_i \mid \Delta \quad \Gamma \vdash N:\delta'_i \mid \Delta}{\Gamma \vdash N:\delta_i\wedge\delta'_i \mid \Delta}(\wedge)}{\Gamma \vdash (\mu\alpha.[\alpha]M)N:\kappa_i\to\nu \mid \Delta}(app)$$

We derive $\Gamma \vdash (\mu\alpha.[\alpha]M)N:\delta \mid \Delta$ by rule $(\wedge)$. □

Lemma 2.22 (SUBJECT EXPANSION) *If $M \to N$ by contracting either a β-redex $(\lambda x.P)Q$ or a μ-redex $(\mu\alpha.[\beta]P)Q$ and $\Gamma \vdash Q:\delta' \mid \Delta$ for some δ', then $\Gamma \vdash N:\delta \mid \Delta$ implies $\Gamma \vdash M:\delta \mid \Delta$.*

Proof. By induction over the structure of derivations, using Lemma 2.21. □

Theorem 2.23 (TYPEABILITY OF $\mathcal{SN}$-TERMS)
For all $M \in \mathcal{SN}$ there exist Γ and Δ and a type δ such that $\Gamma \vdash M:\delta \mid \Delta$.

Proof. Sketch: For $M \in \mathcal{SN}$, let $n(M)$ be the sum of the lengths of all the reduction paths out of M. The proof is by induction over this measure, using Lemma 2.20 for the base case and Lemmas 2.6 and 2.22 for the induction one. □

In the following section it will be shown that Parigot's logical system [7] can be interpreted in ours. Such an interpretation will enable us to get an alternative proof of strong normalisation for terms typeable in that system.

3 Interpretation of Parigot's Logical System

We use a version of Parigot's logical system which is equivalent to the original one if just terms and not also commands are typed. This implies that the rule for $\bot$ does not need to be taken into account. The system below does not includes rules $\forall I$ and $\forall E$, which however have no effect w.r.t. the subject in Parigot's first-order type-assignement system. We call simply typed $\lambda\mu$-calculus the propositional fragment of Parigot's original system.

Definition 3.1 (PARIGOT'S SYMPLY TYPED $\lambda\mu$-CALCULUS)
The set **LF** of *Logical Formulas* is defined by

$$A, B ::= \varphi \mid A\to B$$

Judgments have the shape $\Gamma \vdash_{\mathrm{P}} M : A \mid \Delta$ where $M \in \mathsf{Trm}$ and the basis Γ and the name context Δ associate formulas to, respectively, term variables and names.
Rules:

$$\dfrac{}{\Gamma, x : A \vdash_{\mathrm{P}} x : A \mid \Delta}(Ax)$$

$$\frac{\Gamma, x:A \vdash_{\mathrm{P}} M:B \mid \Delta}{\Gamma \vdash_{\mathrm{P}} \lambda x.M : A{\to}B \mid \Delta}\ (\to I) \qquad \frac{\Gamma \vdash_{\mathrm{P}} M : A{\to}B \mid \Delta \quad \Gamma \vdash_{\mathrm{P}} N : A \mid \Delta}{\Gamma \vdash_{\mathrm{P}} MN : B \mid \Delta}\ (\to E)$$

$$\frac{\Gamma \vdash_{\mathrm{P}} M : A \mid \alpha : A, \Delta}{\Gamma \vdash_{\mathrm{P}} \mu\alpha.[\alpha]M : A \mid \Delta}\ (\mu_1) \qquad \frac{\Gamma \vdash_{\mathrm{P}} M : B \mid \alpha : A, \beta : B, \Delta}{\Gamma \vdash_{\mathrm{P}} \mu\alpha[\beta]M : A \mid \beta : B, \Delta}\ (\mu_2)$$

As usual we write $\Gamma \vdash_{\mathrm{P}} M : A \mid \Delta$ to denote that such a judgment is derivable in Parigot's logical system.

We can interpret formulas into intersection types as follows.

Definition 3.2 The translation functions $(\cdot)^D : \mathbf{LF}{\to}\mathcal{T}_D$ and $(\cdot)^C : \mathbf{LF}{\to}\mathcal{T}_C$ are defined by:

$$\varphi^C \triangleq \nu \times \omega \qquad (A{\to}B)^C \triangleq (A^C{\to}\nu) \times B^C \qquad A^D \triangleq A^C{\to}\nu$$

It is straightforward to show that the above translations are well defined.

Theorem 3.3 *Let* $\Gamma^D = \{x : A^D \mid x : A \in \Gamma\}$ *and* $\Delta^C = \{\alpha : A^C \mid \alpha : A \in \Delta\}$.
If $\Gamma \vdash_{\mathrm{P}} M : A \mid \Delta$ *then* $\Gamma^D \vdash M : A^D \mid \Delta^C$.

Proof. Sketch: There is a one-to-one correspondence between the rules in the two systems; hence it suffices to show that rules are preserved when translating formulas into types. □

Strong normalisation of typeable terms in Parigot's system (first proved in [7]) now follows as a corollary of our characterization result.

Corollary 3.4 (STRONG NORMALISABILITY OF PARIGOT'S SIMPLY TYPED $\lambda\mu$-CALCULUS)
If $\Gamma \vdash_{\mathrm{P}} M : A \mid \Delta$ *then* $M \in \mathcal{SN}$.

Proof. By Theorem 3.3, if $\Gamma \vdash_{\mathrm{P}} M : A \mid \Delta$ then $\Gamma^D \vdash M : A^D \mid \Delta^C$ is derivable in the intersection type system. Hence $M \in \mathcal{SN}$ by Theorem 2.18. □

Conclusion

We have defined an intersection type system which characterizes strongly normalizing $\lambda\mu$-terms, extending to the pure $\lambda\mu$-calculus Pottinger's theorem for the λ-calculus.

We provide a translation of first order types of Parigot's system into intersection types of the restricted system proposed in this paper and prove that derivability is preserved. We are confident that the present result, established for Parigot's propositional system, extends to the full full first-order type-assignement system, so that to obtain an alternative proof of Parigot's strong normalisation theorem.

As we have observed in [2] our intersection-type assignment system can be adapted to de Groote's variant of the $\lambda\mu$-calculus (see e.g. [4]), and we think also to Saurin's $\Lambda\mu$ [11], that satisfy stronger properties than Parigot's original calculus, such as Böhm theorem. We leave to further work the question whether the present characterisation result extends to those cases.

References

[1] S. van Bakel (1992): *Complete restrictions of the Intersection Type Discipline*. *Theoretical Computer Science* 102(1), pp. 135–163.

[2] S. van Bakel, F. Barbanera & U. de' Liguoro (2011): *A Filter Model for $\lambda\mu$*. In Luke Ong, editor: *Proc. of TLCA'11, ARCoSS/LNCS* 6690, pp. 213–228.

[3] Silvia Ghilezan (1996): *Strong Normalization and Typability with Intersection Types*. *Notre Dame Journal of Formal Logic* 37(1), pp. 44–52.

[4] Philippe de Groote (1994): *A CPS-Translation of the Lambda-μ-Calculus*. In: *CAAP, Lecture Notes in Computer Science* 787, pp. 85–99.

[5] H. Herbelin & A. Saurin (2010): *$\lambda\mu$-calculus and $\Lambda\mu$-calculus: a Capital Difference*. Manuscript.

[6] J.-L Krivine (1993): *Lambda calculus, types and models*. Ellis Horwood.

[7] M. Parigot (1992): *An algorithmic interpretation of classical natural deduction*. In: *Proceedings of 3rd International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'92), Lecture Notes in Computer Science* 624, Springer Verlag, pp. 190–201.

[8] M. Parigot (1997): *Proofs of Strong Normalisation for Second Order Classical Natural Deduction*. *Journal of Symbolic Logic* 62(4), pp. 1461–1479.

[9] G. Pottinger (1980): *A Type Assignment for the Strongly Normalizable λ-terms*. In J.P. Seldin & J.R. Hindley, editors: *To H. B. Curry, Essays in combinatory logic, lambda-calculus and formalism*, Academic press, New York, pp. 561–577.

[10] A. Saurin (2010): *Standardization and Böhm Trees for Lambda-μ Calculus*. In: *Functional and Logic Programming, 10th International Symposium, FLOPS 2010, Sendai, Japan, April 19-21, 2010. Proceedings, Lecture Notes in Computer Science* 6009, pp. 134–149.

[11] Alexis Saurin (2008): *On the Relations between the Syntactic Theories of $\lambda\mu$-Calculi*. In: *CSL, Lecture Notes in Computer Science* 5213.

[12] T. Streicher & B. Reus (1998): *Classical Logic, Continuation Semantics and Abstract Machines*. *J. Funct. Program.* 8(6), pp. 543–572.