Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 398 (2008) 82–94
Theoretical Computer Science
www.elsevier.com/locate/tcs

The heart of intersection type assignment: Normalisation proofs revisited

Steffen van Bakel[1]

Department of Computing, Imperial College London, 180 Queen's Gate London SW7 2BZ, UK

To Mariangiola, Mario, and Simona

Abstract

This paper gives a new proof for the approximation theorem and the characterisation of normalisability using intersection types for a system with ω and a $\leq$-relation that is contra-variant over arrow types. The technique applied is to define reduction on derivations and to show a strong normalisation result for this reduction. From this result, the characterisation of strong normalisation and the approximation result will follow easily; the latter, in its turn, will lead to the characterisation of (head) normalisability.

Keywords: Intersection types; Cut-elimination; Strong normalisation; Approximants

0. Introduction

The main result in this paper will be to show that all the famous characterisation properties for the Intersection Type Discipline [7] are all in fact the consequences of a single result: cut-elimination is terminating. This result was already shown to hold for the *strict* system in [3]; the contribution of this paper is to show these results for a notion of type assignment that is closed for η-reduction, i.e. has an inclusion relation on types that is contra-variant on arrow types.

The Intersection Type Discipline as presented in [8] (a more enhanced system was presented in [7]; for an overview of the various existing systems, see [2]), is an extension of Curry's system [9], that consists mainly of allowing for term variables (and terms) to have more than one type, or an empty type. Intersection types are constructed by adding, next to the type constructor '$\to$' of Curry's system, the type constructor '$\cap$' and the type constant 'ω'.

This slight generalisation causes a great change in complexity; in fact, now type assignment is closed for β-equality: $M =_\beta N \Rightarrow (B \vdash M:\sigma \iff B \vdash N:\sigma)$ and (head / strong) normalisation can be characterised by assignable types:

$$\begin{array}{rcl} M \text{ has a head-normal form} & \iff & B \vdash M:\sigma \ \&\ \sigma \neq \omega \\ M \text{ has a normal form} & \iff & B \vdash M:\sigma \ \&\ \omega \text{ does not occur in } B, \sigma \\ M \text{ is strongly normalisable} & \iff & B \vdash M:\sigma, \text{ where } \omega \text{ is not used at all} \end{array}$$

E-mail address: svb@doc.ic.ac.uk.
[1] On sabbatical leave at Inria Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis, France.

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2008.01.020

(see, for example, [7,1,2]). These properties immediately show that type assignment (even in the system that does not contain ω, see [1]) is undecidable.

As in [13,6], the set of terms can be extended by adding the term constant $\bot$. Adding also the reduction rules $\bot N \to_{\beta\bot} \bot$, and $\lambda x.\bot \to_{\beta\bot} \bot$ to the notion of reduction gives rise to the notion of *approximate normal forms* that are in essence finite rooted segments of Böhm-trees. It is well known that interpreting a term by the set of approximants that can be associated to it gives a model for the Lambda Calculus. From the Approximation Theorem, i.e. the observation that there exists a very precise relation between types assignable to a term M and those assignable to its approximants, $\mathcal{A}(M)$ (see [11,1,2], and Definition 24), it is clear that the set of intersection types assignable to a term can be used to define a model for the Lambda Calculus (see [7,1,2]).

Of the above-mentioned results, all but the first will be proved again in this paper; in fact, we will show that these can all be obtained from one more fundamental result, being the strong normalisation of cut-elimination.

In previous papers, the Approximation Theorem and Strong Normalisation Theorem were proved independently (see, respectively, [2] and [1]), though both using the same technique of Computability Predicates [12,10]. In this paper, we will show that both are special cases of strong normalisation of cut-elimination, using a variant of the technique developed in [5], that has also found its application in other fields [4]. We will define a notion of reduction on derivations in '$\vdash$' that generalises cut-elimination, and prove that this kind of reduction is strongly normalisable. It might seem surprising, but this result does not come easy at all. The reason for this is that, unlike for ordinary systems of type assignment, for the intersection system there is a significant difference between derivation reduction and ordinary reduction (see the beginning of Section 2); unlike 'normal' typed or type assignment system, in '$\vdash$' not every term redex occurs with types in a derivation. Moreover, especially the use of a relation '$\leq$' on types, together with a derivation rule ($\leq$), greatly disturbs the smoothness of proofs (see again Section 2).

From this strong normalisation result for derivation reduction, the Approximation Theorem and Strong Normalisation Theorem follow easily. The first of these implies the Head-Normalisation Theorem and (indirectly) the Normalisation Theorem, as was already demonstrated in [2].

The kind of intersection type assignment considered in this paper is that of [2], i.e. the essential intersection type assignment system, a restricted version of the BCD-system of [7], that is equally powerful in terms of typeability and expressiveness. The major feature of this restricted system is, compared to the BCD-system, a restricted version of the derivation rules and the use of strict types (first introduced in [1]).

In [3] similar results were shown for the *strict* intersection type assignment system. This differs from the one considered here in that the $\leq$ relation on types used there is not contra-variant over arrow types, but only allows for the selection of one of the types in an intersection. The contribution of this paper is to generalise that result to the *essential* intersection type assignment system, a notion of type assignment that is also closed for η-reduction.

As shown in [2], the essential system is the nucleus of the BCD-system, in the sense that $B \vdash_{\mathrm{BCD}} M:\sigma \iff \exists B' \sim B, \sigma' \sim \sigma\, [B \vdash M:\sigma]$. This implies that all normalisation results for that system are also consequences of the main result of this paper. It will be feasible to give a direct proof for the BCD system as well, albeit that the notion of reduction on derivations will be slightly different, as then also $(\cap I)-(\cap E)$ cuts will have to contract. Although in the BCD-system the $\leq$-relation is more complex, the derivation rule ($\leq$) is a normal rule, and extending the proof that the computability predicate is closed for ($\leq$) will be easy. It feels safe to conjecture that the main result of this paper (strong normalisation of derivation reduction) will hold in all other intersection type systems with a contra-variant $\leq$-relation on (non-recursive) types.

The outline of this paper is as follows. In Section 1, we will recall the definition of the essential type assignment system of [2], together with some of its main properties. In Section 2, a notion of reduction on derivations in '$\vdash$' is defined, for which we will show a strong normalisation result in Section 3. In Section 4 we will focus on the head normalisation and approximation results, and show that they are consequences of the result of Section 3. We will finish this paper in Section 5 by extending the result of Section 3 to the characterisations of normalisation and termination.

The technique applied in this paper is similar to that of [3], but for the subtle differences caused by the presence of $\leq$, like the first part of Definition 12, Lemma 18, and Theorem 21. Since the systems differ, of course all properties had to be checked again; only those that do not depend on which notion is used – but are of use here – are quoted without proof.

We assume the reader to be familiar with the λ-calculus [6].

1. Intersection type assignment

In this section, the essential type assignment system of [2] is presented, a restricted version of the system presented in [7], together with some of its properties. The major feature of this restricted system is, compared to the BCD-system, a restricted version of the derivation rules and the use of strict types. It also forms a slight extension of the strict type assignment system that was presented in [1]; the main difference is that the strict system is not closed for η-reduction, whereas the essential system is.

Definition 1. (1) Let Φ be a countable (infinite) set of type variables, ranged over by φ. $\mathcal{T}_s$, the set of *strict types*, ranged over by $\phi, \psi, \ldots$, and $\mathcal{T}$, the set of *strict intersection types*, ranged over by $\sigma, \tau, \ldots$, are defined by:

$$\begin{array}{rcl} \phi, \psi & ::= & \varphi \mid (\sigma \to \phi) \\ \sigma, \tau & ::= & (\phi_1 \cap \cdots \cap \phi_n), \ (n \geq 0) \end{array}$$

(2) A *statement* is an expression of the form $M : \sigma$, with $M \in \Lambda$ (the set of lambda terms), and $\sigma \in \mathcal{T}$. M is the *subject* and σ the *predicate* of $M : \sigma$.
(3) A *basis* is a set of statements with only distinct variables as subjects.
(4) For a collection of bases $B_1, \ldots, B_n$, the basis $\bigcap\{B_1, \ldots, B_n\}$ is defined by: $x{:}\phi_1 \cap \cdots \cap \phi_m \in \bigcap\{B_1, \ldots, B_n\}$ if and only if $\{x{:}\phi_1, \ldots, x{:}\phi_m\}$ is the set of all statements about x that occurs in $B_1 \cup \cdots \cup B_n$.

Notice that $\mathcal{T}_s$ is a proper subset of $\mathcal{T}$. Often $B, x{:}\sigma$ will be written for the basis $\bigcap\{B, \{x{:}\sigma\}\}$, when x does not occur in B.

We will write $\underline{n}$ for the set $\{1, \ldots, n\}$, and $\cap_{\underline{n}}\phi_i$ for $\phi_1 \cap \cdots \cap \phi_n$. We define ω as the empty intersection: if $n = 0$, then $\cap_{\underline{n}}\phi_i \equiv \omega$, so ω does not occur in an intersection subtype.

Notice that intersection type schemes (so also ω) occur in strict types only as subtypes at the *left*-hand side of an arrow type scheme. Unless stated otherwise, if $\cap_{\underline{n}}\phi_i$ is used to denote a type, then all ϕ_i $(i \in \underline{n})$ are assumed to be strict.

Definition 2 (*Relations on Types*). (1) The relation $\leq$ is defined as the least pre-order (i.e. reflexive and transitive relation) on $\mathcal{T}$ such that:

$$\begin{array}{lll} \forall i \in \underline{n}\ [\cap_{\underline{n}}\phi_i \ \leq \ \phi_i], & & (n \geq 1) \\ \forall i \in \underline{n}\ [\sigma \leq \phi_i] \ \Rightarrow \ \sigma \leq \cap_{\underline{n}}\phi_i, & & (n \geq 0) \\ \rho \leq \sigma \ \& \ \phi \leq \psi \ \Rightarrow \ \sigma \to \phi \leq \rho \to \psi & : & \end{array}$$

(2) The equivalence relation $\sim$ on types is defined by: $\sigma \sim \tau \iff \sigma \leq \tau \leq \sigma$.
(3) We write $B \leq B'$ if and only if for every $x{:}\sigma' \in B'$ there is an $x{:}\sigma \in B$ such that $\sigma \leq \sigma'$, and $B \sim B' \iff B \leq B' \leq B$.

Notice that $\mathcal{T}$ may be considered modulo $\sim$; then $\leq$ becomes a partial order. In this paper, however, in order to get a strong relation between the structure of types and derivations, types will not be considered modulo $\sim$.

Moreover, since intersections are not allowed to appear on the right-hand side of arrows, $\sigma \to (\tau \cap \rho)$ is not a type; therefore, clause $(\sigma \to \tau) \cap (\sigma \to \rho) \leq \sigma \to \tau \cap \rho$ is not part of the above definition.

For the relation $\leq$, the following properties hold:

Lemma 3. (1) $\varphi \leq \sigma \iff \sigma \equiv \varphi$. *So* $\{\sigma \mid \sigma \sim \varphi\} = \{\varphi\}$.
(2) $\omega \leq \sigma \iff \sigma \equiv \omega$. *So* $\{\sigma \mid \sigma \sim \omega\} = \{\omega\}$.
(3) $\sigma \to \phi \leq \rho \in \mathcal{T}_s \iff \exists \alpha \in \mathcal{T}, \psi \in \mathcal{T}_s\ [\rho \equiv \alpha \to \psi \ \& \ \alpha \leq \sigma \ \& \ \phi \leq \psi\ ]$.
(4) $\cap_{\underline{n}}\phi_i \leq \tau \in \mathcal{T}_s \ \Rightarrow \ \exists i \in \underline{n}\ [\phi_i \leq \tau\ ]$.
(5) $\sigma \leq \tau \ \Rightarrow \ \exists \phi_i\ (i \in \underline{n}), \psi_j\ (j \in \underline{m})\ [\sigma = \cap_{\underline{n}}\phi_i \ \& \ \tau = \cap_{\underline{m}}\psi_j \ \& \ \forall j \in \underline{m}\ \exists i \in \underline{n}\ [\phi_i \leq \psi_j\ ]\ ]$.

Proof. Easy. ■

The (essential) intersection type assignment system is constructed from the set of strict types and the following derivation rules. In this way a syntax directed system is obtained, that satisfies the main properties of the BCD-system (see [2]; the presentation of the derivation rules in that paper differs from the one used here).

Definition 4 (*Intersection Type Assignment*). (1) *Intersection type assignment* and *intersection derivations* are defined by the following natural deduction system:

$$(Ax): \frac{}{B, x{:}\sigma \vdash x : \psi}\ (\sigma \leq \psi) \qquad (\cap I): \frac{B \vdash M{:}\phi_1 \quad \cdots \quad B \vdash M{:}\phi_n}{B \vdash M : \cap_{\underline{n}} \phi_i}\ (n \geq 0)$$

$$(\to I): \frac{B, x{:}\sigma \vdash M{:}\phi}{B \vdash \lambda x.M : \sigma \to \phi} \qquad (\to E): \frac{B \vdash M : \sigma \to \phi \quad B \vdash N : \sigma}{B \vdash MN : \phi}.$$

(2) We write $B \vdash M{:}\sigma$ if this statement is derivable using an intersection derivation, and write $\mathcal{D} :: B \vdash M{:}\sigma$ to specify that this result was obtained through the derivation $\mathcal{D}$.

Notice that $B \vdash M{:}\omega$, for all B and M, as a special case of rule $(\cap I)$.

We should emphasise the difference between this notion of type assignment and the strict one that was defined in [3]; instead of the rule (Ax) given above, it contained the rule

$$(\cap E): \frac{}{B, x{:}\cap_{\underline{n}} \phi_i \vdash_{\mathrm{s}} x : \phi_i}\ (n \geq 1, i \in \underline{n}).$$

Notice that this rule is a special case of rule (Ax) in that $\cap_{\underline{n}} \phi_i \leq \phi_i$, for all $i \in \underline{n}$. This is, in fact, the *only* difference between *strict* and *non-strict* type assignment. As for the difference in derivable statements, in the essential system it is possible to derive $\vdash \lambda x.x : (\alpha \to \beta) \to (\alpha \cap \gamma) \to \beta$, which is not possible in '$\vdash_{\mathrm{s}}$'.

For this notion of type assignment, the following properties hold:

Lemma 5 (*Generation Lemma*). (1) $B \vdash x{:}\sigma \iff \exists \rho \in \mathcal{T}\ [x{:}\rho \in B\ \&\ \rho \leq \sigma\ ]$.
(2) $B \vdash MN{:}\phi \iff \exists \tau \in \mathcal{T}\ [B \vdash M{:}\tau \to \phi\ \&\ B \vdash N{:}\tau\ ]$.
(3) $B \vdash \lambda x.M{:}\sigma \iff \exists \rho \in \mathcal{T}, \phi \in \mathcal{T}_{\mathrm{s}}\ [\sigma = \rho \to \phi\ \&\ B, x{:}\rho \vdash M{:}\phi\ ]$.
(4) $B \vdash M{:}\sigma\ \&\ \sigma \in \mathcal{T} \iff \exists \phi_1, \ldots, \phi_n\ [\sigma = \cap_{\underline{n}} \phi_i\ \&\ \forall i \in \underline{n}\ [B \vdash M{:}\phi_i\ ]]$.
(5) $B \vdash M{:}\sigma \iff \{x{:}\tau \in B \mid x \in fv(M)\} \vdash M{:}\sigma$.

Proof. Easy. ■

Some of the properties of this system, proved in [2], are:

Property 6 (*[2]*). (1) *If $M \to_\eta N$ and $B \vdash M{:}\sigma$, then $B \vdash N{:}\sigma$.*
(2) *If $M =_\beta N$, then $B \vdash M{:}\sigma$ if and only if $B \vdash N{:}\sigma$.*

Although the rule (Ax) is defined only for term variables, $\vdash$ is closed for $\leq$ and weakening; the proof first appeared in [2], and is repeated here for completeness and its importance for this paper.

Lemma 7 (*[2]*). *If $B \vdash M{:}\sigma$ and $B' \leq B, \sigma \leq \tau$, then $B' \vdash M{:}\tau$, so the following is an admissible rule in $\vdash$:*

$$(\leq): \frac{B \vdash M{:}\sigma}{B' \vdash M{:}\tau}\ (B' \leq B, \sigma \leq \tau).$$

Proof. By induction on $\vdash$.

(Ax) : Then $M \equiv x$, and there is $x{:}\rho \in B$ such that $\rho \leq \sigma$. Since $B' \leq B$, there is $x{:}\mu \in B'$ such that $\mu \leq \rho$. Notice that $\mu \leq \rho \leq \sigma \leq \tau$, so, by Lemma 5(1), $B' \vdash x{:}\tau$.

$(\to I)$: Then $M \equiv \lambda x.M'$, and there are ρ, ϕ such that $\sigma = \rho \to \phi$ and $B, x{:}\rho \vdash M'{:}\phi$. By Lemma 3(5) and (3) there are ρ_i, ϕ_i $(i \in \underline{n})$ such that $\tau = \cap_{\underline{n}}(\rho_i \to \phi_i)$, and for $i \in \underline{n}$, $\rho_i \leq \rho$ and $\phi \leq \phi_i$. Since $B' \leq B$ and $\rho_i \leq \rho$, also $B', x{:}\rho_i \leq B, x{:}\rho$, and by induction $B, x{:}\rho_i \vdash M'{:}\phi_i$. So, by $(\to I)$, for every $i \in \underline{n}$, $B \vdash \lambda x.M'{:}\rho_i \to \phi_i$, so, by $(\cap I)$, $B \vdash \lambda x.M'{:}\tau$.

$(\to E)$: Then $M \equiv M_1 M_2$ and there is an μ such that $B \vdash M_1{:}\mu \to \sigma$ and $B \vdash M_2{:}\mu$. Since $\sigma \leq \tau$, also $\mu \to \sigma \leq \mu \to \tau$ and, by induction, $B \vdash M_1{:}\mu \to \tau$. Then, by $(\to E)$, $B \vdash M_1 M_2{:}\tau$.

$(\cap I)$: Then $\sigma = \cap_{\underline{n}} \phi_i$, and, for every $i \in \underline{n}$, $B \vdash M{:}\phi_i$. By Lemma 3(5), there are ψ_j $(j \in \underline{m})$ such that $\tau = \cap_{\underline{m}} \psi_j$ and, for every $j \in \underline{m}$, there is an $i \in \underline{n}$ such that $\phi_i \leq \psi_j$. By induction, for $j \in \underline{m}$, $B' \vdash M{:}\psi_j$. But then $B' \vdash M{:}\tau$, by $(\cap I)$. ■

Notice that, although the proof above is constructive, it is not sufficient to show the result of this paper. We need not just *a* derivation for the desired result, but *all*; see also the example after Definition 9.

We will use the following short-hand notation for derivations.

Definition 8. (1) $\mathcal{D} = \langle Ax\rangle :: B \vdash x:\sigma$ if $\mathcal{D}$ consists of nothing but an application of rule (Ax).

(2) $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle$, if and only if there are ϕ_i $(i \in \underline{n})$ such that, for all $i \in \underline{n}$, $\mathcal{D}_i :: B \vdash M:\phi_i$, $\mathcal{D}$ is obtained from $\mathcal{D}_1, \ldots, \mathcal{D}_n$ by applying rule $(\cap I)$, and $\mathcal{D} :: B \vdash M:\cap_{\underline{n}}\phi_i$.

(3) $\mathcal{D} = \langle \mathcal{D}_1, \to I\rangle$, if and only if there are M_1, σ, ϕ such that $\mathcal{D}_1 :: B, x{:}\sigma \vdash M_1:\phi$, $\mathcal{D}$ is obtained from $\mathcal{D}_1$ by applying rule $(\to I)$, and $\mathcal{D} :: B \vdash \lambda x.M_1:\sigma\to\phi$.

(4) $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle$, if and only if there are P, Q, σ, and ϕ such that $\mathcal{D}_1 :: B \vdash P:\sigma\to\phi$ and $\mathcal{D}_2 :: B \vdash Q:\sigma$, $\mathcal{D}$ is obtained from $\mathcal{D}_1$ and $\mathcal{D}_2$ by applying rule $(\to E)$, and $\mathcal{D} :: B \vdash PQ:\phi$.

We now extend the relation $\leq$ to derivations in $\vdash$; this notion is pivotal in the proof of strong normalisation of derivation reduction.

Definition 9. (1) $\langle Ax\rangle :: B \vdash x:\sigma \leq \langle Ax\rangle :: B' \vdash x:\sigma'$ for all $B' \leq B$, and $\sigma \leq \sigma'$.

(2) $\langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle :: B \vdash M:\cap_{\underline{n}}\phi_i \leq \langle \mathcal{D}'_1, \ldots, \mathcal{D}'_m, \cap I\rangle :: B' \vdash M:\cap_{\underline{m}}\psi_j$, if and only if for every $j \in \underline{m}$ there exists an $i \in \underline{n}$ such that $\mathcal{D}_i \leq \mathcal{D}'_j$.

(3) $\langle \mathcal{D}_1 :: B, x{:}\sigma \vdash M:\phi, \to I\rangle :: B \vdash \lambda x.M:\sigma\to\phi \leq \langle \mathcal{D}'_1 :: B', x{:}\tau \vdash M:\psi, \to I\rangle :: B' \vdash \lambda x.M':\tau\to\psi$ if and only if $\mathcal{D}_1 \leq \mathcal{D}'_1$.

(4) $\langle \mathcal{D}_1 :: B \vdash P:\sigma\to\phi, \mathcal{D}_2 :: B \vdash Q:\sigma, \to E\rangle :: B \vdash PQ:\phi \leq \langle \mathcal{D}'_1 :: B \vdash P:\tau\to\psi, \mathcal{D}'_2 :: B \vdash Q:\tau, \to E\rangle :: B \vdash PQ:\psi$ if and only if $\mathcal{D}'_1 \leq \mathcal{D}_1$, and $\mathcal{D}'_2 \geq \mathcal{D}_2$.

Notice that '$\leq$' is contra-variant in $(\to E)$; this is especially important in the proof of Lemma 18.

Example 10. Let $B = x{:}\alpha\to\omega\to\gamma, y{:}\omega, z{:}\alpha$; take the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash x:\alpha\to\omega\to\gamma} \quad \dfrac{}{B \vdash z:\alpha}}{B \vdash xz:\omega\to\gamma} \quad \dfrac{}{B \vdash yz:\omega}}{B \vdash xz(yz):\gamma}}{B\backslash z \vdash \lambda z.xz(yz):\alpha\to\gamma}}{B\backslash y, z \vdash \lambda yz.xz(yz):\omega\to\alpha\to\gamma}}{\varnothing \vdash \lambda xyz.xz(yz):(\alpha\to\omega\to\gamma)\to\omega\to\alpha\to\gamma}\ (\cap I)$$

As $(\alpha\to\omega\to\gamma)\to\omega\to\alpha\to\gamma \leq (\alpha\to\omega\to\gamma)\to(\delta\to\beta)\to\alpha\cap\delta\to\gamma$, the following derivation is larger in the sense of $\leq$ on derivations (where $B' = x{:}\alpha\to\omega\to\gamma, y{:}\delta\to\beta, z{:}\alpha\cap\delta$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B' \vdash x:\alpha\to\beta\to\gamma}\ (\alpha\to\omega\to\gamma \leq \alpha\to\beta\to\gamma) \quad \dfrac{}{B' \vdash z:\alpha}}{B' \vdash xz:\beta\to\gamma} \quad \dfrac{\dfrac{}{B' \vdash y:\delta\to\beta} \quad \dfrac{}{B' \vdash z:\delta}}{B' \vdash yz:\beta}}{B' \vdash xz(yz):\gamma}}{B'\backslash z \vdash \lambda z.xz(yz):\alpha\cap\delta\to\gamma}}{B'\backslash y, z \vdash \lambda yz.xz(yz):(\delta\to\beta)\to\alpha\cap\delta\to\gamma}}{\varnothing \vdash \lambda xyz.xz(yz):(\alpha\to\omega\to\gamma)\to(\delta\to\beta)\to\alpha\cap\delta\to\gamma}$$

On the other hand, the derivation 'generated' by Lemma 7 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B' \vdash x:\alpha\to\omega\to\gamma}\quad\dfrac{}{B' \vdash z:\alpha}}{B' \vdash xz:\omega\to\gamma}\quad\dfrac{}{B' \vdash yz:\omega}}{B' \vdash xz(yz):\gamma}}{B'\backslash z \vdash \lambda z.xz(yz):\alpha\cap\delta\to\gamma}}{B'\backslash y,z \vdash \lambda yz.xz(yz):(\delta\to\beta)\to\alpha\cap\delta\to\gamma}}{\varnothing \vdash \lambda xyz.xz(yz):(\alpha\to\omega\to\gamma)\to(\delta\to\beta)\to\alpha\cap\delta\to\gamma}\ (\cap I)$$

This derivation is also larger, but limiting the relation to just this choice would not give sufficient expressive power in the proof of the main result of this paper; we need there to include all those derivations that are larger in the more general sense, as is the first derivation.

The following is easy to show, and establishes the relation between '$\leq$' on types and '$\leq$' on derivations:

Lemma 11. (1) *If* $\mathcal{D} :: B \vdash M:\sigma$ *and* $B' \leq B$, $\sigma \leq \sigma'$, *then there exists* $\mathcal{D}' \geq \mathcal{D}$ *such that* $\mathcal{D}' :: B' \vdash M':\sigma'$.
(2) *If* $\mathcal{D} :: B \vdash M:\sigma \leq \mathcal{D}' :: B' \vdash M':\sigma'$, *then* $B' \leq B$, $\sigma \leq \sigma'$.

Proof. (1) We separate two cases:

($\sigma' \in \mathcal{T}_s$) : By induction on the structure of derivations.

(Ax) : Then $\mathcal{D} = \langle Ax\rangle :: B, x{:}\rho \vdash x:\sigma$, with $\rho \leq \sigma$. Since $B' \leq B, x{:}\rho$, there exists $x{:}\mu \in B'$ such that $\mu \leq \rho \leq \sigma \leq \sigma'$. Take $\mathcal{D}' = \langle Ax\rangle :: B' \vdash x:\sigma'$, then $\mathcal{D} \leq \mathcal{D}'$.

($\cap I$) : Then $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle :: B \vdash M:\cap_{\underline{n}}\phi_i$, with $\mathcal{D}_i :: B \vdash M:\phi_i$, for $i \in \underline{n}$; notice that $\mathcal{D} \leq \mathcal{D}_i$. Then, by Lemma 3, there exists $j \in \underline{n}$ such that $\phi_j \leq \sigma'$, and, by induction, there exists $\mathcal{D}'_j :: B' \vdash M:\sigma'$, with $\mathcal{D}_j \leq \mathcal{D}'_j$. Take $\mathcal{D}' = \mathcal{D}'_j$, then $\mathcal{D} \leq \mathcal{D}'$.

($\to I$) : Then $\mathcal{D} = \langle \mathcal{D}_1 :: B, x{:}\tau \vdash M':\phi, \to I\rangle :: B \vdash \lambda x.M':\tau\to\phi$, and $\sigma = \tau\to\phi$. Then $\sigma' = \rho\to\psi$ such that $\rho \leq \tau$ and $\phi \leq \psi$, and $B', x{:}\rho \leq B, x{:}\sigma$. Then, by induction, there exists $\mathcal{D}'_1 \geq \mathcal{D}_1$, such that $\mathcal{D}'_1 :: B', x{:}\rho \vdash M':\psi$. Take $\mathcal{D}' = \langle \mathcal{D}'_1 :: B', x{:}\rho \vdash M':\psi, \to I\rangle :: B' \vdash \lambda x.M':\rho\to\phi$, then $\mathcal{D} \leq \mathcal{D}'$.

($\to E$) : Then $\sigma \in \mathcal{T}_s$, and $\mathcal{D} = \langle \mathcal{D}_1 :: B \vdash M_1:\gamma\to\sigma, \mathcal{D}_2 :: B \vdash M_2:\gamma, \to E\rangle :: B \vdash M_1M_2:\sigma$. Notice that $\gamma\to\sigma \leq \gamma\to\sigma'$; so, by induction, there exists $\mathcal{D}'_1 \geq \mathcal{D}_1$ such that $\mathcal{D}'_1 :: B' \vdash M'_1:\gamma\to\sigma'$; notice that $\mathcal{D}_2 \leq \mathcal{D}_2$. Take $\mathcal{D}' = \langle \mathcal{D}'_1, \mathcal{D}_2, \to E\rangle :: B' \vdash M_1M_2:\sigma'$, then $\mathcal{D} \leq \mathcal{D}'$.

($\sigma' = \cap_{\underline{n}}\phi'_i$) : By Lemma 3, for $i \in \underline{n}$, $\sigma \leq \phi'_i \in \mathcal{T}_s$; by part 2, there exists $\mathcal{D}'_i \geq \mathcal{D}_i$ such that $\mathcal{D}'_i :: B' \vdash M:\phi'_i$. Take $\mathcal{D}' = \langle \mathcal{D}'_i, \ldots, \mathcal{D}'_n, \cap I\rangle :: B' \vdash M:\sigma'$, then $\mathcal{D} \leq \mathcal{D}'$.

(2) Easy, from Definition 9. ■

2. Derivation reduction

In this section, we will define a notion of reduction on derivations and show this notion to be strongly normalisable in the next section.

We start by defining a notion of reduction on derivations $\mathcal{D} :: B \vdash M:\sigma$ that will follow ordinary reduction, by contracting typed redexes that occur in $\mathcal{D}$, i.e. redexes for subterms of M of the shape $(\lambda x.P)Q$, for which the following is a subderivation of $\mathcal{D}$:

$$\dfrac{\dfrac{\dfrac{}{x{:}\sigma \vdash x:\rho}\ (\sigma \leq \rho)}{\mathcal{D}_1}}{\dfrac{B, x{:}\sigma \vdash P:\phi}{B \vdash \lambda x.P:\sigma\to\phi}\ (\to I)}\qquad\dfrac{\mathcal{D}_2}{B \vdash Q:\sigma}}{B \vdash (\lambda x.P)Q:\phi}\ (\to E)$$

(a derivation of this shape, where an introduction rule is followed by the corresponding elimination rule, is called a *cut*). As can be expected, the effect of this reduction will be that this derivation will be replaced by a derivation for the contractum $P[Q/x]$; this can be regarded as a generalisation of cut-elimination, but, because the system at hand uses intersection types together with the relation '$\leq$', has to be defined with care. So, when contracting $\mathcal{D}$ it is in general not the case that the derivation $\mathcal{D}_2$ will just be inserted in the positions of $\mathcal{D}_1$ where a type for the variable x is derived, since that would give an illegal derivation. The ($\leq$)-step 'to be applied at the end of $\mathcal{D}_2$' has to be pushed upwards. We have shown that this is possible in Lemma 11(1); this procedure, in general, changes the structure of the derivation $\mathcal{D}_2$.

Before formally defining reduction on derivations, we will first define a notion of substitution on derivations. It is this operation that deals adequately with the occurrences of derivation rule ($\leq$) in the leaves of a derivation.

Definition 12 (*Derivation Substitution*). For $\mathcal{D} :: B, x{:}\sigma \vdash M : \tau$, and $\mathcal{D}_0 :: B \vdash N : \sigma$, the result $\mathcal{D}'$ of substituting $\mathcal{D}_0$ in $\mathcal{D}$, $\mathcal{D}[\mathcal{D}_0/x{:}\sigma] :: B \vdash M[N/x] : \tau$ is inductively defined by:

(1) $\mathcal{D} = \langle Ax\rangle :: B, x{:}\sigma \vdash x : \tau$, with $\sigma \leq \tau$. Let $\mathcal{D}'_0$ be such that $\mathcal{D}_0 \leq \mathcal{D}'_0 :: B \vdash N : \tau$, then $\mathcal{D}[\mathcal{D}_0/x{:}\sigma] = \mathcal{D}'_0$.
(2) $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle :: B, x{:}\sigma \vdash M : \cap_{\underline{n}} \psi_j$, so $\mathcal{D}_i :: B, x{:}\sigma \vdash M : \psi_i$ for $i \in \underline{n}$.
Let $\mathcal{D}'_i = \mathcal{D}_i[\mathcal{D}_0/x{:}\sigma] :: B \vdash M[N/x] : \psi_i$, then

$$\mathcal{D}' = \langle \mathcal{D}'_1, \ldots, D'_n, \cap I\rangle :: B \vdash M[N/x] : \cap_{\underline{n}} \psi_j = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle[\mathcal{D}_0/x{:}\sigma].$$

(3) $\mathcal{D} = \langle \mathcal{D}_1 :: B, x{:}\sigma, y{:}\tau \vdash M_1 : \psi, \to I\rangle :: B, x{:}\sigma \vdash \lambda y.M_1 : \tau \to \psi$.
Let $\mathcal{D}'_1 = \mathcal{D}_1[\mathcal{D}_0/x{:}\sigma] :: B, y{:}\tau \vdash M_1[N/x] : \psi$, then

$$\mathcal{D}' = \langle \mathcal{D}'_1, \to I\rangle :: B \vdash (\lambda y.M_1)[N/x] : \tau \to \psi = \langle \mathcal{D}'_1, \to I\rangle[\mathcal{D}_0/x{:}\sigma]$$

(4) $\mathcal{D} = \langle \mathcal{D}_1 :: B, x{:}\sigma \vdash P : \tau \to \psi, \mathcal{D}_2 :: B, x{:}\sigma \vdash Q : \tau, \to E\rangle :: B, x{:}\sigma \vdash PQ : \psi$.
Let $\mathcal{D}'_1 = \mathcal{D}_1[\mathcal{D}_0/x{:}\sigma] :: B \vdash P[N/x] : \tau \to \psi$, $\mathcal{D}'_2 = \mathcal{D}_2[\mathcal{D}_0/x{:}\sigma] :: B \vdash Q[N/x] : \tau$,
then $\mathcal{D}' = \langle \mathcal{D}'_1, \mathcal{D}'_2, \to E\rangle :: B \vdash (PQ)[N/x] : \psi = \langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle[\mathcal{D}_0/x{:}\sigma]$.

Also, we need to define the concept of 'position of a subderivation in a derivation'.

Definition 13. Let $\mathcal{D}$ be a derivation, and $\mathcal{D}'$ be a subderivation of $\mathcal{D}$. The position p of $\mathcal{D}'$ in $\mathcal{D}$ is defined by:

(1) If $\mathcal{D}' = \mathcal{D}$, then $p = \varepsilon$.
(2) If the position of $\mathcal{D}'$ in $\mathcal{D}_1$ is q, and $\mathcal{D} = \langle \mathcal{D}_1, \to I\rangle$, or $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle$, then $p = 1q$.
(3) If the position of $\mathcal{D}'$ in $\mathcal{D}_2$ is q, and $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle$, then $p = 2q$.
(4) If the position of $\mathcal{D}'$ in $\mathcal{D}_i$ ($i \in \underline{n}$) is q, and $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle$, then $p = q$.

We now can give a clear definition of reductions on derivations.

Definition 14 (*Derivation Reduction*). (1) We say that the derivation $\mathcal{D} :: B \vdash M : \sigma$ *reduces to* $\mathcal{D}' :: B \vdash M' : \sigma$ *at position* p *with redex* R, if and only if:

($\sigma \in \mathcal{T}_s$) : (a) $\mathcal{D} = \langle\langle \mathcal{D}_1, \to I\rangle, \mathcal{D}_2, \to E\rangle :: B \vdash (\lambda x.M)N : \sigma$. Then $\mathcal{D}$ reduces to $\mathcal{D}_1[\mathcal{D}_2/x{:}\tau] :: B \vdash M[N/x] : \sigma$ at position ε with redex $(\lambda x.M)N$.

(b) If $\mathcal{D}_1$ reduces to $\mathcal{D}'_1$ at position p with redex R, then

(i) $\mathcal{D} = \langle \mathcal{D}_1, \to I\rangle :: B \vdash \lambda x.P : \alpha \to \beta$ reduces at position $1p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}'_1, \to I\rangle :: B \vdash \lambda x.P' : \alpha \to \beta$.

(ii) $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle :: B \vdash PQ : \sigma$ reduces at position $1p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}'_1, \mathcal{D}_2, \to E\rangle :: P'Q : \sigma$.

(iii) $\mathcal{D} = \langle \mathcal{D}_2, \mathcal{D}_1, \to E\rangle :: B \vdash PQ : \sigma$ reduces at position $2p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}_2, \mathcal{D}'_1, \to E\rangle :: PQ' : \sigma$.

($\sigma = \cap_{\underline{n}} \phi_i$) : If $\mathcal{D} :: B \vdash M : \cap_{\underline{n}} \phi_i$, then, for every $i \in \underline{n}$, there is a $\mathcal{D}_i$, such that $\mathcal{D}_i :: B \vdash M : \psi_i$, and $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle$. If there is an $i \in \underline{n}$ such that $\mathcal{D}_i$ reduces to $\mathcal{D}'_i$ at position p with redex R $= (\lambda x.P)Q$ (a subterm of M), then, for all $1 \leq j \neq i \leq n$, either

(a) there is no redex at position p because there is no subderivation at that position, and $\mathcal{D}'_j = \mathcal{D}_j$, with $P[Q/x]$ instead of R, or

(b) $\mathcal{D}_j$ reduces to $\mathcal{D}'_j$ at position p with redex R.

Then $\mathcal{D}$ reduces to $\langle \mathcal{D}'_1, \ldots, \mathcal{D}'_n, \cap I\rangle$ at position p with redex R.

(2) We write $\mathcal{D} \to_{\mathcal{D}} \mathcal{D}'$ if there exists a position p and redex R such that $\mathcal{D}$ reduces to $\mathcal{D}'$ at position p with redex R. If $\mathcal{D}_1 \to_{\mathcal{D}} \mathcal{D}_2 \to_{\mathcal{D}} \mathcal{D}_3$, then $\mathcal{D}_1 \to_{\mathcal{D}} \mathcal{D}_3$.
(3) We abbreviate '$\mathcal{D}$ is strongly normalisable with respect to $\to_{\mathcal{D}}$' by '$SN(\mathcal{D})$', and use $\mathcal{SN}$ for the set of strongly normalisable derivations: $\mathcal{SN} = \{\mathcal{D} \mid SN(\mathcal{D})\}$.

Notice that this reduction corresponds to contracting a redex in the conclusion of the derivation only if that redex appears at least once in a subderivation with type different from ω.

The following lemma formulates the relation between derivation reduction and β-reduction, and is easy to show.

Lemma 15 (*[3]*). *Let $\mathcal{D} :: B \vdash M : \sigma$, and $\mathcal{D} \to_{\mathcal{D}} \mathcal{D}' :: B \vdash N : \sigma$, then $M \twoheadrightarrow_{\beta} N$.*

The following states some standard properties of strong normalisation.

Lemma 16. (1) $SN(\langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle) \Rightarrow SN(\mathcal{D}_1)$ & $SN(\mathcal{D}_2)$.
(2) *If $SN(\mathcal{D}_1 :: B \vdash xM_1 \cdots M_n : \sigma \to \phi)$ and $SN(\mathcal{D}_2 :: B \vdash N : \sigma)$, then also $SN(\langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle)$.*
(3) *Let $\mathcal{D} :: B \vdash M : \sigma$ be $\langle \mathcal{D}_1 \cap \cdots \cap \mathcal{D}_n, \cap I\rangle$, so $\sigma = \cap_{\underline{n}} \phi_i$. If $\mathcal{D} \to_{\mathcal{D}} \mathcal{D}' :: B \vdash M' : \sigma$ at position p, then there exists an $i \in \underline{n}$ such that $\mathcal{D}_i$ reduces to $\mathcal{D}'_i$ at position p with redex* R.
(4) *For all $i \in \underline{n}$, $SN(\mathcal{D}_1 :: B \vdash M : \sigma_i)$ if and only if $SN(\langle \mathcal{D}_1 \cap \cdots \cap \mathcal{D}_n, \cap I\rangle)$.*
(5) *If $SN(\mathcal{D}_1 :: B \vdash C[M[N/x]] : \sigma)$, and $SN(\mathcal{D}_2 :: B \vdash N : \rho)$, then there exists a derivation $\mathcal{D}_3$ such that $SN(\mathcal{D}_3 :: B \vdash C[(\lambda y.M)N] : \sigma)$.*

3. Strong normalisation of derivation reduction

In order to prove that each derivation in '$\vdash$' is strongly normalisable with respect to $\to_{\mathcal{D}}$, a notion of computable derivations is defined (based on the technique of computability predicates [12,10]). We will show that all computable derivations are strongly normalisable with respect to derivation reduction, and then that all derivations in '$\vdash$' are computable.

Definition 17 (*Computable Derivations ([3])*). The Computability Predicate $Comp(\mathcal{D})$ is defined inductively on types by:

$$
\begin{array}{lcl}
Comp(\mathcal{D} :: B \vdash M : \varphi) & \Longleftrightarrow & SN(\mathcal{D}) \\
Comp(\mathcal{D}_1 :: B \vdash M : \sigma \to \phi) & \Longleftrightarrow & Comp(\mathcal{D}_2 :: B \vdash N : \sigma) \Rightarrow Comp(\langle \mathcal{D}_1, \mathcal{D}_2, \to E\rangle :: \\
 & & B \vdash MN : \phi) \\
Comp(\langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle :: B \vdash M : \cap_{\underline{n}} \phi_i) & \Longleftrightarrow & \forall i \in \underline{n}\ [Comp(\mathcal{D}_i :: B \vdash M : \phi_i)].
\end{array}
$$

Notice that, as a special case for the third rule, we get $Comp(\langle \cap I\rangle :: B \vdash M : \omega)$.

The following lemma formulates the relation between the computability predicate and the relation $\leq$ on derivations, and is crucial for the proof of Theorem 21. The main difference between the solution of [3] and the one presented here lies in the fact that here we need to prove this lemma, whereas in [3], it is not needed at all. In the strict system, rule (Ax) corresponds to $(\cap E)$, and existence of a computable derivation of type $\cap_{\underline{n}} \phi_i$ immediately implies the existence of a computable derivation of type ϕ_i via the third part of Definition 17: it is a direct subderivation.

Lemma 18. *If $Comp(\mathcal{D} :: B \vdash M : \sigma)$, and $\mathcal{D} \leq \mathcal{D}'$, then $Comp(\mathcal{D}')$.*

Proof. By induction on the structure of types. Notice that, by Lemma 11(2), $\mathcal{D}' = B' \vdash M : \sigma'$, with $B' \leq B$, $\sigma \leq \sigma'$.

We distinguish two cases:

$(\sigma' \in \mathcal{T}_s)$: $(\sigma = \varphi)$: Since $\varphi \leq \sigma'$, also $\sigma' = \varphi$, and the result is immediate.

$(\sigma = \alpha \to \phi)$: Then $\sigma' = \rho \to \psi$, with $\rho \leq \alpha, \phi \leq \psi$, and let $\mathcal{D}' :: B \vdash M : \rho \to \psi$. To show $Comp(\mathcal{D}')$, by Definition 17, we assume $Comp(\mathcal{D}_0 :: B \vdash N : \rho)$, and use this to show $\langle \mathcal{D}', \mathcal{D}_0, \to E\rangle :: B \vdash MN : \psi$. Since $\mathcal{D}_0 \leq \mathcal{D}'_0 :: B \vdash N : \alpha$, we get $Comp(\mathcal{D}'_0)$ by induction. Assuming $Comp(\mathcal{D} :: B \vdash M : \alpha \to \phi)$, by Definition 17, $Comp(\langle \mathcal{D}, \mathcal{D}'_0, \to E\rangle :: B \vdash MN : \phi)$. Since

$$\langle \mathcal{D}, \mathcal{D}'_0, \to E\rangle \leq \langle \mathcal{D}', \mathcal{D}_0, \to E\rangle :: B \vdash MN : \psi,$$

we get, by induction $Comp(\langle \mathcal{D}', \mathcal{D}_0, \to E\rangle)$. So $Comp(\mathcal{D} :: B \vdash M : \rho \to \psi)$ by Definition 17.

$(\sigma = \cap_{\underline{n}}\phi_i)$: If $Comp(\mathcal{D} :: B \vdash M : \cap_{\underline{n}}\phi_i)$, then $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle$, by Definition 17, and $Comp(\mathcal{D}_i :: B \vdash M : \phi_i)$ for $i \in \underline{n}$. Since $\cap_{\underline{n}}\phi_i \leq \sigma'$, by Lemma 3, there exists $i_j \in \underline{n}$ such that $\phi_{i_j} \leq \sigma'$. Then $\mathcal{D} \leq \mathcal{D}_{i_j} :: B \vdash M : \tau_j$ and, by induction, $Comp(\mathcal{D}_{i_j})$.

$(\sigma' = \cap_{\underline{n}}\phi_i)$: If $Comp(\mathcal{D} :: B \vdash M : \cap_{\underline{n}}\phi_i)$, then, by Definition 17, for every $i \in \underline{n}$ there exists $\mathcal{D}_i$ such that $Comp(\mathcal{D}_i :: B \vdash M : \phi_i)$, and $\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I\rangle$. Since $\cap_{\underline{n}}\phi_i \leq \sigma'$, by Lemma 3, $\sigma' = \cap_{\underline{m}}\psi_j$, and for all $j \in \underline{m}$ there exists $i_j \in \underline{n}$ such that $\phi_{i_j} \leq \psi_j$. Since $\mathcal{D}_i \leq \mathcal{D}_{i_j} :: B \vdash M : \psi_j$, by induction, $Comp(\mathcal{D}_{i_j})$, and, by Definition 17, $Comp(\langle \mathcal{D}_{i_1}, \ldots, \mathcal{D}_{i_m}, \cap I\rangle :: B \vdash M : \cap_{\underline{m}}\psi_j)$. ■

The following lemma states that *Comp* satisfies the standard properties of computability predicates, being that computability implies strong normalisation, and that, for the so-called *neutral* objects, also the converse holds; the proof is the same as that of [3].

Lemma 19 (*[3]*). (1) $Comp(\mathcal{D} :: B \vdash M : \sigma) \Rightarrow SN(\mathcal{D})$.
(2) $SN(\mathcal{D} :: B \vdash xM_1 \cdots M_m : \sigma) \Rightarrow Comp(\mathcal{D})$.

The following Theorem 21 shows that, if the instances of rule (Ax) are to be replaced by computable derivations, then the result itself will be computable. Before coming to this result, we show that the predicate is closed for subject-expansion; we will use an abbreviated notation, and write $\overrightarrow{P}$ for $P_1 \cdots P_n$, as well as $[\overrightarrow{N_i/x_i}]$ for $[N_1/x_1, \ldots, N_n/x_n]$, etc.

Lemma 20. *If $Comp(\mathcal{D}' :: B \vdash Q : \nu)$ and $Comp(\mathcal{D}[\mathcal{D}'/y{:}\nu] :: B \vdash M[Q/y]\overrightarrow{P} : \sigma)$, then there exists a derivation $\mathcal{D}''$ such that $Comp(\mathcal{D}'' :: B \vdash (\lambda y.M)Q\overrightarrow{P} : \sigma)$.*

Proof. By induction on the structure of types.

$(\sigma = \varphi)$:
$$\begin{array}{rl} Comp(\mathcal{D}[\mathcal{D}'/y{:}\nu] :: B \vdash M[Q/y]\overrightarrow{P} : \varphi) \ \&\ Comp(\mathcal{D}' :: B \vdash Q : \nu) & \Rightarrow \ (19(1)) \\ SN(\mathcal{D}[\mathcal{D}'/y{:}\nu]) \ \&\ SN(\mathcal{D}') & \Rightarrow \ (16(5)) \\ \exists \mathcal{D}''\ [SN(\mathcal{D}'' :: B \vdash (\lambda y.M)Q\overrightarrow{P} : \varphi)] & \Rightarrow \ (17) \\ \exists \mathcal{D}''\ [Comp(\mathcal{D}'' :: B \vdash (\lambda y.M)Q\overrightarrow{P} : \varphi)]. & \end{array}$$

$(\sigma = \tau \to \phi)$:
$$\begin{array}{rl} Comp(\mathcal{D}_1 :: B \vdash N : \tau) \ \&\ Comp(\mathcal{D}_2 :: B \vdash Q : \nu) & \Rightarrow \ (17) \\ Comp(\langle \mathcal{D}[\mathcal{D}'/y{:}\nu], \mathcal{D}_2, \to E\rangle :: B \vdash M[Q/y]\overrightarrow{P}N : \phi) & \Rightarrow \ (IH) \\ \exists \mathcal{D}''[Comp(\langle \mathcal{D}'', \mathcal{D}_2, \to E\rangle :: B \vdash (\lambda y.M)Q\overrightarrow{P}N : \phi) & \Rightarrow \ (17) \\ \exists \mathcal{D}''[Comp(\mathcal{D}'' :: B \vdash (\lambda y.M)Q\overrightarrow{P} : \tau \to \phi)] & \end{array}$$

$(\sigma = \cap_{\underline{n}}\phi_i)$: By induction and Definition 17. ■

We now come to the main result.

Theorem 21 (*Replacement Theorem*). *Let $B = x_1{:}\mu_1, \ldots, x_n{:}\mu_n$, $\mathcal{D} :: B \vdash M : \sigma$, and, assume that, for every $i \in \underline{n}$, there are $\mathcal{D}^i$, N^i such that $Comp(\mathcal{D}^i :: B' \vdash N^i : \mu_i)$. Then $Comp(\mathcal{D}[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}] :: B' \vdash M[\overrightarrow{N_i/x_i}] : \sigma)$.*

Proof. By induction on the structure of derivations.

(Ax) : Then $M \equiv x_j$, for some $j \in \underline{n}$, with $\mu_j \leq \sigma$. Since $\mathcal{D}^j \leq \mathcal{D}' :: B' \vdash N^j : \sigma$, from $Comp(\mathcal{D}^j)$, by Lemma 18, $Comp(\langle Ax\rangle :: B \vdash x_j : \sigma)[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}])$.

$(\to I)$: Then $\sigma = \tau \to \psi$, and $\mathcal{D} = \langle \mathcal{D}_1 :: B, y{:}\tau \vdash M' : \psi, \to I\rangle :: B \vdash \lambda y.M' : \tau \to \psi$.

$$\begin{array}{rl} \forall i \in \underline{n}\ [\mathcal{D}_j :: B \vdash N_i : \mu_i] \ \&\ Comp(\mathcal{D}_2 :: B \vdash P : \tau) & \Rightarrow \ (IH) \\ Comp(\mathcal{D}_1[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}, \mathcal{D}_2/y{:}\tau] :: B \vdash M[\overrightarrow{N/x}, P/y] : \psi) & \Rightarrow \ (20) \\ Comp(\langle \langle \mathcal{D}_1[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}], \to I\rangle, \mathcal{D}_2, \to E\rangle :: B \vdash (\lambda y.M[\overrightarrow{N_i/x_i}])P : \tau) & \Rightarrow \ (17) \\ Comp(\langle \mathcal{D}_1[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}], \to I\rangle :: B \vdash \lambda y.M[\overrightarrow{N_i/x_i}] : \tau \to \psi) & \end{array}$$

and $\mathcal{D}' = \langle \mathcal{D}_1[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}], \to I\rangle = \mathcal{D}[\overrightarrow{\mathcal{D}_i/x_i{:}\mu_i}]$.

Cases $(\cap I)$ and $(\to E)$ follow by induction. ■

Using this, we prove a strong normalisation result for derivation reduction.

Theorem 22 (*Strong Normalisation*). *If $\mathcal{D} :: B \vdash M : \sigma$, then $SN(\mathcal{D})$.*

Proof. By Lemma 19(2), for every $x_i{:}\tau_i \in B$, there exists $\mathcal{D}_{x_i} = \langle Ax\rangle :: x_i{:}\tau_i \vdash x_i{:}\tau_i$ such that $Comp(\mathcal{D}_{x_i})$, so by Theorem 21, $Comp(\mathcal{D}[\overbrace{\mathcal{D}_{x_i}/x_i{:}\tau_i}\,] :: B \vdash M[\overbrace{x_i/x_i}\,]{:}\sigma)$. Notice that $M[\overbrace{x_i/x_i}\,] = M$ and $\mathcal{D}[\overbrace{\mathcal{D}_{x_i}/x_i{:}\tau_i}\,] = \mathcal{D}$, and by Lemma 19(1), $SN(\mathcal{D})$. ■

4. Approximation and head normalisation

We will now show that the above strong normalisation result leads to the approximation theorem, for which we will prepare the ground by introducing the necessary concepts.

4.1. Approximants

The notion of approximant for lambda terms was first presented in [13], and is defined using the notion of terms in $\lambda\bot$-normal form (like in [6], $\bot$ (called *bottom*) is used, instead of Ω; also, the symbol $\sqsubseteq$ is used as a relation on $\lambda\bot$-terms, inspired by a similar relation defined on Böhm-trees in [6]).

Definition 23. (1) The set of $\lambda\bot$-*terms* is defined as the set λ of lambda terms, by extending the syntax with $\bot$ (called *bottom*).

(2) The notion of reduction $\to_{\beta\bot}$ is defined as $\to_\beta$, extended by:

$$\lambda x.\bot \ \to_{\beta\bot}\ \bot \quad \textit{and} \quad \bot M \ \to_{\beta\bot}\ \bot .$$

(3) The set of *normal forms for elements of* $\lambda\bot$ *with respect to* $\to_{\beta\bot}$, is the set $\mathcal{N}$ of $\lambda\bot$-*normal forms* or *approximate normal forms*, ranged over by A, inductively defined by:

$$A ::= \bot \mid \lambda x.A\ (A \neq \bot) \mid x\,A_1 \cdots A_n\ (n \geq 0).$$

The rules of the system '$\vdash$' are generalised to terms containing $\bot$ by allowing for the terms to be elements of $\lambda\bot$. Notice that, because type assignment is almost syntax directed, if $\bot$ occurs in a term M and $\mathcal{D} :: B \vdash M{:}\sigma$, then in $\mathcal{D}$, $\bot$ appears in a position where the rule $(\cap I)$ is used with $n = 0$. Moreover, the terms $\lambda x.\bot$ and $\bot M_1 \cdots M_n$ are typeable by ω only.

Definition 24. (1) The relation $\sqsubseteq \subseteq \lambda\bot^2$ is defined by:

$$\bot \ \sqsubseteq\ M \qquad\qquad M \sqsubseteq M' \ \Rightarrow\ \lambda x.M \sqsubseteq \lambda x.M'$$
$$x \ \sqsubseteq\ x \qquad M_1 \sqsubseteq M_1' \ \&\ M_2 \sqsubseteq M_2' \ \Rightarrow\ M_1M_2 \sqsubseteq M_1'M_2'.$$

If $A \in \mathcal{N}$, $M \in \Lambda$, and $A \sqsubseteq M$, then A is called a *direct approximant* of M.

(2) The relation $\underset{\sim}{\sqsubseteq} \subseteq \mathcal{N} \times \lambda$ is defined by: $A \underset{\sim}{\sqsubseteq} M \iff \exists M' =_\beta M\ [A \sqsubseteq M']$.

(3) If $A \underset{\sim}{\sqsubseteq} M$, then A is called an *approximant* of M, and $\mathcal{A}(M) = \{\, A \in \mathcal{N} \mid A \underset{\sim}{\sqsubseteq} M \,\}$.

Type assignment is closed for '$\sqsubseteq$':

Lemma 25 (*Cf. [3]*). $B \vdash M{:}\sigma \ \&\ M \sqsubseteq M' \ \Rightarrow\ B \vdash M'{:}\sigma$.

Proof. By easy induction on the definition of $\sqsubseteq$. ■

The following definition introduces an operation of join on $\lambda\bot$-terms.

Definition 26. (1) On $\lambda\bot$, the partial mapping $\sqcup : \lambda\bot \times \lambda\bot \to \lambda\bot$ is defined by:

$$\bot\sqcup M \ \equiv\ M\sqcup\bot \equiv M \qquad (\lambda x.M)\sqcup(\lambda x.N) \ \equiv\ \lambda x.(M\sqcup N)$$
$$x\sqcup x \ \equiv\ x \qquad\qquad (M_1M_2)\sqcup(N_1N_2) \ \equiv\ (M_1\sqcup N_1)\,(M_2\sqcup N_2)$$

$\sqcup$ is pronounced as *join*.

(2) If $M\sqcup N$ is defined, then M and N are called *compatible*.

From now on, to shorten proofs, $\bot$ will be the same as the empty join, i.e. if $M \equiv M_1\sqcup\cdots\sqcup M_n$ $(= \sqcup_{\underline{n}} M_i)$, and $n = 0$, then $M \equiv \bot$.

The last alternative in the definition of $\sqcup$ defines the join on applications in a more general way than Scott's, that states $(M_1M_2)\sqcup(N_1N_2) \sqsubseteq (M_1\sqcup N_1)(M_2\sqcup N_2)$, since it is not always sure if a join of two arbitrary terms exists. However, we will use our more general definition only on terms that are compatible.

The following lemma shows that $\sqcup$ acts as least upper bound of compatible terms.

Lemma 27 ([3]). (1) *If $M_1 \sqsubseteq M$, and $M_2 \sqsubseteq M$, then $M_1 \sqcup M_2$ is defined, and:*

$$M_1 \sqsubseteq M_1 \sqcup M_2,\ M_2 \sqsubseteq M_1 \sqcup M_2,\ \text{and } M_1 \sqcup M_2 \sqsubseteq M.$$

(2) *If $M \sqsubseteq M_i$, for $i \in \underline{n}$, then $M \sqsubseteq \sqcup_{\underline{n}} M_i$.*
(3) *If $M \sqsubseteq N$, and $N \sqsubseteq P$, then $M \sqsubseteq P$.*
(4) *If $M \sqsubseteq M_1 M_2$ and $M \neq \bot$, then there are M_3, M_4 such that $M = M_3 M_4$, and $M_3 \sqsubseteq M_1$, $M_4 \sqsubseteq M_2$.*

Lemma 28. *If $\mathcal{D} :: B \vdash M : \sigma$, with $\mathcal{D}$ in normal form, then there exists $M' \in \mathcal{N}$ such that $M' \sqsubseteq M$, and $\mathcal{D} :: B \vdash M' : \sigma$.*

Proof. By induction on the structure of derivations.

($\mathcal{D} = \langle Ax \rangle$) : Immediate.
($\mathcal{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n, \cap I \rangle$) : Then $\sigma = \cap_{\underline{n}} \phi_i$ and, for every $i \in \underline{n}$, $\mathcal{D}_i :: B \vdash M : \phi_i$, and, by induction there exists $M_i \in \mathcal{N}$ such that $M_i \sqsubseteq M_1$ and $\mathcal{D}_i :: B \vdash M_i : \phi_i$. Notice that then these M_i are compatible, so $\sqcup_{\underline{n}} M_i$ exists, and by Lemma 25, also $\mathcal{D}_i :: B \vdash \sqcup_{\underline{n}} M_i : \phi_i$. Then, by rule $(\cap I)$, we have $B \vdash \sqcup_{\underline{n}} M_i : \cap_{\underline{n}} \phi_i$. Notice that, since $\sqcup$ acts as least upper bound, $\sqcup_{\underline{n}} M_i \sqsubseteq M$.
($\mathcal{D} = \langle \mathcal{D}_1, \rightarrow I \rangle$) : Then $M \equiv \lambda x.M_1$, and $\sigma = \alpha \rightarrow \phi$, and $B, x{:}\alpha \vdash M_1 : \phi$. So, by induction, there exists $M_1' \in \mathcal{N}$ such that $M_1' \sqsubseteq M_1$ and $B, x{:}\alpha \vdash M_1' : \phi$. Then, by rule $(\rightarrow I)$ we obtain $B \vdash \lambda x.M_1' : \alpha \rightarrow \phi$. Notice that $\lambda x.M_1' \sqsubseteq \lambda x.M_1$, and $\lambda x.M_1' \in \mathcal{N}$.
($\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2, \rightarrow E \rangle$) : Then $M \equiv M_1 M_2$, and there is a τ such that $B \vdash M_1 : \tau \rightarrow \phi$, and $B \vdash M_2 : \tau$. Then, by induction, there are $M_1', M_2' \in \mathcal{N}$ such that $M_1' \sqsubseteq M_1$, $M_2' \sqsubseteq M_2$, $B \vdash M_1' : \tau \rightarrow \phi$, and $B \vdash M_2' : \tau$. Then, by $(\rightarrow E)$, $B \vdash M_1' M_2' : \phi$. Notice that $M_1' M_2' \sqsubseteq M_1 M_2$. Since $\mathcal{D}$ is in normal form, $\mathcal{D}_1$ does not finish with $(\rightarrow I)$, so M_1' is not an abstraction. Since $\tau \rightarrow \phi$ is strict, neither can it be $\bot$; then $M_1' M_2' \in \mathcal{N}$. ■

Notice that the case $\sigma = \omega$ is present in the case $(\cap I)$ of the proof. Then $n = 0$, and $\sqcup_{\underline{n}} M_i = \bot$. Moreover, since M' need not be the same as M, the second derivation in Lemma 28 is not exactly the same; however, it has the same structure in terms of applied derivation rules.

5. Approximation and normalisation results

In this section, we will conclude the main contribution of this paper by showing two main results, that are both direct consequences of the strong normalisation result proved in Section 3. Both results have been proven, at least partially, in [1,2].

5.1. Approximation result

First we will prove the *approximation* result. From this, the well-known characterisation of (head) normalisation of lambda terms using intersection types follows easily, i.e., all terms having a (head-)normal form are typeable in '$\vdash$' (with a type without ω-occurrences). The second result is the well-known characterisation of strong normalisation of typeable lambda terms, i.e. all terms, typeable in '$\vdash$' without the type constant ω, are strongly normalisable.

Using Theorem 22, as for the BCD-system and the strict system, the relation between types assignable to a lambda term and those assignable to its approximants can be formulated as follows:

Theorem 29 (*Approximation Theorem Cf. [3]*).

$$B \vdash M : \sigma \iff \exists A \in \mathcal{A}(M)\ [B \vdash A : \sigma\ ].$$

Proof. ($\Rightarrow$) : Let $\mathcal{D} :: B \vdash M : \sigma$, then, by Theorem 22, $SN(\mathcal{D})$. Let $\mathcal{D}_0 :: B \vdash N : \sigma$ be the normal form of $\mathcal{D}$ with respect to $\rightarrow_{\mathcal{D}}$, then by Lemma 15, $M \twoheadrightarrow_\beta N$, and by Lemma 28, there is $N' \in \mathcal{N}$ such that $\mathcal{D}_0 :: B \vdash N' : \sigma$, and $N' \sqsubseteq N$, so $N' \in \mathcal{A}(M)$.
($\Leftarrow$) : Since $A \in \mathcal{A}(M)$, there is an M' such that $M' =_\beta M$ and $A \sqsubseteq M'$. Then, by Lemma 25, $B \vdash M' : \sigma$, and, by Theorem 6(2), also $B \vdash M : \sigma$. ■

Using this result, the following becomes easy; the proof is identical to that in [3], although formally on a different notion of type assignment.

Theorem 30 (*Cf. [3]*). $\exists B, \sigma\ [B \vdash M:\sigma] \iff M$ *has a head-normal form.*

5.2. Intersection type assignment without ω

As in [1] for the strict system, we will prove that the essential intersection type assignment system satisfies the (strong) normalisation properties of the BCD-system.

We will first prove that the set of all terms typeable by the system without ω is the set of all strongly normalisable terms. To start, we define ω-free types.

Definition 31. (1) $\mathcal{T}_{\not\omega}$, the set of *ω-free intersection types*, ranged over by σ, τ etc, is inductively defined by:

$$\phi, \psi \ ::= \ \varphi \mid (\sigma \to \phi)$$
$$\sigma, \tau \ ::= \ \cap_{\underline{n}} \phi_i, \quad (n \geq 1).$$

(2) On $\mathcal{T}_{\not\omega}$ the relation $\leq$ is as defined in Definition 2, except for the second alternative.

$$\begin{array}{rcll} \forall i \in \underline{n}\ [\cap_{\underline{n}}\phi_i & \leq & \sigma_i] & (n \geq 1) \\ \forall i \in \underline{n}\ [\sigma \leq \phi_i] & \Rightarrow & \sigma \leq \cap_{\underline{n}}\phi_i & (n \geq 1) \\ \sigma \leq \tau \leq \rho & \Rightarrow & \sigma \leq \rho. & \end{array}$$

The relations $\leq$ and $\sim$ are extended to bases as before.

(3) If $M:\sigma$ is derivable from a basis B, using only ω-free types and the derivation rules of '$\vdash$', we write $B \vdash_{\not\omega} M:\sigma$.

Notice that the only difference between this definition and Definitions 1 and 2 is that $n \geq 1$ rather than $n \geq 0$.

The following results were shown in [2].

Theorem 32 (*[2]*). (1) $\exists B, \sigma\ [B \vdash M:\sigma\ \&\ B, \sigma\ \omega\text{-free}] \iff M$ *has a normal form.*
(2) *If (the normal form) A is $\bot$-free, then there are B, and σ, such that $B \vdash_{\not\omega} A:\sigma$.*
(3) *If $B \vdash_{\not\omega} M[N/x]:\sigma$ and $B \vdash_{\not\omega} N:\rho$, then $B \vdash_{\not\omega} (\lambda x.M)N:\sigma$.*

5.3. Strong normalisation for intersection type assignment without ω

To show the strong normalisation result, notice that Theorem 32(3) is also essentially the proof for the statement that each strongly normalisable term can be typed in the system '$\vdash_{\not\omega}$'. A proof for this property in the context of the strict system appeared in [3]; since the strict system is a subsystem of the essential system, the proof is also valid here.

Theorem 33 (*[3]*). *If M is strongly normalisable, then there are B and σ such that $B \vdash_{\not\omega} M:\sigma$.*

Theorem 34 shows that the set of strongly normalisable terms is exactly the set of terms typeable in the intersection system without using the type constant ω.

Theorem 34. *If $B \vdash_{\not\omega} M:\sigma$ for some B and σ, then M is strongly normalisable with respect to $\to_\beta$.*

Proof. If $\mathcal{D} :: B \vdash_{\not\omega} M:\sigma$, then also $\mathcal{D} :: B \vdash M:\sigma$. Then $\mathcal{D}$ is strongly normalisable with respect to $\to_{\mathcal{D}}$ by Theorem 22. Since $\mathcal{D}$ contains no ω, all redexes in M correspond to redexes in $\mathcal{D}$; since derivation reduction does not introduce ω, this property is preserved by reduction. So also M is strongly normalisable with respect to $\to_\beta$. ■

Concluding remarks

This paper presents a result that has eluded me for more than a decade. The quest for it started in 1995, while I worked in Turin, and many afternoons were spent discussing with Mariangiola, trying to understand the ins and outs of the bottom system (see [3]), then thought to be the key to the proof of strong normalisation.

But I could not find the proof, and, over the following years, filled many pages with attempts, trying to find the correct notion of computability. During this initial proof search, I quickly found a solution for the strict system, but since I considered that solution almost trivial compared to what I was really looking for, I left it 'on the shelf' for

many years. Only after a casual conversation with Simona did I understand its importance and decided to submit it; it ended up as [3].

Then, years later, in the quiet, comfortable environment that is Inria in Sophia Antipolis, France, I finally had the decisive 'flash' and found the solution in the definition of the $\leq$-relation on derivations as defined here. As is not uncommon, all that was needed was a slight generalisation of the notions already at hand.

And it fills me with pride that I managed to finish this proof on time, to make it in time for my 'most important' intersection result to appear in the festschrift for my three generous, inspiring Italian instructors, who have received me so welcoming every time at *il dipartimento* in Turin when I started my regular sequence of visits in 1988.

References

[1] S. van Bakel, Complete restrictions of the intersection type discipline, Theoretical Computer Science 102 (1) (1992) 135–163.
[2] S. van Bakel, Intersection type assignment systems, Theoretical Computer Science 151 (2) (1995) 385–435.
[3] S. van Bakel, Cut-elimination in the strict intersection type assignment system is strongly normalising, Notre Dame Journal of Formal Logic 45 (1) (2004) 35–63.
[4] S. van Bakel, M. Fernández, Approximation and normalization results for typeable term rewriting systems, in: HOA'95, in: Lecture Notes in Computer Science, vol. 1074, Springer-Verlag, 1996, pp. 17–36.
[5] S. van Bakel, M. Fernández, Normalization results for typeable rewrite systems, Information and Computation 2 (133) (1997) 73–116.
[6] H. Barendregt, The Lambda Calculus: its Syntax and Semantics, North-Holland, Amsterdam, 1984 (revised edition).
[7] H. Barendregt, M. Coppo, M. Dezani-Ciancaglini, A filter lambda model and the completeness of type assignment, Journal of Symbolic Logic 48 (4) (1983) 931–940.
[8] M. Coppo, M. Dezani-Ciancaglini, An extension of the basic functionality theory for the λ-calculus, Notre Dame Journal of Formal Logic 21 (4) (1980) 685–693.
[9] H.B. Curry, R Feys, Combinatory Logic, vol. 1, North-Holland, Amsterdam, 1958.
[10] J.-Y. Girard, Y. Lafont, P. Taylor, Proofs and types, in: Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1989.
[11] S. Ronchi Della Rocca, B. Venneri, Principal type schemes for an extended type theory, Theoretical Computer Science 28 (1984) 151–169.
[12] W.W. Tait, Intensional interpretation of functionals of finite type I, Journal of Symbolic Logic 32 (2) (1967) 198–223.
[13] C.P. Wadsworth, The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus, SIAM Journal on Computing 5 (1976) 488–521.