Characterisation of Normalisation Properties for $\lambda\mu$ using Strict Negated Intersection Types

(ACM Transactions on Computational Logic, Volume 19 (1), 2018, Article No. 3)

Steffen van Bakel

Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2BZ, UK
s.vanbakel@imperial.ac.uk

Abstract

We show characterisation results for normalisation, head-normalisation, and strong normalisation for $\lambda\mu$ using intersection types. We reach these results for a strict notion of type assignment for $\lambda\mu$ that is the natural restriction of the domain-based system of [10] for $\lambda\mu$ by limiting the type inclusion relation to just intersection elimination. We show that this system respects $\beta\mu$-equality, by showing both soundness and completeness results. We then define a notion of reduction on derivations that corresponds to cut-elimination, and show that this is strongly normalisable.

We use this strong normalisation result to show an approximation result, and through that a characterisation of head-normalisation. Using the approximation result, we show that there is a very strong relation between the system of [10] and ours.

We then introduce a notion of type assignment that eliminates ω as an assignable type, and show, using the strong normalisation result for derivation reduction, that all terms typeable in this system are strongly normalisable as well, and show that all strongly normalisable terms are typeable.

We conclude by adding type variables to our system, and show that system essentially is that of [6].

keywords: lambda-mu calculus, intersection types, semantics, normalisation

Introduction

The Intersection Type Discipline [18] has proven to be an expressive tool for studying termination and semantics for Church's λ-calculus [20] (see also [17]). Intersection type assignment is defined as an extension of the standard, implicative type assignment known as Curry's system [24] (see also [28]), which expresses function composition and application; the extension made consists of relaxing the requirement that a parameter for a function should have a single type, adding the type constructor '$\cap$' next to '$\rightarrow$'. This simple extension allows for a great leap in complexity: not only can a (filter) model be built for the λ-calculus using intersection types, also strong normalisation (termination) can be characterised via assignable types; naturally, type assignment becomes undecidable. The literature on intersection types is vast; it was first defined by Coppo and Dezani [21] and its development took place over a number of years, culminating in the paper by Barendregt, Coppo, and Dezani-Ciancaglini [18], and has been explored by many since.

Semantics using intersection types cannot be defined for all calculi. In [8], the author investigated the possibility of defining semantics using intersection (and union) types in the context of the sequent calculus $\mathcal{X}$, as defined by Lengrand [30], and later studied with Lescanne and the author [13, 14]; $\mathcal{X}$ is a sequent calculus that enjoys the Curry-Howard isomorphism with respect to the implicative fragment of Gentzen's LK [25]. Later, in [5] the same was done for the calculus $\lambda\mu\tilde{\mu}$ defined by Curien and Herbelin [23]. The main conclusion of those papers is that, in symmetric calculi (like $\lambda\mu\tilde{\mu}$ and $\mathcal{X}$) it is inevitable that intersection and union are truly dual, and that the very nature of those calculi makes a sound and complete system unachievable, so there intersection (and union) types do not induce a semantics.

With those negative results in mind, the author investigated the question if these would also hold for less extensive systems based on classical logic, and explored the possibility of defining a notion of type assignment for Parigot's $\lambda\mu$ [33] that uses intersection types; $\lambda\mu$ is an extension of the untyped λ-calculus obtained by adding *named terms* and a name-abstraction operator μ. In [6] the author showed that (surprisingly, in view of the negative results mentioned above) it is indeed possible to define a notion of type assignment for $\lambda\mu$ that is closed under conversion. Since the point of departure was the work on $\mathcal{X}$ and $\lambda\mu\tilde{\mu}$, the system of that paper uses intersection types for term variables, and union types for names. As a direct consequence of this result, it is possible to define a filter model for $\lambda\mu$, and this was presented together with Barbanera and de'Liguoro [10]; the intersection type theory of that paper is developed with Streicher and Reus's [38] domain construction for $\lambda\mu$ as departure point. This later was followed by the proof that, as for the λ-calculus, the underlying intersection type system for $\lambda\mu$ allows for the full characterisation of strongly normalisable terms [11]. These papers were later combined (and revised) into [12].

One of the perhaps surprising aspects of the system defined in those papers is that union is no longer used, just intersection. Inspired by Streicher and Reus's domain, $\lambda\mu$-terms are separated into terms and *streams* (or *stacks*); then names act as the destination of streams, as variables are the destination of terms. Terms can be typed with types δ, which express functionality, and streams by types κ, essentially a sequence of δs, and intersection becomes the natural tool to group types for streams as well. Another difference with traditional notions of type assignment is that terms are assigned types that express what streams they can operate on; so, rather than stating $\lambda xy.xy : (A{\rightarrow}B){\rightarrow}A{\rightarrow}B$, for example, the system uses $\lambda xy.xy : (\delta\times\omega{\rightarrow}\rho)\times\delta\times\omega{\rightarrow}\rho$, expressing that it can take a stream containing at least two arguments, the first of type $\delta\times\omega{\rightarrow}\rho$ (stating it is a function that takes a stream as argument of which the first element is of type δ) and the second of type δ; the final ω in the product type here acts as an 'end of typed input' symbol. A direct consequence of taking the domain-directed approach to type assignment is that, naturally[1], intersection becomes a 'top level' type constructor, that lives at the same level as arrow, for example. This gives readable types and easy to understand type assignment rules, but also induces a type inclusion relation '$\leq$' and type assignment rule $(\leq)$ that complicate proofs and give a rather intricate and frankly rather unworkable generation lemma (see [12] for details and Section 10 below).

So, in view of what has been accomplished for the λ-calculus [2, 7], the natural question to ask is: is it possible to define a *strict* version of this notion, that is closed for conversion as well, and does without the contra-variant character of $\leq$, and hopefully would allow for more easily constructed proofs? We show here that this is the case: we will define such a strict version and show that it is closed for both subject reduction and expansion. The main restriction with respect to the system of [12] is limiting '$\leq$' on types to a relation that is no longer contra-variant, and allows only for the selection of a component of an intersection type.

Using this system, we will then focus on various characterisations of normalisation results,

[1] It is indeed tempting to see set intersection on the domain directly linked with the intersection type constructor, but, in fact, the similarity is, in the opinion of this author, misleading as it does not hold for the other set operators. For example, set-inclusion '$\subseteq$' is not as strongly linked to type inclusion $\leq$: not every subset of a set interpreted as a type will yield a subtype, but only those created through intersection.

as already shown for the λ-calculus in a collection of papers. For example, Barendregt, Coppo and Dezani-Ciancaglini [18] have shown that cut-elimination is normalising, which leads to characterisation of (head)-normalisation through assignable types; Ronchi and Venneri [36] have shown the approximation result, and Pottinger [34] showed a characterisation of the strongly normalisable terms. The author has shown these results for the strict intersection system [2] in a series of papers, summarised in [7]. Here we will show all these results for the notion of strict negated intersection type assignment for $\lambda\mu$ that is defined in this paper.

Outline of this paper:

In Section 1 we will give a quick overview of Parigot's $\lambda\mu$-calculus [33], for which in Section 2 we will present the notion of intersection type assignment of Bakel-Barbanera-Liguoro-LMCS [12], and state some of the properties of that system that are relevant to this paper. This notion is non-standard in that it contains no type-variables and is directly based on Streicher and Reus's [38] 'negated' domain construction. In Section 3 we then will present a strict variant of the notion of Section 2 where we essentially simplify the type language and remove the contra-variant $\leq$-relation, and in Section 4 will show that this system is closed under conversion and gives a semantics for $\lambda\mu$ in Theorem 4.5. Then in Section 5 we will define a formal notion of derivation reduction, which follows term reduction and is a kind of cut-elimination; in Section 6, Theorem 6.6, we will show that this notion of reduction is strongly normalisable.

In Section 7 we will define a notion of approximation for $\lambda\mu$-terms, and show that these can be used to define a semantics for $\lambda\mu$ as stated in Theorem 7.5. In Section 8, Theorem 8.4, we will show that, for every $\lambda\mu$-term that is typeable in the strict system, there exists an approximant that can be assigned the same type; this result follows directly from the fact that derivation reduction is strongly normalisable, as stated in Theorem 6.6. This result then leads to a characterisation of head normalisation using assignable types in Theorem 8.5.

In Section 9 we will remove the type constant ω from our system, essentially no longer permitting untyped terms; using this system, we will give a characterisation of normalisation in Theorem 9.8. In Theorem 9.12 we will show that in this restricted system, all terms are typeable if and only if they are strongly normalisable; also this result follows directly from Theorem 6.6. In Section 10 we then will compare the notions of type assignment we define here and that of [12].

One particular property of the system of [12], and the one we present here, is that both are type-variable free. In Section 11 we will investigate what the effect is of adding type variables to the type language; we will show that this in fact brings a notion of strict type assignment that is almost identical to that of [6], but for the fact that here negated types are used, and intersection of continuation types is there expressed through union.

Note: The results presented in sections 7, 8, and 9 are based on results presented in [9], although that paper dealt with a slightly different notion of type assignment. We will write $\underline{n}$ for the set $\{1,\ldots,n\}$ and use a vector notation for the abbreviation of sequences, so write $\vec{X}_{\underline{n}}$ for the sequence $X_1,\ldots,X_n$, and $\vec{X}$ if the number of elements in the sequence is not important.

1 The $\lambda\mu$-calculus

In this section we will present Parigot's pure $\lambda\mu$-calculus as introduced by Parigot [33]. It is an extension of the untyped λ-calculus obtained by adding *names* and a name-abstraction operator μ and was intended as a proof calculus for a fragment of classical logic. Logical formulas of the implicational fragment of the propositional calculus can be assigned as types to $\lambda\mu$-terms much in the formulae-as-types paradigm of the Curry-Howard correspondence between typed λ-calculus and intuitionistic logic.

Derivable statements have the shape $\Gamma \vdash M : A \mid \Delta$, where A is the main (*active*) conclusion of the statement of which M is the witness, and Δ contains the alternative conclusions, consisting of pairs of names and types; the left-hand context Γ, as usual, is a mapping from term variables to types, and represents the assumptions about free variables of M.

Definition 1.1 (Term Syntax [33]) The *terms* of $\lambda\mu$ are defined by the grammar (where $x,y,\ldots$ range over *term variables*, and $\alpha,\beta,\ldots$ over *names*):

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

As usual, we consider λ and μ to be binders; the sets $fv(M)$ and $fn(M)$ of, respectively, *free variables* and *free names* in a term M are defined in the usual way. We adopt Barendregt's convention on terms, so will assume that free and bound variables and names are different.

To explain the difference between variables and names, $\lambda\mu$ inherits from the λ-calculus the fact that term variables act as destinations of operands during reduction, in that in the contraction of a β-redex $(\lambda x.M)N$, the term N will take the place of all occurrences of x in M. Names on the other hand act as pointers to sub-terms, and the contraction of the μ-redex $(\mu\alpha.[\beta]M)N$ will result in placing N *behind* every sub-term P in M that is named α (so $\mu\gamma.[\alpha]P$ is a sub-term of M, for some γ), making N an operand to P. So β-reduction is the normal functional computational step, whereas μ-reduction essentially is a re-distribution of terms.

Definition 1.2 (Substitution [33]) Substitution takes two forms:

$$\begin{array}{lll} \textit{term substitution:} & M\{N/x\} & (N \text{ is substituted for } x \text{ in } M) \\ \textit{structural substitution:}^{2} & M\{L{\cdot}\gamma/\alpha\} & (\text{every command of the shape } [\alpha]N \text{ in } M \\ & & \text{is replaced by } [\gamma]NL) \end{array}$$

More precisely, $M\{L{\cdot}\gamma/\alpha\}$ is defined by:

$$\begin{array}{rcll@{\qquad}rcl} ([\alpha]P)\{L{\cdot}\gamma/\alpha\} & \triangleq & [\gamma]P\{L{\cdot}\gamma/\alpha\}L & & x\{L{\cdot}\gamma/\alpha\} & \triangleq & x \\ ([\delta]P)\{L{\cdot}\gamma/\alpha\} & \triangleq & [\delta]P\{L{\cdot}\gamma/\alpha\} & (\alpha \neq \delta) & (\lambda x.P)\{L{\cdot}\gamma/\alpha\} & \triangleq & \lambda x.P\{L{\cdot}\gamma/\alpha\} \\ (\mu\beta.\mathtt{C})\{L{\cdot}\gamma/\alpha\} & \triangleq & \mu\beta.\mathtt{C}\{L{\cdot}\gamma/\alpha\} & & (PQ)\{L{\cdot}\gamma/\alpha\} & \triangleq & (P\{L{\cdot}\gamma/\alpha\})(Q\{L{\cdot}\gamma/\alpha\}) \end{array}$$

Both substitutions are capture avoiding, using renaming of bound variable or names (α-conversion) when necessary.

Notice that, in the third alternative, since our intention is to substitute the free occurrences of α in $\mu\beta.\mathtt{C}$, by Barendregt's convention we can assume $\beta \neq \alpha$.

We will write $M\{\overrightarrow{N_n}{\cdot}\gamma/\alpha\}$ for $M\{N_1{\cdot}\gamma_1/\alpha\}\{N_2{\cdot}\gamma_2/\gamma_1\}\cdots\{N_n{\cdot}\gamma/\gamma_{n-1}\}$.

Definition 1.3 ($\lambda\mu$-reduction [33]) *i*) Reduction in $\lambda\mu$ is based on the following rules:

$$\begin{array}{rrcll} (\beta): & (\lambda x.M)N & \to & M\{N/x\} & (\textit{logical reduction}) \\ (\mu): & (\mu\beta.\mathtt{C})Q & \to & \mu\gamma.\mathtt{C}\{Q{\cdot}\gamma/\beta\}Q & (\textit{structural reduction})^{3} \\ (\textsc{Ren}): & [\beta]\mu\gamma.\mathtt{C} & \to & \mathtt{C}\{\beta/\gamma\} & (\textit{renaming}) \end{array}$$

[2] Since this operation does not substitute one syntactic structure by another but rather inserts a term in a precise manner, the name 'substitution' is perhaps misleading here, and 'insertion' would be better. We will use the latter terminology later in Definition 5.3.

ii) We write '$\rightarrow_{\beta\mu}$' for the reduction relation that is the compatible closure of these rules, '$\rightarrow^*_{\beta\mu}$' for its transitive closure, and '$=_{\beta\mu}$' for the equivalence relation generated by it.

iii) We can also consider the two extensional rules:

$$\begin{array}{rrcll} (\eta): & \lambda x.Mx & \rightarrow & M & (x \notin fv(M)) \\ (\mu\eta): & \mu\alpha.[\alpha]M & \rightarrow & M & (\alpha \notin fn(M)) \end{array}$$

It is possible to formulate more extensional rules, but we will not consider those in this paper; the equivalent of Theorem 4.5 could not be shown to hold for those rules. Confluence for this notion of reduction has been shown by Py [35].

Below we will need the concept of head-normal form for $\lambda\mu$, which is defined as follows:

Definition 1.4 (Head-normal forms) The $\lambda\mu$ *head-normal forms* (with respect to $\rightarrow_{\beta\mu}$) are defined through the grammar:

$$\begin{array}{rcll} \boldsymbol{H} & ::= & xM_1\cdots M_n & (n \geq 0) \\ & \mid & \lambda x.\boldsymbol{H} & \\ & \mid & \mu\alpha.[\beta]\boldsymbol{H} & (\boldsymbol{H} \neq \mu\gamma.[\delta]\boldsymbol{H}') \end{array}$$

Standard type assignment for $\lambda\mu$ is defined by:

Definition 1.5 (Classical Typing for $\lambda\mu$) *i*) The types for $\lambda\mu$ are defined through:

$$A,B \ ::= \ \varphi \mid A{\rightarrow}B$$

ii) A *variable context* Γ is a partial mapping from term variables to types, denoted as a finite set of *statements* $x{:}A$, such that the *subject* of the statements (x) are distinct.

iii) We write $\Gamma,x{:}A$ for the context defined by:

$$\begin{array}{rcll} \Gamma,x{:}A & \triangleq & \Gamma \cup \{x{:}A\}, & \text{if } \Gamma \text{ is not defined on } x \\ & \triangleq & \Gamma, & \text{if } x{:}A \in \Gamma \end{array}$$

We write $x \notin \Gamma$ when there exists no type A such that $x{:}A \in \Gamma$.

iv) *Name contexts* Δ as partial mappings from names to types and the notions $\alpha{:}\kappa,\Delta$ and $\alpha \notin \Delta$ are defined in a similar way.

v) The type assignment rules are:

$$(Ax): \frac{}{\Gamma,x{:}A \vdash x : A \mid \Delta} \qquad (\mu): \frac{\Gamma \vdash M : B \mid \alpha{:}A,\beta{:}B,\Delta}{\Gamma \vdash \mu\alpha.[\beta]M : A \mid \beta{:}B,\Delta}\ (\alpha \notin \Delta) \quad \frac{\Gamma \vdash M : A \mid \alpha{:}A,\Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : A \mid \Delta}\ (\alpha \notin \Delta)$$

$$({\rightarrow}I): \frac{\Gamma,x{:}A \vdash M : B \mid \Delta}{\Gamma \vdash \lambda x.M : A{\rightarrow}B \mid \Delta}\ (x \notin \Gamma) \qquad ({\rightarrow}E): \frac{\Gamma \vdash M : A{\rightarrow}B \mid \Delta \quad \Gamma \vdash N : A \mid \Delta}{\Gamma \vdash MN : B \mid \Delta}$$

We write $\Gamma \vdash_{\lambda\mu} M : A \mid \Delta$ for judgements derivable in this system.

We can think of $\mu\alpha.[\beta]M$ as a *context switch* or *redirection*; it stores the type of M amongst the alternative conclusions by giving it the name β, and redirects the operands to the terms called α in M.

Throughout this paper, we will extend Barendregt's convention to judgements $\Gamma \vdash_{\lambda\mu} M : \delta \mid \Delta$ by seeing the variables that occur in Γ and names in Δ as binding occurrences over M as well,

[3] A more common notation for this rule would be $(\mu\alpha.[\delta]M)N \rightarrow \mu\alpha.[\delta]M\{N/\alpha\}$, using the fact that α disappears during reduction, and can be picked as name for the newly created applications instead of γ. But, in fact, this is not the same α (as the named term has changed), as reflected in the fact that its type changes during reduction; see also Example 2.6, where before the reduction $(\mu\alpha.[\alpha]x)x \rightarrow \mu\gamma.[\gamma]xx$, α has type $\delta \times \kappa$, and after γ has type κ.

$$
\dfrac{\dfrac{}{x{:}(A\to B)\to A \vdash x : (A\to B)\to A \mid \alpha{:}A}\ (Ax) \quad \dfrac{\dfrac{\dfrac{}{x{:}(A\to B)\to A, y{:}A \vdash y : A \mid \alpha{:}A, \beta{:}B}\ (Ax)}{x{:}(A\to B)\to A, y{:}A \vdash \mu\beta.[\alpha]y : B \mid \alpha{:}A}\ (\mu)}{x{:}(A\to B)\to A \vdash \lambda y.\mu\beta.[\alpha]y : A\to B \mid \alpha{:}A}\ (\to I)}{\dfrac{\dfrac{x{:}(A\to B)\to A \vdash x(\lambda y.\mu\beta.[\alpha]y) : A \mid \alpha{:}A}{x{:}(A\to B)\to A \vdash \mu\alpha.[\alpha](x(\lambda y.\mu\beta.[\alpha]y)) : A \mid \emptyset}\ (\mu)}{\vdash \lambda x.\mu\alpha.[\alpha](x(\lambda y.\mu\beta.[\alpha]y)) : ((A\to B)\to A)\to A \mid \emptyset}\ (\to I)}\ (\to E)
$$

Figure 1: A derivation for a term representing Peirce's Law in $\vdash_{\lambda\mu}$

for all notions of type assignment; in particular, we can assume that no variable in Γ nor name in Δ is bound in M.

Example 1.6 As an example illustrating the fact that this system is more powerful than the system for the λ-calculus, Figure 1 shows that it is possible to inhabit Peirce's Law (due to Ong-Stewart [32]); the underlying logic of the system of Definition 1.5 corresponds to *minimal classical logic* [1].

2 The intersection type assignment system for $\lambda\mu$

In [12], a filter model was presented for $\lambda\mu$; but instead of defining a suitable type system for $\lambda\mu$ and then proving that it actually induces a filter model, as was done in [18, 2], that paper followed the opposite route. As mentioned in [12]: "It emerged in [22] that models constructed as set of filters of intersection types are exactly the ω-algebraic lattices, a category of complete lattices, but with Scott-continuous maps as morphisms. ω-algebraic lattices are posets whose structure is fully determined by a countable subset of elements, called 'compact points' for topological reasons. Now the crucial fact is that given an ω-algebraic lattice D, the set $Compact(D)$ of its compact points can be described by putting its elements into a one-to-one correspondence with a suitable set of intersection types, in such a way that the order over $Compact(D)$ is reflected by the inverse of the $\leq$ pre-order over types. Then one can show that the filter structure $\mathcal{F}_D$ obtained from the type pre-order is isomorphic with the original D." Starting from Streicher and Reus's [38] *models of continuations* of the $\lambda\mu$-calculus , the authors extracted the type syntax and the corresponding type theory out of the construction of the model, a solution of the 'negated' domain equations $D = C \to R$ and $C = D \times C$, where R is an arbitrary domain of 'results'. Here C is a set of what are called 'continuations', which are infinite tuples of elements in D, which is the domain of continuous functions from C to R and is the set of 'denotations' of terms. The syntax of types follows this construction closely: $\lambda\mu$-terms are separated into terms of type δ and sequences of terms (*streams*, or *stacks*, of the shape $L_1 :: L_2 :: \cdots :: L_n$). Streams are not syntactical entities themselves, but are considered to be the semantics for continuation types $\kappa = \delta_1 \times \cdots \times \delta_n \times \omega$, where ω acts as an 'end of sequence' symbol; the types $\delta_1,\ldots,\delta_n$ are types for the first n relevant terms in the stream, and ω is that for the non-relevant tail.

Definition 2.1 ([12]) *i)* $\mathcal{T}_D$ and $\mathcal{T}_C$ are the sets of intersection types defined by the grammar:

$$
\begin{array}{llcll}
\mathcal{T}_R: & \rho & ::= & v_a \mid \omega \mid \rho \wedge \rho & (a \in Compact(R)) \\
\mathcal{T}_D: & \delta & ::= & \kappa \to \rho \mid \omega \mid \delta \wedge \delta & (\textit{term types}) \\
\mathcal{T}_C: & \kappa & ::= & \delta \times \kappa \mid \omega \mid \kappa \wedge \kappa & (\textit{continuation types})
\end{array}
$$

We let σ, τ range over $\mathcal{T}_D \cup \mathcal{T}_C$ and assume '$\wedge$' to bind more strongly than '$\times$', and '$\times$' more strongly than '$\rightarrow$'.

ii) The type inclusion relations '$\leq_\wedge$' and '$\sim_\wedge$' are defined as the smallest pre-orders satisfying:

$$\frac{}{\sigma \wedge \tau \leq_\wedge \sigma} \quad \frac{}{\sigma \wedge \tau \leq_\wedge \tau} \quad \frac{}{\omega \leq_\wedge \omega \rightarrow \omega} \quad \frac{}{\upsilon \leq_\wedge \omega \rightarrow \upsilon} \quad \frac{}{\sigma \leq_\wedge \omega} \quad \frac{}{\omega \rightarrow \upsilon \leq_\wedge \upsilon} \quad \frac{}{\omega \leq_\wedge \omega \times \omega}$$

$$\frac{}{(\kappa \rightarrow \delta_1) \wedge (\kappa \rightarrow \delta_2) \leq_\wedge \kappa \rightarrow (\delta_1 \wedge \delta_2)} \quad \frac{}{(\delta_1 \times \kappa_1) \wedge (\delta_2 \times \kappa_2) \leq_\wedge (\delta_1 \wedge \delta_2) \times (\kappa_1 \wedge \kappa_2)}$$

$$\frac{\rho \leq_\wedge \sigma \quad \rho \leq_\wedge \tau}{\rho \leq_\wedge \sigma \wedge \tau} \quad \frac{\kappa_2 \leq_\wedge \kappa_1 \quad \rho_1 \leq_\wedge \rho_2}{\kappa_1 \rightarrow \rho_1 \leq_\wedge \kappa_2 \rightarrow \rho_2} \quad \frac{\delta_1 \leq_\wedge \delta_2 \quad \kappa_1 \leq_\wedge \kappa_2}{\delta_1 \times \kappa_1 \leq_\wedge \delta_2 \times \kappa_2} \quad \frac{\sigma \leq_\wedge \tau \quad \tau \leq_\wedge \sigma}{\sigma \sim_\wedge \tau}$$

iii) Much as in Definition 1.5, a *variable context* Γ is a mapping from term variables to types in $\mathcal{T}_D$, presented as a set, and we define $\Gamma, x{:}\delta$ and $x \in \Gamma$ as before.

iv) We extend the relation '$\leq_\wedge$' to variable contexts by:

$$\Gamma_1 \leq_\wedge \Gamma_2 \;\triangleq\; \forall x{:}\delta_2 \in \Gamma_2\; \exists x{:}\delta_1 \in \Gamma_1\; [\delta_1 \leq_\wedge \delta_2]$$

v) *Name contexts* Δ and the notions $\alpha{:}\kappa, \Delta$, $\alpha \in \Delta$, and $\Delta_1 \leq_\wedge \Delta_2$ are defined in a similar way.

As the domains the type theory is based on are 'negated', so are the types; we will come back to that in Section 3. Notice that ω is used in two different ways: to mark the end of a stream-type, and as the type used for terms that are ignored; we consider ω in the second use a 'proper' type.

The '$\leq_\wedge$'-relation as defined above is the usual one on arrow types, contra-variant in the first argument and co-variant in the second. It is straightforward to show that $\omega \sim_\wedge \omega \rightarrow \omega$, $\omega \sim_\wedge \omega \times \omega$, $(\kappa \rightarrow \rho_1) \wedge (\kappa \rightarrow \rho_2) \sim_\wedge \kappa \rightarrow (\rho_1 \wedge \rho_2)$, and $(\delta_1 \times \kappa_1) \wedge (\delta_2 \times \kappa_2) \sim_\wedge (\delta_1 \wedge \delta_2) \times (\kappa_1 \wedge \kappa_2)$.

Definition 2.2 ([12]) Intersection type assignment for $\lambda\mu$ is defined through the following rules (where T ranges over both terms and commands):

$$(Ax): \frac{}{\Gamma, x{:}\delta \vdash x : \delta \mid \Delta} \qquad (\omega): \frac{}{\Gamma \vdash T : \omega \mid \Delta}$$

$$(Abs): \frac{\Gamma, x{:}\delta \vdash M : \kappa \rightarrow \rho \mid \Delta}{\Gamma \vdash \lambda x.M : \delta \times \kappa \rightarrow \rho \mid \Delta}\ (x \notin \Gamma) \qquad (Cmd): \frac{\Gamma \vdash M : \delta \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash [\alpha]M : \delta \times \kappa \mid \alpha{:}\kappa, \Delta}$$

$$(App): \frac{\Gamma \vdash M : \delta \times \kappa \rightarrow \rho \mid \Delta \quad \Gamma \vdash N : \delta \mid \Delta}{\Gamma \vdash MN : \kappa \rightarrow \rho \mid \Delta} \qquad (\mu): \frac{\Gamma \vdash \mathsf{C} : (\kappa' \rightarrow \rho) \times \kappa' \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash \mu\alpha.\mathsf{C} : \kappa \rightarrow \rho \mid \Delta}\ (\alpha \notin \Delta)$$

$$(\wedge): \frac{\Gamma \vdash T : \sigma \mid \Delta \quad \Gamma \vdash T : \tau \mid \Delta}{\Gamma \vdash T : \sigma \wedge \tau \mid \Delta} \qquad (\leq_\wedge): \frac{\Gamma \vdash T : \sigma \mid \Delta \quad \sigma \leq_\wedge \tau}{\Gamma \vdash T : \tau \mid \Delta}$$

We write $\Gamma \vdash_\wedge T : \sigma \mid \Delta$ for judgements derivable in this system.

Notice that, in (Abs), the type $\kappa \rightarrow \rho$ is *not* a subtype of $\delta \times \kappa \rightarrow \rho$.

These typing rules are direct interpretations from the clauses that define a term interpretation into the filter model $\mathcal{F}_D$ (see [12] for details). An earlier version of the system, as presented in [10], allowed υ as a term type as well; as this created unwanted effects (it is possible to derive $x{:}\upsilon \vdash_\wedge x : \upsilon \mid$, but (Abs) cannot be applied to this), this was abolished in [12].

Example 2.3 To illustrate the interaction between the rules, take $\delta' = \delta \wedge (\delta \times \kappa \rightarrow \rho)$ and $\kappa'' = \kappa \wedge (\delta' \times \kappa')$. We can derive $\emptyset \vdash_{\mathrm{S}} \mu\gamma.[\gamma]\lambda x.\mu\alpha.[\gamma]xx : \kappa'' \rightarrow \rho \mid \emptyset$, as shown in Figure 2.

There is one feature of this system that is perhaps worth pointing out. Continuation types all end in ω, allowing for a feature that is not present in other notions of type assignment. The philosophy of the system is that continuation types $\delta_1 \times \cdots \delta_n \times \omega$ are types of (possible infinite) streams of terms, of which at most the types of the first n are relevant for the type

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(Ax)}{x{:}\delta' \vdash x : \delta \times \kappa \to \rho \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(\leq_\wedge) \qquad \dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(Ax)}{x{:}\delta' \vdash x : \delta \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(\leq_\wedge)}{x{:}\delta' \vdash xx : \kappa \to \rho \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(App)}{x{:}\delta' \vdash [\gamma]xx : \kappa \to \rho \times \kappa'' \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(Cmd)}{x{:}\delta' \vdash [\gamma]xx : \kappa \to \rho \times \kappa \mid \alpha{:}\kappa',\gamma{:}\kappa''}\;(\leq_\wedge)}{x{:}\delta' \vdash \mu\alpha.[\gamma]xx : \kappa' \to \rho \mid \gamma{:}\kappa''}\;(\mu)}{\emptyset \vdash \lambda x.\mu\alpha.[\gamma]xx : \delta' \times \kappa' \to \rho \mid \gamma{:}\kappa''}\;(Abs)}{\emptyset \vdash [\gamma]\lambda x.\mu\alpha.[\gamma]xx : (\delta' \times \kappa' \to \rho) \times (\kappa'') \mid \gamma{:}\kappa''}\;(Cmd)}{\emptyset \vdash [\gamma]\lambda x.\mu\alpha.[\gamma]xx : (\delta' \times \kappa' \to \rho) \times \delta' \times \kappa' \mid \gamma{:}\kappa''}\;(\leq_\wedge)}{\emptyset \vdash \mu\gamma.[\gamma]\lambda x.\mu\alpha.[\gamma]xx : \kappa'' \to \rho \mid \emptyset}\;(\mu)$$

Figure 2: A derivation for $\emptyset \vdash_{\mathrm{S}} \mu\gamma.[\gamma]\lambda x.\mu\alpha.[\gamma]xx : \kappa'' \to \rho \mid \emptyset$

assignment of the term under consideration, and ω is used to cover the remainder.

Example 2.4 Assume we have derivations for $\Gamma \vdash M : (\delta_1 \times \omega) \to \rho \mid \alpha{:}\delta_1 \times \omega, \Delta$, and $\Gamma \vdash N_i : \delta_i \mid \Delta$, for $i \in \underline{2}$. Then we can construct (in '$\vdash_\wedge$'):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash M : (\delta_1 \times \omega) \to \rho \mid \alpha{:}\delta_1 \times \omega, \Delta}{\Gamma \vdash [\alpha]M : ((\delta_1 \times \omega) \to \rho) \times (\delta_1 \times \omega) \mid \alpha{:}\delta_1 \times \omega, \Delta}}{\Gamma \vdash \mu\alpha.[\alpha]M : (\delta_1 \times \omega) \to \rho \mid \Delta} \qquad \Gamma \vdash N_1 : \delta_1 \mid \Delta}{\Gamma \vdash (\mu\alpha.[\alpha]M)N_1 : \omega \to \rho \mid \Delta}}{\Gamma \vdash (\mu\alpha.[\alpha]M)N_1 : (\delta_2 \times \delta_3 \times \omega) \to \rho \mid \Delta}\;(\leq_\wedge) \qquad \Gamma \vdash N_2 : \delta_2 \mid \Delta}{\Gamma \vdash (\mu\alpha.[\alpha]M)N_1N_2 : (\delta_3 \times \omega) \to \rho \mid \Delta}$$

Notice that the type for α only expresses that terms named α can take one argument (of type δ_1), and that, through $(\leq_\wedge)$, we can 'pump that up' to more terms.

This feature is convenient below, since it allows us to not have to formally define intersections of continuations types, and only use those as short-hand notation for the 'zipping up' of continuation types (see the remark after Definition 3.1). It is rather obsolete, however, when trying to type a term in '$\vdash_\wedge$', since we could have started the derivation with $\alpha{:}\delta_1 \times \delta_2 \times \delta_3 \times \omega$ from the beginning; notice that the *only* characteristic of the type $\delta_1 \times \omega$ needed for α when typing M is that it *starts* with δ_1, which is of course also true for $\delta_1 \times \delta_2 \times \delta_3 \times \omega$. So we can construct:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash M : \delta_1 \times \omega \to \rho \mid \alpha{:}\delta_1 \times \delta_2 \times \delta_3 \times \omega, \Delta}{\Gamma \vdash [\alpha]M : (\delta_1 \times \omega \to \rho) \times (\alpha{:}\delta_1 \times \delta_2 \times \delta_3 \times \omega) \mid \alpha{:}\delta_1 \times \delta_2 \times \delta_3 \times \omega, \Delta}}{\Gamma \vdash [\alpha]M : (\delta_1 \times \omega \to \rho) \times (\delta_1 \times \omega) \mid \alpha{:}\delta_1 \times \delta_2 \times \delta_3 \times \omega, \Delta}\;(\leq_\wedge)}{\Gamma \vdash \mu\alpha.[\alpha]M : \delta_1 \times \delta_2 \times \delta_3 \times \omega \to \rho \mid \Delta} \qquad \Gamma \vdash N_1 : \delta_1 \mid \Delta}{\Gamma \vdash (\mu\alpha.[\alpha]M)N_1 : \delta_2 \times \delta_3 \times \omega \to \rho \mid \Delta} \qquad \Gamma \vdash N_2 : \delta_2 \mid \Delta}{\Gamma \vdash (\mu\alpha.[\alpha]M)N_1N_2 : \delta_3 \times \omega \to \rho \mid \Delta}$$

For this system, [12] shows a series of results that confirm the validity of the construction, like that assignable types are invariant under conversion (both under reduction and expansion, but only with respect to β and μ reduction, so not for renaming), both directly and through the filter model and semantics. It also gives a characterisation of strong normalisation for a subsystem that limits the use of the type constant ω.

Theorem 2.5 ([12]) *i) Let $\mathcal{M}$ be a $\lambda\mu$-model, and $\models_{\mathcal{M}}$ stand for semantic satisfiability in $\mathcal{M}$. Then: $\Gamma \vdash T : \sigma \mid \Delta$, if and only if $\Gamma \models_{\mathcal{M}} T : \sigma \mid \Delta$.*

ii) Let $M \rightarrow_\beta N$. Then: $\Gamma \vdash_\wedge M : \delta \mid \Delta$ if and only if $\Gamma \vdash_\wedge N : \delta \mid \Delta$.

iii) Let $M \rightarrow_\mu N$. Then: $\Gamma \vdash_\wedge M : \delta \mid \Delta$ if and only if $\Gamma \vdash_\wedge N : \delta \mid \Delta$.

iv) There exist Γ, Δ and δ such that $\Gamma \vdash_\wedge M : \delta \mid \Delta$ without using rule (ω) at all in the derivation, if and only if M is strongly normalising.

Some of the proofs in [12] are complicated through the fact that type assignment is defined also via type assignment to named terms, and that the '$\leq$'-relation is contra-variant. In particular, analysing the structure of a derivation through the generation lemma (see Lemma 10.3), is intricate (see also the proof of Lemma 10.4).

Example 2.6 ([12]) Take the reduction $(\mu\alpha.[\alpha]x)x \rightarrow \mu\gamma.[\gamma]xx$. The latter term contains self application which we can type using intersection types. Let $\delta' = \delta \wedge (\delta \times \kappa \rightarrow \rho)$, then in '$\vdash_\wedge$' we have both:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \alpha{:}\delta\times\kappa}\,(Ax)}{x{:}\delta' \vdash x : \delta\times\kappa\rightarrow\rho \mid \alpha{:}\delta\times\kappa}\,(\leq)}{x{:}\delta' \vdash [\alpha]x : (\delta\times\kappa\rightarrow\rho)\times(\delta\times\kappa) \mid \alpha{:}\delta\times\kappa}\,(Cmd)}{x{:}\delta' \vdash \mu\alpha.[\alpha]x : \delta\times\kappa\rightarrow\rho \mid}\,(\mu) \quad \dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \alpha{:}\delta\times\kappa}\,(Ax)}{x{:}\delta' \vdash x : \delta \mid \alpha{:}\delta\times\kappa}\,(\leq)}{x{:}\delta' \vdash (\mu\alpha.[\alpha]x)x : \kappa\rightarrow\rho \mid}\,(App)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \gamma{:}\kappa}\,(Ax)}{x{:}\delta' \vdash x : \delta\times\kappa\rightarrow\rho \mid \gamma{:}\kappa}\,(\leq) \quad \dfrac{\dfrac{}{x{:}\delta' \vdash x : \delta' \mid \gamma{:}\kappa}\,(Ax)}{x{:}\delta' \vdash x : \delta \mid \gamma{:}\kappa}\,(\leq)}{x{:}\delta' \vdash xx : \kappa\rightarrow\rho \mid \gamma{:}\kappa}\,(App)}{x{:}\delta' \vdash [\gamma]xx : (\kappa\rightarrow\rho)\times\kappa \mid \gamma{:}\kappa}\,(Cmd)}{x{:}\delta' \vdash \mu\gamma.[\gamma]xx : \kappa\rightarrow\rho \mid}\,(\mu)$$

Observe that the 'cut type' in the first derivation, $\delta\times\kappa$ (appearing twice in the type $(\delta\times\kappa\rightarrow\rho)$ $\times(\delta\times\kappa)$ of the premise of (μ)), differs from the cut type κ in $(\kappa\rightarrow\rho)\times\kappa$ occurring in the premise of (μ) of the second derivation; indeed, the latter is of a smaller size than the former.

Although formally defined on the variant $\Lambda\mu$ defined by de Groote [26], where naming and μ-binding are separated, many of the results in [12] are only shown for $\lambda\mu$. In fact, commands are pseudo terms, and other than having their origin in the filter model, it is not easy to give an intuitive explanation for rules (Cmd) and (μ). However, it is possible to show that the rules

$$\dfrac{\Gamma \vdash M : \kappa'\rightarrow\rho \mid \alpha{:}\kappa, \beta{:}\kappa', \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \kappa\rightarrow\rho \mid \beta{:}\kappa', \Delta} \qquad \dfrac{\Gamma \vdash M : \kappa\rightarrow\rho \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \kappa\rightarrow\rho \mid \Delta}$$

are admissible, which more closely correspond to rules (μ) of $\vdash_{\lambda\mu}$, and effectively express the redirection behaviour. Since more related to the definitions that follow, we show the admissibility of more generalised versions of these rules instead:

Lemma 2.7 The following two rules are admissible in '$\vdash_\wedge$':

$$(\mu_1) : \ \dfrac{\Gamma \vdash M : \kappa'\rightarrow\rho \mid \alpha{:}\kappa'', \beta{:}\kappa, \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \kappa''\rightarrow\rho \mid \beta{:}\kappa, \Delta}\ (\kappa \leq_\wedge \kappa') \qquad (\mu_2) : \ \dfrac{\Gamma \vdash M : \kappa'\rightarrow\rho \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \kappa\rightarrow\rho \mid \Delta}\ (\kappa \leq_\wedge \kappa')$$

Proof:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash M : \kappa'\rightarrow\rho \mid \alpha{:}\kappa'', \beta{:}\kappa, \Delta}{\Gamma \vdash M : \kappa\rightarrow\rho \mid \alpha{:}\kappa'', \beta{:}\kappa, \Delta}\,(\kappa \leq_\wedge \kappa')}{\Gamma \vdash [\beta]M : (\kappa\rightarrow\rho)\times\kappa \mid \alpha{:}\kappa'', \beta{:}\kappa, \Delta}\,(Cmd)}{\Gamma \vdash \mu\alpha.[\beta]M : \kappa''\rightarrow\rho \mid \beta{:}\kappa, \Delta}\,(\mu) \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash M : \kappa'\rightarrow\rho \mid \alpha{:}\kappa, \Delta}{\Gamma \vdash M : \kappa\rightarrow\rho \mid \alpha{:}\kappa, \Delta}\,(\kappa \leq_\wedge \kappa')}{\Gamma \vdash [\alpha]M : (\kappa\rightarrow\rho)\times\kappa \mid \alpha{:}\kappa, \Delta}\,(Cmd)}{\Gamma \vdash \mu\alpha.[\alpha]M : \kappa\rightarrow\rho \mid \Delta}\,(\mu)$$

Notice that these rules include the two mentioned above, taking $\kappa' = \kappa$.

3 Strict type assignment with negated intersection types

In this section we will define '$\vdash_S$', a notion of type assignment that corresponds to a restriction of the system of Bakel-Barbanera-Liguoro-LMCS [12], where we limit the occurrence of intersections to only appear as components of continuation types (so do not allow intersections *of* continuation types), and do not allow the right-hand part of an arrow type to be an intersection. We will also no longer use ω both to mark the end of a continuation type, and for untyped terms as is traditional, but rather use Ω for the first, and ω for the latter. But, more importantly, we remove the general inference rule $(\leq)$, and change the rules to explicitly state when a '$\leq$'-step is allowed, as in (Ax). We also limit '$\leq$', essentially, to only allow the selection of a component in an intersection type and remove the contra-variant character. This will greatly facilitate proofs, and allow for a more comprehensible presentation and treatment. We will investigate the relation between '$\vdash_\wedge$' and '$\vdash_S$' in Section 10.

Moreover, as indicated above, we can see the types as negated by restricting the domain of results to one element v,[4] and from now on will write $\neg\kappa$ for $\kappa{\to}v$. Note that, using this notation, the notion '$\vdash_\wedge$' could be defined using the (perhaps more intuitive) rules

$$(Abs): \frac{\Gamma, x{:}\delta \vdash M : \neg\kappa \mid \Delta}{\Gamma \vdash \lambda x.M : \delta{\to}\neg\kappa \mid \Delta}\ (x \notin \Gamma) \qquad (App): \frac{\Gamma \vdash M : \delta{\to}\neg\kappa \mid \Delta \quad \Gamma \vdash N : \delta \mid \Delta}{\Gamma \vdash MN : \neg\kappa \mid \Delta}$$

However, notice that the conclusion type in (App) is now a negated type; in order to be able to use these rules again, we would then have to add the conversion rules

$$\frac{\Gamma \vdash M : \delta{\to}\neg\kappa \mid \Delta}{\Gamma \vdash M : \neg(\delta \times \kappa) \mid \Delta} \qquad \frac{\Gamma \vdash M : \neg(\delta \times \kappa) \mid \Delta}{\Gamma \vdash M : \delta{\to}\neg\kappa \mid \Delta}$$

which would unnecessarily complicate reasoning over the structure of derivations.

As already implicitly used in the rules above, as a logical formula, the type $A{\to}\neg B$ corresponds in classical logic to $\neg(A\&B)$. However, since logically $A\&B$ is the same as $B\&A$, but $A{\to}\neg B$ as a type does not correspond to $B{\to}\neg A$, the order of the components is important; it would therefore be better to use a non-commutative conjunction type constructor rather than '&'. The operator '$\times$', traditionally linked to pairing, serves this property nicely.

Irrespective of the changes mentioned above, we will show that it is still possible to show that type assignment is closed for conversion on terms, underlining that also for $\lambda\mu$ a contra-variant type inclusion relation is not needed to gain a sufficiently expressive notion of intersection type assignment.

Definition 3.1 (Strict Negated Types) We define *strict negated intersection types* (strict types for short) by the grammar:

$$\begin{array}{rcll} \mathtt{A},\mathtt{B} & ::= & \neg\mathtt{C} & (\textit{basic types}) \\ \mathtt{R},\mathtt{S},\mathtt{T} & ::= & \omega \mid \mathtt{A}_1 \cap \cdots \cap \mathtt{A}_n \quad (n \geq 1) & (\textit{intersection types}) \\ \mathtt{C},\mathtt{D} & ::= & \Omega \mid \mathtt{S} \times \mathtt{C} & (\textit{continuation types}) \end{array}$$

We let '$\cap$' bind more strongly than '$\times$', and '$\neg$' more than '$\cap$'.[5]

Reading $\neg\mathtt{C}$ as $\mathtt{C}{\to}v$ and Ω as ω, the set of types defined by the above grammar is a subset of that defined in Definition 2.1. Notice that, for any continuation type $\mathtt{C}$ there are $n \geq 0$, $\mathtt{S}_i$ $(i \in \underline{n})$ such that $\mathtt{C} = \mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \Omega$. For convenience, we will write $\cap_I \mathtt{A}_i$ for $\mathtt{A}_{i_1} \cap \cdots \cap \mathtt{A}_{i_n}$ where $I = \{ i_1, \ldots, i_n \}$, $\cap_\emptyset \mathtt{A}_i$ for ω, and $\cap_{\underline{n}} \mathtt{A}_i$ for $\mathtt{A}_1 \cap \ldots \cap \mathtt{A}_n$.

[4] In fact, as can be expected, the results of [12] are indifferent to the number of compact elements in R.

[5] Observe that implication, '$\to$', is no longer a type constructor here.

Definition 3.2 (Strict Type Inclusion) We define '$\leq_{\mathrm{S}}$' as the smallest partial order and '$\sim_{\mathrm{S}}$' as the smallest equivalence relation satisfying the rules below.

$$\frac{}{\cap_I A_i \leq A_j}\ (j \in I) \qquad \frac{S \leq A_i \quad (\forall i \in I)}{S \leq \cap_I A_i} \qquad \frac{}{\mathsf{C} \leq \Omega} \qquad \frac{S \leq T \quad \mathsf{C} \leq \mathsf{D}}{S \times \mathsf{C} \leq T \times \mathsf{D}}$$

$$\frac{S \leq_{\mathrm{S}} T \quad T \leq_{\mathrm{S}} S}{S \sim_{\mathrm{S}} T} \qquad \frac{\mathsf{C} \leq_{\mathrm{S}} \mathsf{D} \quad \mathsf{D} \leq_{\mathrm{S}} \mathsf{C}}{\mathsf{C} \sim_{\mathrm{S}} \mathsf{D}} \qquad \frac{\mathsf{C} \sim_{\mathrm{S}} \mathsf{D}}{\neg\mathsf{C} \sim_{\mathrm{S}} \neg\mathsf{D}}$$

We consider types modulo '$\sim_{\mathrm{S}}$'.

By abuse of notation, we will also write $S \cap T$, where $S = \cap_{\underline{n}} A_i$ and $T = \cap_{\underline{m}} B_j$, which represents the intersection type $A_1 \cap \cdots \cap A_n \cap B_1 \cap \cdots \cap B_m$. Moreover, below we will use intersections of continuation types (notice that these are not allowed under the formal definition of types) as a short-hand notation: let $\mathsf{C} = S_1 \times \cdots \times S_n \times \Omega$, and $\mathsf{D} = T_1 \times \cdots \times T_m \times \Omega$, and assume, without loss of generality, that $m > n$; then we define

$$\mathsf{C} \cap \mathsf{D} \ \triangleq\ S_1 \cap T_1 \times \cdots \times S_n \cap T_n \times T_{n+1} \times \cdots \times T_m \times \Omega.$$

(this operation is only really needed in sections 8 and 9). Notice that then $\mathsf{C} \cap \mathsf{D} \leq \mathsf{D}$ and $\mathsf{C} \cap \mathsf{D} \leq \mathsf{C}$.

The following properties over '$\leq_{\mathrm{S}}$' hold directly by definition:

Proposition 3.3 *i) For all* S, $S \leq_{\mathrm{S}} \omega$.

ii) If $S \leq_{\mathrm{S}} T$, *then* $S = \cap_I A_i$, $T = \cap_J B_j$, *and for every* $j \in J$ *there exists* $i \in I$ *such that* $A_i = B_j$.

iii) If $\mathsf{C} \leq_{\mathrm{S}} \mathsf{D}$, *then* $\mathsf{D} = \Omega$, *or* $\mathsf{C} = S \times \mathsf{C}'$, $\mathsf{D} = T \times \mathsf{D}'$, *and* $S \leq_{\mathrm{S}} T$ *and* $\mathsf{C}' \leq_{\mathrm{S}} \mathsf{D}'$.

We now come to the definition of strict type assignment. It essentially follows the approach of Definition 2.2, but for the fact that: *i) strict types are used; ii)* $(\leq_\wedge)$ *has been removed; iii)* '$\leq_{\mathrm{S}}$' *is part of rules* (Ax), (μ_2), *and* (μ_1); and *iv) rules* (ω) *and* $(\cap)$ *are joined, using the fact that we see* ω *as an empty intersection.* Moreover, we only assign types to terms, so drop the separate treatment of commands.

Definition 3.4 (Strict Type Assignment) *i*) A *variable context* Γ is a partial mapping from term variables to strict types; the notion $\Gamma, x{:}S$ is defined as before. *Name contexts* Δ as partial mappings from names to continuation types and the notion $\alpha{:}\mathsf{C}, \Delta$ are defined as before.

ii) As above, the relation '$\leq_{\mathrm{S}}$' is naturally extended to variable contexts as follows:

$$\Gamma' \leq_{\mathrm{S}} \Gamma \ \triangleq\ \forall x{:}S \in \Gamma\ \exists x{:}T \in \Gamma'\ [T \leq_{\mathrm{S}} S];$$

$\Delta' \leq_{\mathrm{S}} \Delta$ is defined similarly.

iii) *Strict type assignment* for $\lambda\mu$-terms is defined by the following natural deduction system:

$$(Ax): \frac{}{\Gamma, x{:}S \vdash x : A \mid \Delta}\ (S \leq_{\mathrm{S}} A) \qquad (\cap): \frac{\Gamma \vdash M : A_i \mid \Delta \quad (\forall i \in I)}{\Gamma \vdash M : \cap_I A_i \mid \Delta}\ (I = \emptyset \vee |I| \geq 2)$$

$$(Abs): \frac{\Gamma, x{:}S \vdash M : \neg\mathsf{C} \mid \Delta}{\Gamma \vdash \lambda x.M : \neg(S \times \mathsf{C}) \mid \Delta}\ (x \notin \Gamma) \qquad (\mu_1): \frac{\Gamma \vdash M : \neg\mathsf{D} \mid \alpha{:}\mathsf{C}, \beta{:}\mathsf{C}', \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \neg\mathsf{C} \mid \beta{:}\mathsf{C}', \Delta}\ (\beta \neq \alpha \notin \Delta, \mathsf{C}' \leq_{\mathrm{S}} \mathsf{D})$$

$$(App): \frac{\Gamma \vdash M : \neg(S \times \mathsf{C}) \mid \Delta \quad \Gamma \vdash N : S \mid \Delta}{\Gamma \vdash MN : \neg\mathsf{C} \mid \Delta} \qquad (\mu_2): \frac{\Gamma \vdash M : \neg\mathsf{D} \mid \alpha{:}\mathsf{C}, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \neg\mathsf{C} \mid \Delta}\ (\alpha \notin \Delta, \mathsf{C} \leq_{\mathrm{S}} \mathsf{D})$$

We write $\Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$ for judgements derivable using these rules, and prefix this with '$\mathcal{D} ::$ ' if we want to name the derivation.

Notice that a continuation type C is never a derivable type for a term, and that we cannot derive $\Gamma, x{:}\omega \vdash_{\mathrm{S}} x : \omega \mid \Delta$ using (Ax), but only through $(\cap)$.

It is possible to define this system using the following rules:

$$(App') : \frac{\Gamma \vdash M : \neg(S \times C) \mid \Delta \quad \Gamma \vdash N : S \mid \Delta}{\Gamma \vdash MN : \neg C \mid \Delta}\ (S \neq \omega) \qquad \frac{\Gamma \vdash M : \neg(\omega \times C) \mid \Delta}{\Gamma \vdash MN : \neg C \mid \Delta}$$

which would express that ω is a '*don't care*' type; we view terms that are assigned the type constant ω as *untyped*.

Example 3.5 Take $S = A \cap \neg(A \times C)$, and let $C'' \leq_{\mathrm{S}} C$ and $C'' \leq_{\mathrm{S}} S \times C'$ (we can always take $C'' = C \cap S \times C'$); notice that $S \leq_{\mathrm{S}} A$ and $S \leq_{\mathrm{S}} \neg(A \times C)$. Mirroring Example 2.3, we can derive $\emptyset \vdash_{\mathrm{S}} \mu\gamma.[\gamma]\lambda x.\mu\alpha.[\gamma]xx : \neg C'' \mid \emptyset$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}S \vdash x : \neg(A \times C) \mid \alpha{:}C', \gamma{:}C''}\,(Ax) \qquad \dfrac{}{x{:}S \vdash x : A \mid \alpha{:}C', \gamma{:}C''}\,(Ax)}{x{:}S \vdash xx : \neg C \mid \alpha{:}C', \gamma{:}C''}\,(App)}{x{:}S \vdash \mu\alpha.[\gamma]xx : \neg C' \mid \gamma{:}C''}\,(\mu_1)}{\emptyset \vdash \lambda x.\mu\alpha.[\gamma]xx : \neg(S \times C') \mid \gamma{:}C''}\,(Abs)}{\emptyset \vdash \mu\gamma.[\gamma]\lambda x.\mu\alpha.[\gamma]xx : \neg C'' \mid \emptyset}\,(\mu_2)$$

Notice that, for '$\vdash_{\mathrm{S}}$', we can reformulate (Ax) as:

$$(Ax) : \frac{}{\Gamma, x{:}\cap_I A_i \vdash x : A_j \mid \Delta}\ (j \in I)$$

(but cannot reformulate rules (μ_1) and (μ_2) in a similar way) and that rule

$$(Ax') : \frac{}{\Gamma, x{:}S \vdash x : T \mid \Delta}\ (S \leq_{\mathrm{S}} T)$$

is admissible. In fact, we can show:

Lemma 3.6 The type assignment rule (Ax') is derivable.

Proof: If $S \leq_{\mathrm{S}} T$, then by Lemma 3.3, there are $n, m \geq 0$, and A_i $(i \in \underline{n})$, B_j $(j \in \underline{m})$, such that $S = \cap_{\underline{n}} A_i$, $T = \cap_{\underline{m}} B_j$, and $\{B_1, \ldots, B_m\} \subseteq \{A_1, \ldots, A_n\}$; but then $\cap_{\underline{n}} A_i \leq_{\mathrm{S}} B_j$, for every $j \in \underline{m}$. So, in particular, we can construct:

$$\frac{\dfrac{}{\Gamma, x{:}\cap_{\underline{n}} A_i \vdash_{\mathrm{S}} x : B_1 \mid \Delta}\,(Ax) \quad \cdots \quad \dfrac{}{\Gamma, x{:}\cap_{\underline{n}} A_i \vdash_{\mathrm{S}} x : B_m \mid \Delta}\,(Ax)}{\Gamma, x{:}\cap_{\underline{n}} A_i \vdash_{\mathrm{S}} x : \cap_{\underline{m}} B_j \mid \Delta}\,(\cap)$$

The following properties are standard and of use in many of the proofs of this paper.

Lemma 3.7 (Thinning & Weakening) *i) Let $\Gamma \vdash_{\mathrm{S}} M : A \mid \Delta$; take $\Gamma_M = \{x{:}S \in \Gamma \mid x \in fv(M)\}$ and $\Delta_M = \{\alpha{:}C \in \Delta \mid \alpha \in fn(M)\}$, then $\Gamma_M \vdash_{\mathrm{S}} M : A \mid \Delta_M$.*

ii) a) Let $\Gamma \vdash_{\mathrm{S}} M : A \mid \Delta$, and $y \notin \Gamma$ and $y \notin bv(M)$, then $\Gamma, y{:}B \vdash_{\mathrm{S}} M : A \mid \Delta$.

b) Let $\Gamma \vdash_{\mathrm{S}} M : A \mid \Delta$, and $\alpha \notin \Delta$ and $\alpha \notin bn(M)$, then $\Gamma \vdash_{\mathrm{S}} M : A \mid \alpha{:}C, \Delta$.

Proof: By easy induction.

We can show:

Lemma 3.8 i) $\Gamma \vdash_{\mathrm{S}} M : \cap_I A_i \mid \Delta$, if and only if $\Gamma \vdash_{\mathrm{S}} M : A_i \mid \Delta$, for all $i \in I$.

ii) If $\Gamma_i \vdash_{\mathrm{S}} M : A_i \mid \Delta_i$ for all $i \in I$, then $\cap_I \Gamma_i \vdash_{\mathrm{S}} M : \cap_I A_i \mid \cap_I \Delta_i$.

iii) $\Gamma \vdash_{\mathrm{S}} M : S \cap T \mid \Delta$, if and only if $\Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$ and $\Gamma \vdash_{\mathrm{S}} M : T \mid \Delta$.

iv) $\Gamma, x{:}S \vdash_{\mathrm{S}} x : T \mid \Delta$, if and only if $S \leq_{\mathrm{S}} T$.

v) $\Gamma \vdash_{\mathrm{S}} M : S \mid \alpha{:}C, \beta{:}C, \Delta$ if and only if $\Gamma \vdash_{\mathrm{S}} M\{\alpha/\beta\} : S \mid \alpha{:}C, \Delta$.

Proof: Easy.

Although '$\leq_S$' is restricted to just three rules in '$\vdash_S$', we can show that a generic $(\leq_S)$ rule (as the $(\leq_\wedge)$ rule of '$\vdash_\wedge$') is admissible.

Lemma 3.9 *If* $\Gamma \vdash_S M : S \mid \Delta$, $\Gamma' \leq_S \Gamma$, $\Delta' \leq_S \Delta$, *and* $S \leq_S T$, *then* $\Gamma' \vdash_S M : T \mid \Delta'$.

Proof: By induction on the structure of derivations.

(Ax): Then $M \equiv x$, $S = A$, and there exists $x{:}R \in \Gamma$ such that $R \leq_S A$. Since $\Gamma' \leq_S \Gamma$, there exists $x{:}R' \in \Gamma'$ such that $R' \leq_S R$. Notice that then $R' \leq_S T$, and by (Ax'), $\Gamma' \vdash_S x : T \mid \Delta'$.

(Abs): Then $M = \lambda x.N$, $S = \neg(R \times C)$ and $\Gamma, x{:}R \vdash_S N : \neg C \mid \Delta$. Since $\neg(R \times C) \leq_S T$, we have $S = T$. Then by induction $\Gamma', x{:}R \vdash_S N : \neg C \mid \Delta'$, and we get $\Gamma' \vdash_S \lambda x.N : \neg(R \times C) \mid \Delta'$ by (Abs).

(App): Then $M \equiv PQ$, $S = \neg C$, and there exists R such that $\Gamma \vdash_S M : \neg(R \times C) \mid \Delta$ and $\Gamma \vdash_S N : R \mid \Delta$. Since $\neg C \leq_S T$, we have $T = \neg C$, so by induction $\Gamma' \vdash_S P : \neg(R \times C) \mid \Delta'$ and by (App) we get $\Gamma' \vdash_S PQ : \neg C \mid \Delta'$.

(μ_1): Then $M = \mu\alpha.[\beta]N$, $S = \neg C$, $\Delta = \beta{:}C_0, \Delta_0$ and $\Gamma \vdash_S N : \neg D \mid \alpha{:}C, \beta{:}C_0, \Delta_0$ with $C_0 \leq_S D$. Since $\neg C \leq_S T$, in fact $T = \neg C$. Since $\Delta' \leq_S \Delta$, there exist $C'_0 \leq_S C_0$ and $\Delta'_0 \leq_S \Delta_0$ such that $\Delta' = \beta{:}C'_0, \Delta'_0$ and $\alpha{:}C, \beta{:}C'_0, \Delta'_0 \leq_S \alpha{:}C, \beta{:}C_0, \Delta_0$. Then by induction we have $\Gamma' \vdash_S N : \neg D \mid \alpha{:}C, \beta{:}C'_0, \Delta'_0$. Since $C'_0 \leq_S C_0 \leq_S D$, we have $\Gamma' \vdash_S \mu\alpha.[\beta]N : \neg C \mid \Delta'$ by (μ_1).

(μ_2): Then $M = \mu\alpha.[\alpha]N$, $S = \neg C$, and $\Gamma \vdash_S N : \neg D \mid \alpha{:}C, \Delta$ with $C \leq_S D$. Since $\neg C \leq_S T$, in fact $T = \neg C$, so by induction $\Gamma' \vdash_S N : \neg D \mid \alpha{:}C, \Delta'$. Then also $\Gamma' \vdash_S \mu\alpha.[\alpha]N : \neg C \mid \Delta'$ by (μ_2).

$(\cap)$: Then $S = \cap_I A_i$, and $\Gamma \vdash_S M : A_i \mid \Delta$, for all $i \in I$. Also, by Lemma 3.3 $T = \cap_J A_j$ with $J \subseteq I$, and $\Gamma' \vdash_S M : A_j \mid \Delta'$, for all $j \in J$, so by $(\cap)$, $\Gamma' \vdash_S M : T \mid \Delta'$. □

4 Subject reduction and expansion

We will now show the first of our main results, by showing that our notion of type assignment is sound and complete, *i.e.* closed under conversion between terms. We start by showing two variants of the substitution lemma.

Lemma 4.1 (Structural substitution lemma) $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : T \mid \gamma{:}C, \Delta$ *if and only if there exists* S *such that* $\Gamma \vdash_S N : S \mid \Delta$, *and* $\Gamma \vdash_S M : T \mid \alpha{:}S \times C, \Delta$.

Proof: By nested induction; the outermost is on the structure of types, and the innermost on the structure of terms.

$(T = \omega)$: $(\Rightarrow)$: Take $S = \omega$; by $(\cap)$ we have both $\Gamma \vdash_S N : \omega \mid \Delta$ and $\Gamma \vdash_S M : \omega \mid \alpha{:}\omega \times C, \Delta$.

$(\Leftarrow)$: By $(\cap)$ we have $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : \omega \mid \gamma{:}C, \Delta$.

$(T = \cap_I A_i)$: $(\Rightarrow)$: If $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : \cap_I A_i \mid \gamma{:}C, \Delta$, then by Lemma 3.8 we have that $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : A_i \mid \gamma{:}C, \Delta$, for every $i \in I$; then by induction, there exists S_i such that $\Gamma \vdash_S M : A_i \mid \alpha{:}S_i \times C, \Delta$ and $\Gamma \vdash_S N : S_i \mid \Delta$. Take $S = \cap_I S_i$; then $S \leq_S S_i$ and by Lemma 3.9, $\Gamma \vdash_S M : A_i \mid \alpha{:}S \times C, \Delta$, for every $i \in I$. By $(\cap)$ we get both $\Gamma \vdash_S N : S \mid \Delta$ and $\Gamma \vdash_S M : T \mid \alpha{:}S \times C, \Delta$.

$(\Leftarrow)$: If $\Gamma \vdash_S M : \cap_I A_i \mid \alpha{:}S \times C, \Delta$, then by $(\cap)$, $\Gamma \vdash_S M : A_i \mid \alpha{:}S \times C, \Delta$, for every $i \in I$. If also $\Gamma \vdash_S N : S \mid \Delta$, then by induction $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : A_i \mid \gamma{:}C, \Delta$, and by $(\cap)$, $\Gamma \vdash_S M\{N{\cdot}\gamma/\alpha\} : T \mid \gamma{:}C, \Delta$.

$(T = \neg C')$: By induction on the structure of terms (we only show the interesting cases in detail).

$(M \equiv x)$: Then $x\{N{\cdot}\gamma/\alpha\} = x$.

$(\Rightarrow)$: Take $S = \omega$. If $\Gamma \vdash_S x : \neg C' \mid \gamma{:}C, \Delta$ then by Lemma 3.7 (both (*ii*) and (*i*)), also

$\Gamma \vdash_S x : \neg C' \mid \alpha{:}\omega \times C, \Delta$, and by $(\cap)$ we have $\Gamma \vdash_S N : \omega \mid \Delta$.

$(\Leftarrow)$: If $\Gamma \vdash_S x : \neg C' \mid \alpha{:}S \times C, \Delta$, then by Lemma 3.7 also $\Gamma \vdash_S x : \neg C' \mid \gamma{:}C, \Delta$

$(M \equiv \lambda y.P)$: Notice that $(\lambda y.P)\{N{\cdot}\gamma/\alpha\} = \lambda y.P\{N{\cdot}\gamma/\alpha\}$.

$$\begin{array}{lcl}
\exists S\ [\Gamma \vdash_S \lambda y.P : \neg C' \mid \alpha{:}S \times C, \Delta \times \Gamma \vdash_S N : S \mid \Delta] & \Longleftrightarrow & (Abs)\\
\exists S, R, D\ [\Gamma, y{:}R \vdash_S P : \neg D \mid \alpha{:}S \times C, \Delta \,\&\, C' = R \times D \,\&\, \Gamma \vdash_S N : S \mid \Delta] & \Longleftrightarrow & (IH)\\
\exists R, D\ [\Gamma, y{:}R \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg D \mid \gamma{:}C, \Delta \,\&\, C' = R \times D] & \Longleftrightarrow & (Abs)\\
\Gamma \vdash_S (\lambda y.P)\{N{\cdot}\gamma/\alpha\} : \neg C' \mid \Delta & &
\end{array}$$

$(M = PQ)$: Notice that $(PQ)\{N{\cdot}\gamma/\alpha\} = P\{N{\cdot}\gamma/\alpha\}\,Q\{N{\cdot}\gamma/\alpha\}$.

$(\Rightarrow)$: Then there exists R such that both $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg(R \times C') \mid \gamma{:}C, \Delta$ and $\Gamma \vdash_S Q\{N{\cdot}\gamma/\alpha\} : R \mid \gamma{:}C, \Delta$. Then by induction, there are S_1, S_2 such that $\Gamma \vdash_S P : \neg(R \times C') \mid \alpha{:}S_1 \times C, \Delta$ and $\Gamma \vdash_S N : S_1 \mid \Delta$, as well as $\Gamma \vdash_S Q : R \mid \alpha{:}S_2 \times C, \Delta$ and $\Gamma \vdash_S N : S_2 \mid \Delta$. Take $S = S_1 \cap S_2$; notice that then $S \times C \leq_S S_i \times C$, so $\alpha{:}S \times C, \Delta \leq_S \alpha{:}S_i \times C, \Delta$, for $i = 1,2$. Then by Lemma 3.9 both $\Gamma \vdash_S P : \neg(R \times C') \mid \alpha{:}S \times C, \Delta$ and $\Gamma \vdash_S Q : R \mid \alpha{:}S \times C, \Delta$, and by (App) we get $\Gamma \vdash_S PQ : A \mid \alpha{:}S \times C, \Delta$. Notice that $\Gamma \vdash_S N : S \mid \Delta$ follows by Lemma 3.8.

$(\Leftarrow)$: Then there exists R such that $\Gamma \vdash_S P : \neg(R \times C') \mid \alpha{:}S \times C, \Delta$ and $\Gamma \vdash_S Q : R \mid \alpha{:}S \times C, \Delta$. Then by induction $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg(R \times C') \mid \gamma{:}C, \Delta$ and $\Gamma \vdash_S Q\{N{\cdot}\gamma/\alpha\} : R \mid \gamma{:}C, \Delta$ and the result follows by (App).

$(M \equiv \mu\delta.[\beta]P,\ \delta \neq \beta,\ \alpha \neq \beta)$: Notice that then $(\mu\delta.[\beta]P)\{N{\cdot}\gamma/\alpha\} = \mu\delta.[\beta]P\{N{\cdot}\gamma/\alpha\}$.

$$\begin{array}{lcl}
\exists S\ [\Gamma \vdash_S \mu\delta.[\beta]P : \neg C' \mid \alpha{:}S \times C, \Delta \,\&\, \Gamma \vdash_S N : S \mid \Delta] & \Longleftrightarrow & (\mu_1)\\
\exists S, D, C''\ [\Gamma \vdash_S P : \neg D \mid \delta{:}C', \beta{:}C'', \alpha{:}S \times C, \Delta' \,\&\, \Gamma \vdash_S N : S \mid \Delta \,\& & & \\
\qquad\qquad \Delta = \beta{:}C'', \Delta' \,\&\, C'' \leq_S D] & \Longleftrightarrow & (IH)\\
\exists D, C''\ [\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg D \mid \delta{:}C', \beta{:}C'', \gamma{:}C, \Delta' \,\&\, \Delta = \beta{:}C'', \Delta' \,\&\, C'' \leq_S D] & \Longleftrightarrow & (\mu_1)\\
\Gamma \vdash_S (\mu\delta.[\beta]P)\{N{\cdot}\gamma/\alpha\} : \neg C' \mid \gamma{:}C, \Delta & &
\end{array}$$

Again, for $\Leftarrow$, if $\alpha \notin fn(P)$, then $S = \omega$.

$(M \equiv \mu\delta.[\alpha]P,\ \delta \neq \alpha)$: Notice that then $(\mu\delta.[\alpha]P)\{N{\cdot}\gamma/\alpha\} = \mu\delta.[\gamma]P\{N{\cdot}\gamma/\alpha\}N$.

$(\Rightarrow)$: If $\Gamma \vdash_S (\mu\delta.[\alpha]P)\{N{\cdot}\gamma/\alpha\} : \neg C' \mid \gamma{:}C, \Delta$, then $\Gamma \vdash_S \mu\delta.[\gamma]P\{N{\cdot}\gamma/\alpha\}N : \neg C' \mid \gamma{:}C, \Delta$. By (μ_1) there exists D such that $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\}N : \neg D \mid \delta{:}C', \gamma{:}C, \Delta$, and $C \leq_S D$. Then, by (App), there exists R such that $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg(R \times D) \mid \delta{:}C', \gamma{:}C, \Delta$ and $\Gamma \vdash_S N : R \mid \Delta$. By induction, there exists T such that $\Gamma \vdash_S P : \neg(R \times D) \mid \delta{:}C', \alpha{:}T \times C, \Delta$ and $\Gamma \vdash_S N : T \mid \Delta$. Take $S = R \cap T$, then $S \times C \leq_S T \times C$; so by Lemma 3.9 also $\Gamma \vdash_S P : \neg(R \times D) \mid \delta{:}C', \alpha{:}S \times C, \Delta$. Since also $S \times C \leq_S R \times C$, we get $\Gamma \vdash_S \mu\delta.[\alpha]P : \neg C' \mid \alpha{:}S \times C, \Delta$ by (μ_1) and $\Gamma \vdash_S N : S \mid \Delta$ by $(\cap)$.

$(\Leftarrow)$: If $\Gamma \vdash_S \mu\delta.[\alpha]P : \neg C' \mid \alpha{:}S \times C, \Delta$, then by (μ_1) there exists D such that $\Gamma \vdash_S P : \neg D \mid \delta{:}C', \alpha{:}S \times C, \Delta$ and $S \times C \leq_S D$. If $D \neq \Omega$, then $D = S' \times D'$ with $S \leq_S S'$ and $C \leq_S D'$; if $D = \Omega$, take $S' = S$, and $D' = \Omega$. Since $\Gamma \vdash_S N : S \mid \Delta$, by Lemma 3.9 also $\Gamma \vdash_S N : S' \mid \Delta$; since δ is bound, by Lemma 3.7:(*ii*) also $\Gamma \vdash_S N : S' \mid \delta{:}C', \Delta$. Then by induction $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\} : \neg(S' \times D') \mid \delta{:}C', \gamma{:}C, \Delta$. Since γ is fresh, by Lemma 3.7:(*ii*) also $\Gamma \vdash_S N : S' \mid \delta{:}C', \gamma{:}C, \Delta$, and by (App) we get $\Gamma \vdash_S P\{N{\cdot}\gamma/\alpha\}N : \neg D' \mid \delta{:}C', \gamma{:}C, \Delta$. From $C \leq_S D'$, by applying (μ_1) we get $\Gamma \vdash_S \mu\delta.[\gamma]P\{N{\cdot}\gamma/\alpha\}N : \neg C' \mid \gamma{:}C, \Delta$.

$(M \equiv \mu\delta.[\beta]P$ with $\delta \neq \beta$ and $\alpha \neq \beta$, $M \equiv \mu\delta.[\delta]P$, $M \equiv \lambda y.P)$: By induction. □

Naturally, the type inclusion relation plays a role in this proof, but nowhere in the proof do we need that to be contra-variant, so it only depends on '$\leq_S$'.

Likewise, we can show a similar result for term substitution:

Lemma 4.2 (Term substitution lemma) *$\Gamma \vdash_S M\{N/x\} : T \mid \Delta$ if and only if there exists S such that $\Gamma, x{:}S \vdash_S M : T \mid \Delta$ and $\Gamma \vdash_S N : S \mid \Delta$.*

Proof: As in the previous proof, the proof is by nested induction on the structure of types and the structure of terms. Here we just show the case $T = \neg C$.

$(M \equiv x): (\Rightarrow):$ Take $S = \neg C$; notice that $\Gamma, x{:}\neg C \vdash_S x : \neg C \mid \Delta$ by (Ax), and that $\Gamma \vdash_S N : \neg C \mid \Delta$ follows from $\Gamma \vdash_S x\{N/x\} : \neg C \mid \Delta$.

$(\Leftarrow):$ If $\Gamma, x{:}S \vdash_S x : \neg C \mid \Delta$, then $S \leq_S \neg C$. From $\Gamma \vdash_S N : S \mid \Delta$ and Lemma 3.9, we have $\Gamma \vdash_S N : \neg C \mid \Delta$, so $\Gamma \vdash_S x\{N/x\} : \neg C \mid \Delta$.

$(M \equiv y \neq x): (\Rightarrow):$ We have $y\{N/x\} \equiv y$ and $x \notin fv(y)$; take $S = \omega$, then by Lemma 3.7:(*i*), $\Gamma, x{:}\omega \vdash_S y : \neg C \mid \Delta$ and by $(\cap)$ we have $\Gamma \vdash_S N : \omega \mid \Delta$.

$(\Leftarrow):$ If $\Gamma, x{:}S \vdash_S y : \neg C \mid \Delta$, then by Lemma 3.7:(*i*) also $\Gamma \vdash_S y : \neg C \mid \Delta$.

$(M \equiv \lambda y.P):$

$$\begin{array}{ll}
\exists S\ [\Gamma, x{:}S \vdash_S \lambda y.P : \neg C \mid \Delta \;\&\; \Gamma \vdash_S N : S \mid \Delta] & \Longleftrightarrow\ (Abs) \\
\exists S, C', R\ [\Gamma, x{:}S, y{:}R \vdash_S P : \neg C' \mid \Delta \;\&\; C = R \times C' \;\&\; \Gamma \vdash_S N : S \mid \Delta] & \Longleftrightarrow\ (Weakening) \\
\exists S, C', R\ [\Gamma, x{:}S, y{:}R \vdash_S P : \neg C' \mid \Delta \;\&\; C = R \times C' \;\&\; \Gamma, y{:}R \vdash_S N : S \mid \Delta] & \Longleftrightarrow\ (IH) \\
\exists C', R\ [\Gamma, y{:}R \vdash_S P\{N/x\} : \neg C' \mid \Delta \;\&\; C = R \times C'] & \Longleftrightarrow\ (Abs) \\
\Gamma \vdash_S (\lambda y.P)\{N/x\} : \neg C \mid \Delta &
\end{array}$$

$(M = PQ): (\Rightarrow):$ Then there exists R such that both $\Gamma \vdash_S P\{N/x\} : \neg(R \times C) \mid \Delta$ and $\Gamma \vdash_S Q\{N/x\} : R \mid \Delta$. Then by induction, there are S_1, S_2 such that $\Gamma, x{:}S_1 \vdash_S P : \neg(R \times C) \mid \Delta$ and $\Gamma \vdash_S N : S_1 \mid \Delta$, as well as $\Gamma, x{:}S_2 \vdash_S Q : R \mid \Delta$ and $\Gamma \vdash_S N : S_2 \mid \Delta$. Take $S = S_1 \cap S_2$, then $S \leq_S S_i$ for $i = 1,2$. Then by Lemma 3.7:(*ii*) we get $\Gamma, x{:}S \vdash_S P : \neg(R \times C) \mid \Delta$ and $\Gamma, x{:}S \vdash_S Q : R \mid \Delta$, and by (App) we get $\Gamma, x{:}S \vdash_S PQ : \neg C \mid \Delta$; notice that $\Gamma \vdash_S N : S \mid \Delta$ by Lemma 3.3.

$(\Leftarrow):$ If $\Gamma, x{:}S \vdash_S PQ : \neg C \mid \Delta$, then there exists R such that $\Gamma, x{:}S \vdash_S P : \neg(R \times C) \mid \Delta$ and $\Gamma, x{:}S \vdash_S Q : R \mid \Delta$. Then, by induction, we have $\Gamma \vdash_S P\{N/x\} : \neg(R \times C) \mid \Delta$ and $\Gamma \vdash_S Q\{N/x\} : R \mid \Delta$; the result follows by (App).

$(M \equiv \mu\alpha.[\beta]P$ with $\alpha \neq \beta)$, $(M \equiv \mu\alpha.[\alpha]P):$ By induction. ☐

We can now show a soundness result for '$\vdash_S$', that states that assignable types are preserved under reduction.

Theorem 4.3 (Subject reduction) *If $M \rightarrow_{\beta\mu} N$ and $\Gamma \vdash_S M : S \mid \Delta$, then $\Gamma \vdash_S N : S \mid \Delta$.*

Proof: By induction on the definition of the one-step reduction relation; as above, we restrict the proof to the case that $S = A$, and just show the base cases of reduction.

$((\mu\beta.[\delta]P)Q \rightarrow \mu\gamma.[\delta]P\{Q{\cdot}\gamma/\beta\},\ \delta \neq \beta):$ If $\Gamma \vdash_S (\mu\beta.[\delta]P)Q : A \mid \Delta$, then by (App) there exist S, C, C', Δ such that $A = \neg C$, $\Delta = \delta{:}C', \Delta'$, and both $\Gamma \vdash_S \mu\beta.[\delta]P : \neg(S \times C) \mid \delta{:}C', \Delta'$ and $\Gamma \vdash_S Q : S \mid \delta{:}C', \Delta'$. Moreover, by (μ_1) there exist D such that $\Gamma \vdash_S P : \neg D \mid \beta{:}S \times C, \delta{:}C', \Delta'$ with $C' \leq_S D$. Then by Lemma 4.1 $\Gamma \vdash_S P\{Q{\cdot}\gamma/\beta\} : \neg D \mid \gamma{:}C, \delta{:}C', \Delta'$. By (μ_1) we get $\Gamma \vdash_S \mu\gamma.[\delta]P\{Q{\cdot}\gamma/\beta\} : \neg C \mid \delta{:}C', \Delta'$.

$((\mu\beta.[\beta]P)Q \rightarrow \mu\gamma.[\gamma]P\{Q{\cdot}\gamma/\beta\}Q):$ If $\Gamma \vdash_S (\mu\beta.[\beta]P)Q : A \mid \Delta$, then by (App) there exist S, C such that $A = \neg C$, $\Gamma \vdash_S \mu\beta.[\beta]P : \neg(S \times C) \mid \Delta$ and $\Gamma \vdash_S Q : S \mid \Delta$. Moreover, by (μ_2) there exist D such that $\Gamma \vdash_S P : \neg D \mid \beta{:}S \times C, \Delta'$ with $S \times C \leq_S D$. If $D \neq \Omega$, then there are S, C' such that $D = S' \times D'$ with $S \leq_S S'$ and $C \leq_S D'$; if $D = \Omega$, take $S' = S$, and $D' = \Omega$. Then we get $\Gamma \vdash_S P\{Q{\cdot}\gamma/\beta\} : \neg(S' \times C') \mid \gamma{:}C, \Delta'$ by Lemma 4.1. By Lemma 3.9 we have $\Gamma \vdash_S Q : S' \mid \Delta$, and by (App) we get $\Gamma \vdash_S P\{Q{\cdot}\gamma/\beta\}Q : \neg C' \mid \gamma{:}C, \Delta'$. Then by (μ_2) $\Gamma \vdash_S \mu\gamma.[\gamma]P\{Q{\cdot}\gamma/\beta\}Q : \neg C \mid \Delta'$.

$((\lambda x.P)Q \rightarrow P\{Q/x\}):$ If $\Gamma \vdash_S (\lambda x.P)Q : A \mid \Delta$, then there exists S, C such that $A = \neg C$ and both $\Gamma, x{:}S \vdash_S P : \neg C \mid \Delta$ and $\Gamma \vdash_S Q : S \mid \Delta$. The result follows by Lemma 4.2.

$(\mu\alpha.[\beta]\mu\gamma.[\delta]P \rightarrow \mu\alpha.([\delta]P)\{\beta/\gamma\}):$ If $\Gamma \vdash_S \mu\alpha.[\beta]\mu\gamma.[\delta]P : A \mid \Delta$, then there exist Δ_1, C_1, C_2, D_1 such that $A = \neg C_1$, $\Delta = \beta{:}C_2, \Delta_1$, $C_2 \leq_S D_1$, and $\Gamma \vdash_S \mu\gamma.[\delta]P : \neg D_1 \mid \beta{:}C_2, \alpha{:}C_1, \Delta_1$, where

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\Gamma \vdash_{\mathrm{S}} x : \neg(\mathtt{S}\times\omega\times\mathtt{C}) \mid \emptyset}\,(Ax)
\quad
\dfrac{}{\Gamma \vdash_{\mathrm{S}} z : \mathtt{S} \mid \emptyset}\,(Ax')
}{\Gamma \vdash_{\mathrm{S}} xz : \neg(\omega\times\mathtt{C}) \mid \emptyset}\,(App)
\quad
\dfrac{}{\Gamma \vdash_{\mathrm{S}} yz : \omega \mid \emptyset}\,(\cap)
}{\Gamma \vdash_{\mathrm{S}} xz(yz) : \neg\mathtt{C} \mid \emptyset}\,(App)
}{x{:}\neg(\mathtt{S}\times\omega\times\mathtt{C}), y{:}\mathtt{T} \vdash_{\mathrm{S}} \lambda z.xz(yz) : \neg(\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)
}{x{:}\neg(\mathtt{S}\times\omega\times\mathtt{C}) \vdash_{\mathrm{S}} \lambda yz.xz(yz) : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)
}{\emptyset \vdash_{\mathrm{S}} \lambda xyz.xz(yz) : \neg(\neg(\mathtt{S}\times\omega\times\mathtt{C})\times\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)
\quad
\dfrac{
\dfrac{
\dfrac{}{a{:}\mathtt{S}, b{:}\omega \vdash_{\mathrm{S}} a : \neg\mathtt{C} \mid \emptyset}\,(Ax)
}{a{:}\mathtt{S} \vdash_{\mathrm{S}} \lambda b.a : \neg(\omega\times\mathtt{C}) \mid \emptyset}\,(Abs)
}{\emptyset \vdash_{\mathrm{S}} \lambda ab.a : \neg(\mathtt{S}\times\omega\times\mathtt{C}) \mid \emptyset}\,(Abs)
}{\emptyset \vdash_{\mathrm{S}} (\lambda xyz.xz(yz))(\lambda ab.a) : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(App)
$$

Figure 3: A derivation for $\emptyset \vdash_{\mathrm{S}} (\lambda xyz.xz(yz))(\lambda ab.a) : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset$

perhaps $\alpha = \beta$ and then $\mathtt{C}_1 = \mathtt{C}_2$. Likewise, there exist $\Delta_2, \mathtt{C}_3, \mathtt{D}_2$ such that $\Delta_1 = \delta{:}\mathtt{C}_3, \Delta_2$, $\mathtt{C}_3 \leq_{\mathrm{S}} \mathtt{D}_2$, and $\Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D}_2 \mid \gamma{:}\mathtt{D}_1, \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_2, \alpha{:}\mathtt{C}_1, \Delta_2$. Since $\mathtt{C}_2 \leq_{\mathrm{S}} \mathtt{D}_1$, by Lemma 3.9, we have also $\Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D}_2 \mid \gamma{:}\mathtt{C}_2, \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_2, \alpha{:}\mathtt{C}_1, \Delta_2$. We now distinguish two cases:

$(\delta = \gamma)$: Then $\mathtt{C}_2 = \mathtt{C}_3$ and by Lemma 3.8 we have $\Gamma \vdash_{\mathrm{S}} P\{\beta/\gamma\} : \neg\mathtt{D}_2 \mid \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_3, \alpha{:}\mathtt{C}_1, \Delta_2$. Then $\mu\alpha.([\delta]P)\{\beta/\gamma\} = \mu\alpha.[\beta]P\{\beta/\gamma\}$, so $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]P\{\beta/\gamma\} : \neg\mathtt{C}_1 \mid \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_2, \Delta_2$ follows by (μ_1). We apply (μ_2) in case $\alpha = \beta$; then $\mathtt{C}_1 = \mathtt{C}_3$.

$(\delta \neq \gamma)$: Then by Lemma 3.8 we have $\Gamma \vdash_{\mathrm{S}} P\{\beta/\gamma\} : \neg\mathtt{D}_2 \mid \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_2, \alpha{:}\mathtt{C}_1, \Delta_2$. Since now $\mu\alpha.([\delta]P)\{\beta/\gamma\} = \mu\alpha.[\delta]P\{\beta/\gamma\}$, then $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\delta]P\{\beta/\gamma\} : \neg\mathtt{C}_1 \mid \delta{:}\mathtt{C}_3, \beta{:}\mathtt{C}_2, \Delta_2$ follows by (μ_1), and apply (μ_2) when $\alpha = \delta$. □

We can also show the reverse of the previous soundness result, that assignable types are preserved under expansion.

Theorem 4.4 (Subject expansion) *If $M \rightarrow_{\beta\mu} N$ and $\Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta$, then $\Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$.*

Proof: By induction on the definition of the one-step reduction relation; as above, we restrict the proof to the case that $\mathtt{S} = \mathtt{A}$, and just show the base cases.

$((\mu\beta.[\delta]P)Q \rightarrow \mu\gamma.[\delta]P\{Q{\cdot}\gamma/\beta\},\ \delta \neq \beta)$: By construction, $\gamma \neq \delta$; if $\Gamma \vdash_{\mathrm{S}} \mu\gamma.[\delta]P\{Q{\cdot}\gamma/\beta\} : \mathtt{A} \mid \Delta$, by (μ_1) there are Δ', $\mathtt{C}$, $\mathtt{C}'$, and $\mathtt{D}$ such that $\mathtt{A} = \neg\mathtt{C}$, $\Gamma \vdash_{\mathrm{S}} P\{Q{\cdot}\gamma/\beta\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C}, \delta{:}\mathtt{C}', \Delta'$, $\Delta = \delta{:}\mathtt{C}', \Delta'$, and $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$. By Lemma 4.1, there exists $\mathtt{S}$ such that $\Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \beta{:}\mathtt{S}\times\mathtt{C}, \delta{:}\mathtt{C}', \Delta'$ and $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \delta{:}\mathtt{C}', \Delta'$. By (μ_1), we get $\Gamma \vdash_{\mathrm{S}} \mu\beta.[\delta]P : \neg(\mathtt{S}\times\mathtt{C}) \mid \delta{:}\mathtt{C}', \Delta'$ and by (App) we get $\Gamma \vdash_{\mathrm{S}} (\mu\beta.[\delta]P)Q : \mathtt{A} \mid \Delta$.

$((\mu\beta.[\beta]P)Q \rightarrow \mu\gamma.[\gamma]P\{Q{\cdot}\gamma/\beta\}Q)$: If $\Gamma \vdash_{\mathrm{S}} \mu\gamma.[\gamma]P\{Q{\cdot}\gamma/\beta\}Q : \mathtt{A} \mid \Delta$, then by (μ_2) there are $\mathtt{C}$ and $\mathtt{D}$ such that $\mathtt{A} = \neg\mathtt{C}$, $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}$ and $\Gamma \vdash_{\mathrm{S}} P\{Q{\cdot}\gamma/\beta\}Q : \neg\mathtt{D} \mid \gamma{:}\mathtt{C}, \Delta$. Then by (App) there exists $\mathtt{S}_1$ such that $\Gamma \vdash_{\mathrm{S}} P\{Q{\cdot}\gamma/\beta\} : \neg(\mathtt{S}_1\times\mathtt{D}) \mid \gamma{:}\mathtt{C}, \Delta$. and $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{S}_1 \mid \gamma{:}\mathtt{C}, \Delta$. By Lemma 4.1, there exists $\mathtt{S}_2$ such that $\Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{S}_1\times\mathtt{D}) \mid \beta{:}\mathtt{S}_2\times\mathtt{C}, \Delta$ and $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{S}_2 \mid \Delta$. Take $\mathtt{S} = \mathtt{S}_1 \cap \mathtt{S}_2$, then $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \Delta$ by Lemma 3.8, and $\Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{S}_1\times\mathtt{D}) \mid \beta{:}\mathtt{S}\times\mathtt{C}, \Delta$ by Lemma 3.9. Then $\mathtt{S}\times\mathtt{C} \leq_{\mathrm{S}} \mathtt{S}_1\times\mathtt{D}$, so by (μ_2), $\Gamma \vdash_{\mathrm{S}} \mu\beta.[\beta]P : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta$; then, by (App), we get $\Gamma \vdash_{\mathrm{S}} (\mu\beta.[\delta]P)Q : \mathtt{A} \mid \Delta$.

$((\lambda x.M)N \rightarrow M\{N/x\})$: If $\Gamma \vdash_{\mathrm{S}} M\{N/x\} : \mathtt{A} \mid \Delta$, then by Lemma 4.2, there exists $\mathtt{S}, \mathtt{C}$ such that $\mathtt{A} = \neg\mathtt{C}$, and both $\Gamma, x{:}\mathtt{S} \vdash_{\mathrm{S}} M : \neg\mathtt{C} \mid \Delta$ and $\Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta$. Then, by both (Abs) and (App) we get $\Gamma \vdash_{\mathrm{S}} (\lambda x.M)N : \mathtt{A} \mid \Delta$.

$(M \rightarrow N$ through renaming$)$: Then N is of the shape $\mu\alpha.[\delta]P\{\beta/\gamma\}$. We distinguish the following cases (where we assume that distinct identifiers are not equal):

$(N = \mu\alpha.[\alpha]P\{\alpha/\gamma\})$: If $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\alpha]P\{\alpha/\gamma\} : \mathtt{A} \mid \Delta$, then by (μ_2), there are Δ', $\mathtt{C}$, and $\mathtt{D}$ such that $\Gamma \vdash_{\mathrm{S}} M\{\alpha/\gamma\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \Delta$, $\mathtt{A} = \neg\mathtt{C}$, and $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}$. Then also $\Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \gamma{:}\mathtt{C}, \Delta$.

Then either:

$(M = \mu\alpha.[\alpha]\mu\gamma.[\gamma]P)$: By (μ_2), $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\gamma]P : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\Delta$, and $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\alpha]\mu\gamma.[\gamma]P : \neg\mathtt{C} \mid \Delta$ by (μ_2).

$(M = \mu\alpha.[\alpha]\mu\gamma.[\alpha]P)$: By (μ_1), $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\alpha]P : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\Delta$, and $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\alpha]\mu\gamma.[\alpha]P : \neg\mathtt{C} \mid \Delta$ follows by (μ_2).

$(N = \mu\alpha.[\alpha]P\{\beta/\gamma\})$: Then $M = \mu\alpha.[\beta]\mu\gamma.[\alpha]P$. If $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\alpha]P\{\beta/\gamma\} : \mathtt{A} \mid \Delta$, then by (μ), there are Δ', $\mathtt{C}$, $\mathtt{C}'$, and $\mathtt{D}$ such that $\mathtt{A} = \neg\mathtt{C}$, $\Delta = \beta{:}\mathtt{C}',\Delta'$, $\Gamma \vdash_{\mathrm{s}} P\{\beta/\gamma\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$, and $\mathtt{C} \leq_{\mathrm{s}} \mathtt{D}$; then also $\Gamma \vdash_{\mathrm{s}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\gamma{:}\mathtt{C}',\beta{:}\mathtt{C}',\Delta'$. Then by (μ_1) we have $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\alpha]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$, and $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\beta]\mu\gamma.[\alpha]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta'$ again by (μ_1).

$(N = \mu\alpha.[\beta]P\{\beta/\gamma\})$: If $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\beta]P\{\beta/\gamma\} : \mathtt{A} \mid \Delta$, then by (μ_1) there are Δ', $\mathtt{C}$, $\mathtt{C}'$, and $\mathtt{D}$ such that $\mathtt{A} = \neg\mathtt{C}$, $\Delta = \beta{:}\mathtt{C}',\Delta'$, $\Gamma \vdash_{\mathrm{s}} P\{\beta/\gamma\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$, and $\mathtt{C}' \leq_{\mathrm{s}} \mathtt{D}$. Then $\Gamma \vdash_{\mathrm{s}} P : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\gamma{:}\mathtt{C}',\beta{:}\mathtt{C}',\Delta'$, and either:

$(M = \mu\alpha.[\beta]\mu\gamma.[\gamma]P)$: By (μ) we get $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\gamma]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$, and by (μ_1) we get $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\beta]\mu\gamma.[\gamma]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta'$.

$(M = \mu\alpha.[\beta]\mu\gamma.[\beta]P)$: By (μ_1) we get $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\beta]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$ and again by (μ_1) we get $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\beta]\mu\gamma.[\beta]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta'$.

$(N = \mu\alpha.[\delta]P\{\alpha/\gamma\})$: Then $M = \mu\alpha.[\alpha]\mu\gamma.[\delta]P$. If $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\delta]P\{\alpha/\gamma\} : \mathtt{A} \mid \Delta$, then by (μ_1) there are Δ', $\mathtt{C}$, $\mathtt{C}'$, and $\mathtt{D}$ such that $\Delta = \delta{:}\mathtt{C}',\Delta'$, $\mathtt{A} = \neg\mathtt{C}$, $\mathtt{C}' \leq_{\mathrm{s}} \mathtt{D}$, and $\Gamma \vdash_{\mathrm{s}} P\{\alpha/\gamma\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\delta{:}\mathtt{C}',\Delta'$. Then also $\Gamma \vdash_{\mathrm{s}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\ \gamma{:}\mathtt{C},\delta{:}\mathtt{C}',\Delta'$. By (μ_1), we get $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\delta]P : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\delta{:}\mathtt{C}',\Delta'$, and $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\alpha]\mu\gamma.[\delta]P : \neg\mathtt{C} \mid \delta{:}\mathtt{C}',\Delta'$ by (μ_2).

$(N = \mu\alpha.[\delta]P\{\beta/\gamma\})$: Then $M = \mu\alpha.[\beta]\mu\gamma.[\delta]P$.
If $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\delta]P\{\beta/\gamma\} : \mathtt{A} \mid \Delta$, then by (μ_1), there are Δ', $\mathtt{C}$, $\mathtt{C}'$, $\mathtt{C}''$, and $\mathtt{D}$ such that $\mathtt{A} = \neg\mathtt{C}$, $\Delta = \beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta'$, $\mathtt{C}'' \leq_{\mathrm{s}} \mathtt{D}$, and $\Gamma \vdash_{\mathrm{s}} P\{\beta/\gamma\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta'$; then also $\Gamma \vdash_{\mathrm{s}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\gamma{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta'$. By (μ_1) we get $\Gamma \vdash_{\mathrm{s}} \mu\gamma.[\delta]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta'$ and we get $\Gamma \vdash_{\mathrm{s}} \mu\alpha.[\beta]\mu\gamma.[\delta]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta'$ by (μ_1). $\square$

The following result is now immediate:

Theorem 4.5 (Semantics) *If $\Gamma \vdash_{\mathrm{s}} M : \mathtt{A} \mid \Delta$ and $M =_{\beta\mu} N$, then $\Gamma \vdash_{\mathrm{s}} N : \mathtt{A} \mid \Delta$.*

Proof: By induction on the definition of '$=_{\beta\mu}$', using the previous two results. $\square$

Example 4.6 As a consequence of the last result, observing that $(\lambda xyz.xz(yz))(\lambda ab.a) \to^*_{\beta\mu} \lambda yz.z$, we can assign to $(\lambda xyz.xz(yz))(\lambda ab.a)$ any type that is assignable to $\lambda yz.z$. Let $\mathtt{S} \leq_{\mathrm{s}} \neg\mathtt{C}$, then we can derive

$$\dfrac{\dfrac{\dfrac{}{y{:}\mathtt{T},z{:}\mathtt{S} \vdash_{\mathrm{s}} z : \neg\mathtt{C} \mid \emptyset}\,(Ax')}{y{:}\mathtt{T} \vdash_{\mathrm{s}} \lambda z.z : \neg(\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)}{\emptyset \vdash_{\mathrm{s}} \lambda yz.z : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)$$

Let $\Gamma = x{:}\neg(\mathtt{S}\times\omega\times\mathtt{C}),y{:}\mathtt{T},z{:}\mathtt{S}$, then we can derive:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\mathrm{s}} x : \neg(\mathtt{S}\times\omega\times\mathtt{C}) \mid \emptyset}\,(Ax) \quad \dfrac{}{\Gamma \vdash_{\mathrm{s}} z : \mathtt{S} \mid \emptyset}\,(Ax')}{\Gamma \vdash_{\mathrm{s}} xz : \neg(\omega\times\mathtt{C}) \mid \emptyset}\,(App) \quad \dfrac{}{\Gamma \vdash_{\mathrm{s}} yz : \omega \mid \emptyset}\,(\cap)}{\Gamma \vdash_{\mathrm{s}} xz(yz) : \neg\mathtt{C} \mid \emptyset}\,(App)}{x{:}\neg(\mathtt{S}\times\omega\times\mathtt{C}),y{:}\mathtt{T} \vdash_{\mathrm{s}} \lambda z.xz(yz) : \neg(\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)}{x{:}\neg(\mathtt{S}\times\omega\times\mathtt{C}) \vdash_{\mathrm{s}} \lambda yz.xz(yz) : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs)}{\emptyset \vdash_{\mathrm{s}} \lambda xyz.xz(yz) : \neg(\neg(\mathtt{S}\times\omega\times\mathtt{C})\times\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(Abs) \quad \dfrac{\dfrac{\dfrac{}{a{:}\mathtt{S},b{:}\omega \vdash_{\mathrm{s}} a : \neg\mathtt{C} \mid \emptyset}\,(Ax)}{a{:}\mathtt{S} \vdash_{\mathrm{s}} \lambda b.a : \neg(\omega\times\mathtt{C}) \mid \emptyset}\,(Abs)}{\emptyset \vdash_{\mathrm{s}} \lambda ab.a : \neg(\mathtt{S}\times\omega\times\mathtt{C}) \mid \emptyset}\,(Abs)}{\emptyset \vdash_{\mathrm{s}} (\lambda xyz.xz(yz))(\lambda ab.a) : \neg(\mathtt{T}\times\mathtt{S}\times\mathtt{C}) \mid \emptyset}\,(App)$$

Based on the result of Theorem 4.5, we could follow the path of [2] and the approach of [12] and continue to define a strict filter semantics, but forego that here for the moment.

As to the two η-reduction rules (η) and $(\mu\eta)$, we cannot show both similar soundness and completeness results.

Remark 4.7 As is also the case for the strict type assignment system for the pure λ-calculus [2], we cannot show that assignable types are preserved by (η) in '$\vdash_{\mathrm{S}}$'. For example, in '$\vdash_{\mathrm{S}}$' we can derive (where $\mathtt{C} \neq \mathtt{D}$):

$$\dfrac{\dfrac{\dfrac{}{x{:}\neg\mathtt{C}\cap\neg\mathtt{D},y{:}\neg(\neg\mathtt{C}\times\mathtt{C}) \vdash_{\lambda\mu} y : \neg(\neg\mathtt{C}\times\mathtt{C}) \mid \emptyset}\,(Ax) \quad \dfrac{}{x{:}\neg\mathtt{C}\cap\neg\mathtt{D},y{:}\neg(\neg\mathtt{C}\times\mathtt{C}) \vdash_{\lambda\mu} x : \neg\mathtt{C} \mid \emptyset}\,(Ax)}{x{:}\neg\mathtt{C}\cap\neg\mathtt{D},y{:}\neg(\neg\mathtt{C}\times\mathtt{C}) \vdash_{\lambda\mu} yx : \neg\mathtt{C} \mid \emptyset}\,(App)}{y{:}\neg(\neg\mathtt{C}\times\mathtt{C}) \vdash_{\lambda\mu} \lambda x.yx : \neg(\neg\mathtt{C}\cap\neg\mathtt{D}\times\mathtt{C}) \mid \emptyset}\,(Abs)$$

but can derive $y{:}\neg(\neg\mathtt{C}\times\mathtt{C}) \vdash_{\lambda\mu} y : \neg(\neg\mathtt{C}\cap\neg\mathtt{D}\times\mathtt{C}) \mid \emptyset$ only through contra-variance.

As to preservation under expansion for $\mu\eta$, we can show:

$(\mu\alpha.[\alpha]P \to P \textit{ with } \alpha \notin fn(M))$: Assume $\Gamma \vdash_{\mathrm{S}} P : \mathtt{A} \mid \Delta$, then there exists $\mathtt{C}$ such that $A = \neg\mathtt{C}$; since $\alpha \notin fn(M)$, by Lemma 3.7:(*i*) we can assume α does not occur in Δ; then also $\Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\Delta$ by Lemma 3.7:(*ii*). Then by (μ_2), we get $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\alpha]P : \neg\mathtt{C} \mid \Delta$.

but for η this is not possible. For example, $x{:}\neg\Omega \vdash_{\mathrm{S}} x : \neg\Omega \mid \emptyset$, but $x{:}\neg\Omega \not\vdash_{\mathrm{S}} \lambda y.xy : \neg\Omega \mid \emptyset$.

5 Derivation reduction

In this section, we will define a notion of reduction on derivations in '$\vdash_{\mathrm{S}}$', and show this to be strongly normalisable. As in [4, 7], this property will lead naturally to the characterisation of strong normalisation, approximation, head-normalisation, and normalisation. We will show this property using the proof technique of computability predicates as defined by Tait [39] ; it can be used to show the characterisation properties separately, but is only needed here for the one main result.

Strong normalisation of cut-elimination is a well-established property in the area of logic and has been studied profoundly in the past. In the area of type assignment for the λ-calculus (and the $\lambda\mu$-calculus), the corresponding property is that of strong normalisation of derivation reduction (also called cut-elimination in, for example, [18]), which mimics the normal reduction on terms to which the types are assigned.

The added complexity of intersection types implies that, unlike for ordinary systems of type assignment, there is a significant difference between derivation reduction and ordinary reduction; because of the presence of the type-constant ω, unlike 'normal' typed or type assignment system, not every term-redex occurs with types in a derivation.

Fore reasons of brevity, to save space, and for easy of definition, we will use the following notation for derivations, that aims to show the structure, in linear notation, in terms of rules applied.

Definition 5.1 *i*) If derivation $\mathcal{D}$ consists of an application of (Ax), then there are $n \geq 1$, $\mathtt{A}_1$, ..., $\mathtt{A}_n$, such that $\mathcal{D} :: \Gamma,x{:}\cap_{\underline{n}}\mathtt{A}_i \vdash_{\mathrm{S}} x : \mathtt{A}_j \mid \Delta$ with $j \in \underline{n}$; we then write

$$\mathcal{D} = \langle Ax \rangle :: \Gamma,x{:}\cap_{\underline{n}}\mathtt{A}_i \vdash_{\mathrm{S}} x : \mathtt{A}_j \mid \Delta.$$

ii) If derivation $\mathcal{D}$ ends with (Abs), there are x, M, $\mathtt{S}$, and $\mathtt{C}$ such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} \lambda x.M : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta$, and there is a sub-derivation $\mathcal{D}' :: \Gamma,x{:}\mathtt{S} \vdash_{\mathrm{S}} M : \neg\mathtt{C} \mid \Delta$ in $\mathcal{D}$; we then write

$$\mathcal{D} = \langle \mathcal{D}';Abs \rangle :: \Gamma \vdash_{\mathrm{S}} \lambda x.M : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta.$$

iii) If derivation $\mathcal{D}$ ends with (App), there are P, Q, such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} PQ : \neg\mathtt{C} \mid \Delta$, and there are $\mathtt{S}$ and sub-derivations $\mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{S} \times \mathtt{C}) \mid \Delta$ and $\mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \Delta$ in $\mathcal{D}$; we then write

$$\mathcal{D} = \langle \mathcal{D}_1 ; \mathcal{D}_2 ; App \rangle :: \Gamma \vdash_{\mathrm{S}} PQ : \neg\mathtt{C} \mid \Delta.$$

iv) If derivation $\mathcal{D}$ ends with (μ_1), there are $\mathtt{C}$ and $\mathtt{C}'$ such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]M : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta$, and there exists $\mathtt{D} \geq_{\mathrm{S}} \mathtt{C}'$ and a sub-derivation $\mathcal{D}' :: \Gamma \vdash_{\mathrm{S}} M : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta$; we then write

$$\mathcal{D} = \langle \mathcal{D}' ; \mu_1 \rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]M : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta$$

v) If derivation $\mathcal{D}$ ends with (μ_2), there are $\mathtt{C}$ and $\mathtt{S}$ such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\alpha]M : \neg\mathtt{C} \mid \Delta$, and there exists $\mathtt{D} \geq_{\mathrm{S}} \mathtt{C}$ and a sub-derivation $\mathcal{D}' :: \Gamma \vdash_{\mathrm{S}} M : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta$; we then write

$$\mathcal{D} = \langle \mathcal{D}' ; \mu_2 \rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\alpha]M : \neg\mathtt{C} \mid \Delta$$

vi) If derivation $\mathcal{D}$ ends with $(\cap)$, there are $n \geq 0$, $\mathtt{A}_1, \ldots, \mathtt{A}_n$ such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta$, and, for all $i \in \underline{n}$, there exists a $\mathcal{D}_i :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta$ that is a sub-derivation of $\mathcal{D}$; we then write

$$\mathcal{D} = \langle \mathcal{D}_1 ; \ldots ; \mathcal{D}_n ; \cap \rangle :: \Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta.$$

We will often abbreviate this short-hand notation further, and simply write $\langle \mathcal{D}_1 ; \mathcal{D}_2 ; App \rangle$ instead of

$$\langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{S} \times \mathtt{C}) \mid \Delta ; \mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \Delta ; App \rangle :: \Gamma \vdash_{\mathrm{S}} PQ : \neg\mathtt{C} \mid \Delta,$$

for example, when the actual term and types are known or of no concern.

We write $\mathcal{D} \leq_{\mathrm{S}} \mathcal{D}'$ when $\mathcal{D} = \langle \mathcal{D}_1 ; \ldots ; \mathcal{D}_n ; \cap \rangle :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$, $\mathcal{D}' = \langle \mathcal{D}'_1 ; \ldots ; \mathcal{D}'_m ; \cap \rangle :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{T} \mid \Delta$, and $\{\, \mathcal{D}'_j \mid j \in \underline{m} \,\} \subseteq \{\, \mathcal{D}_i \mid i \in \underline{n} \,\}$; notice that then $\mathtt{S} \leq_{\mathrm{S}} \mathtt{T}$.

We identify derivations that have the same structure in that they have the same rules applied in the same order (so derivations involving the same term, apart from sub-terms that are typed by ω) and say that these have the *same structure*; the types derived need not be the same. The notion of reduction on derivations $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$ defined in this section will follow ordinary reduction (on terms), by contracting typed redexes that occur in $\mathcal{D}$, *i.e.* redexes for sub-terms of M of the shape $(\lambda x.P)Q$, $(\mu\alpha.[\beta]P)Q$, or $\mu\alpha.[\beta]\mu\gamma.[\delta]P$ that are typed with types different from ω.

For the first, the following is a sub-derivation of $\mathcal{D}$:

$$\begin{array}{l} \langle \langle \mathcal{D}_1 :: \Gamma, x{:}\cap_{\underline{n}}\mathtt{A}_i \vdash_{\mathrm{S}} P : \neg\mathtt{C} \mid \Delta ; Abs \rangle :: \Gamma \vdash_{\mathrm{S}} \lambda x.P : \neg(\cap_{\underline{n}}\mathtt{A}_i \times \mathtt{C}) \mid \Delta ; \\ \qquad\qquad \mathcal{D}_2 :: \langle \mathcal{D}_2^1 ; \ldots ; \mathcal{D}_2^n ; \cap \rangle :: \Gamma \vdash_{\mathrm{S}} Q : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta ; App \rangle :: \Gamma \vdash_{\mathrm{S}} (\lambda x.P)Q : \neg\mathtt{C} \mid \Delta, \end{array}$$

For the second, we have

$$\begin{array}{l} \langle \langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{S} \times \mathtt{C},\beta{:}\mathtt{C}',\Delta ; \mu_1 \rangle :: \\ \qquad\qquad \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]P : \neg(\mathtt{S} \times \mathtt{C}) \mid \beta{:}\mathtt{C}',\Delta ; \\ \qquad\qquad\qquad \mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \beta{:}\mathtt{C}',\Delta ; App \rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\alpha.[\beta]P)Q : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta \end{array}$$

with $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$, and for the third

$$\begin{array}{l} \langle \langle \mathcal{D} :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\gamma{:}\mathtt{C}'',\delta{:}\mathtt{D}',\Delta ; r_1 \rangle :: \Gamma \vdash_{\mathrm{S}} \mu\gamma.[\delta]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\delta{:}\mathtt{D}',\Delta ; r_2 \rangle :: \\ \qquad\qquad\qquad\qquad \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]\mu\gamma.[\delta]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\delta{:}\mathtt{D}',\Delta \end{array}$$

where $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$ and $\mathtt{C} \leq_{\mathrm{S}} \mathtt{C}''$, and $r_i \in \{\, \mu_1,\mu_2 \,\}$ for $i \in \underline{2}$, depending on if $\alpha = \beta$, etc. A derivation of either of these structures will be called a *derivation redex*. We will define reduction on derivations by replacing the derivation for a term redex by a derivation for its contractum; this has, because the system at hand uses intersection types, including ω, to be defined with care, since in $\mathcal{D} :: \Gamma \vdash_{\mathrm{s}} M : A \mid \Delta$ it is possible that M contains a redex whereas $\mathcal{D}$ does not.

Consider a derivation for the redex $(\lambda x.P)Q$, which we can assume to be shaped like:

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{}{\Gamma,x{:}\cap_{\underline{n}}A_i \vdash_{\mathrm{s}} x : A_{q_1} \mid \Delta}\,(Ax) \quad \cdots \quad \dfrac{}{\Gamma,x{:}\cap_{\underline{n}}A_i \vdash_{\mathrm{s}} x : A_{q_m} \mid \Delta}\,(Ax) \\ \boxed{\quad\mathcal{D}_1\quad} \\ \Gamma,x{:}\cap_{\underline{n}}A_i \vdash_{\mathrm{s}} P : \neg C \mid \Delta\end{array}}{\Gamma \vdash_{\mathrm{s}} \lambda x.P : \neg(\cap_{\underline{n}}A_i \times C) \mid \Delta}\,(Abs) \qquad \dfrac{\begin{array}{c}\boxed{\mathcal{D}_2^1} \\ \Gamma \vdash_{\mathrm{s}} Q : A_1 \mid \Delta\end{array} \quad \cdots \quad \begin{array}{c}\boxed{\mathcal{D}_2^n} \\ \Gamma \vdash_{\mathrm{s}} Q : A_n \mid \Delta\end{array}}{\Gamma \vdash_{\mathrm{s}} Q : \cap_{\underline{n}}A_i \mid \Delta}\,(\cap)}{\Gamma \vdash_{\mathrm{s}} (\lambda x.P)Q : \neg C \mid \Delta}\,(App)$$

with $n \geq 2$; note that $A_{q_j} \in \{\, A_1,\ldots,A_n \,\}$, for all $j \in \underline{m}$, and that this derivation directly corresponds to the linear notation we presented above, so $\mathcal{D}_2 :: \langle \mathcal{D}_2^1;\ldots;\mathcal{D}_2^n;\cap\rangle$.

Contracting this derivation redex will construct a derivation for the term $P\{Q/x\}$, and will be written as

$$\mathcal{D}_1\{\,\mathcal{D}_2/x{:}\cap_{\underline{n}}A_i\,\} :: \Gamma \vdash_{\mathrm{s}} P\{Q/x\} : \neg C \mid \Delta.$$

However, when creating a derivation for $P\{Q/x\}$, it is *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: notice that *no* sub-derivation for $\Gamma \vdash_{\mathrm{s}} x : \cap_{\underline{n}}A_i \mid \Delta$ need exist in $\mathcal{D}_1$, and that the system lacks an $(\cap E)$ rule. Instead, since each A_{q_j} occurs in $\cap_{\underline{n}}A_i$, the approach used in this paper for derivation substitution will be to replace all derivations $\mathcal{D}_j = \langle Ax\rangle :: \Gamma,x{:}\cap_{\underline{n}}A_i \vdash_{\mathrm{s}} x : A_{q_j} \mid \Delta$ by the derivation $\mathcal{D}_2^{q_j} :: \Gamma \vdash_{\mathrm{s}} Q : A_{q_j} \mid \Delta$, and replace x by Q throughout the derivation $\mathcal{D}_1$ (notice that, by Barendregt's convention, this substitution is capture avoiding) to obtain:

$$\begin{array}{c}\begin{array}{c}\boxed{\mathcal{D}_2^{q_1}} \\ \Gamma \vdash_{\mathrm{s}} Q : A_{q_1} \mid \Delta\end{array} \quad \cdots \quad \begin{array}{c}\boxed{\mathcal{D}_2^{q_m}} \\ \Gamma \vdash_{\mathrm{s}} Q : A_{q_m} \mid \Delta\end{array} \\ \boxed{\qquad\quad\mathcal{D}_1\qquad\quad} \\ \Gamma \vdash_{\mathrm{s}} P\{Q/x\} : \neg C \mid \Delta\end{array}$$

Based on that intuition, first we formally define the notion of derivation substitution that deals with the derivation equivalent of (normal) term substitution.

Definition 5.2 (Derivation substitution) Give the derivations $\mathcal{D} :: \Gamma,x{:}S \vdash_{\mathrm{s}} M : T \mid \Delta$, and $\mathcal{D}_0 :: \Gamma \vdash_{\mathrm{s}} N : S \mid \Delta$, the derivation $\mathcal{D}\{\,\mathcal{D}_0/x{:}S\,\} :: \Gamma \vdash_{\mathrm{s}} M\{N/x\} : T \mid \Delta$, the result of *substituting $\mathcal{D}_0$ for $x{:}S$ in $\mathcal{D}$*, is defined by induction on the structure of derivations by:

$(\langle Ax\rangle :: \Gamma,x{:}S \vdash_{\mathrm{s}} x : T \mid \Delta)$: Then $S = \cap_{\underline{n}}A_i$ and $T = A_j$ with $j \in \underline{n}$. Then $\mathcal{D}_0$ is shaped like:

$$\langle \mathcal{D}_0^1 :: \Gamma \vdash_{\mathrm{s}} N : A_1 \mid \Delta;\ldots;\mathcal{D}_0^n :: \Gamma \vdash_{\mathrm{s}} N : A_n \mid \Delta;\cap\rangle :: \Gamma \vdash_{\mathrm{s}} N : \cap_{\underline{n}}A_i \mid \Delta,$$

so, in particular, $\mathcal{D}_0^j :: \Gamma \vdash_{\mathrm{s}} N : A_j \mid \Delta$. Then $\mathcal{D}\{\,\mathcal{D}_0/x{:}S\,\} \triangleq \mathcal{D}_0^j$.

$(\langle Ax\rangle :: \Gamma,x{:}S \vdash_{\mathrm{s}} y : T \mid \Delta$ *with* $x \not\equiv y)$: Then $\mathcal{D}\{\,\mathcal{D}_0/x{:}S\,\} \triangleq \langle Ax\rangle :: \Gamma \vdash_{\mathrm{s}} y : T \mid \Delta$.

$(\langle \mathcal{D}_1 :: \Gamma,x{:}S,y{:}R \vdash_{\mathrm{s}} P : \neg C \mid \Delta;Abs\rangle :: \Gamma,x{:}S \vdash_{\mathrm{s}} \lambda y.P : \neg(R \times C) \mid \Delta)$: Notice that $x \neq y$ and $y \notin fv(N)$. Let

$$\mathcal{D}_1' \;=\; \mathcal{D}_1\{\,\mathcal{D}_0/x{:}S\,\} :: \Gamma,y{:}R \vdash_{\mathrm{s}} P\{N/x\} : \neg C \mid \Delta.$$

Then $\mathcal{D}\{\,\mathcal{D}_0/x{:}A\,\} \;\triangleq\; \langle \mathcal{D}_1';Abs\rangle :: \Gamma \vdash_{\mathrm{s}} (\lambda y.P)\{N/x\} : \neg(R \times C) \mid \Delta$.

$(\langle \mathcal{D}_1 :: \Gamma,x{:}S \vdash_{\mathrm{s}} P : \neg(R \times C) \mid \Delta;\mathcal{D}_2 :: \Gamma,x{:}S \vdash_{\mathrm{s}} Q : R \mid \Delta;App\rangle :: \Gamma,x{:}S \vdash_{\mathrm{s}} PQ : \neg C \mid \Delta)$: Let

$$\begin{array}{rcl}\mathcal{D}_1' &=& \mathcal{D}_1\{\,\mathcal{D}_0/x{:}S\,\} :: \Gamma \vdash_{\mathrm{s}} P\{N/x\} : \neg(R \times C) \mid \Delta, \text{ and} \\ \mathcal{D}_2' &=& \mathcal{D}_2\{\,\mathcal{D}_0/x{:}S\,\} :: \Gamma \vdash_{\mathrm{s}} Q\{N/x\} : R \mid \Delta,\end{array}$$

then $\mathcal{D}\{\,\mathcal{D}_0/x{:}S\,\} \;\triangleq\; \langle \mathcal{D}_1',\mathcal{D}_2';App\rangle :: \Gamma \vdash_{\mathrm{s}} (PQ)\{N/x\} : \neg C \mid \Delta$.

$(\langle \mathcal{D}_1 :: \Gamma,x{:}S \vdash_{\mathrm{s}} P : \neg D \mid \alpha{:}C,\beta{:}C',\Delta;\mu_1\rangle :: \Gamma,x{:}S \vdash_{\mathrm{s}} \mu\alpha.[\beta]P : \neg C \mid \beta{:}C',\Delta)$: Let

$$\mathcal{D}'_1 \;=\; \mathcal{D}_1\{\mathcal{D}_0/x{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N/x\} : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta.$$

Then $\mathcal{D}\{\mathcal{D}_0/x{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1 ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\alpha.[\beta]P)\{N/x\} : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta.$

$(\langle \mathcal{D}_1 :: \Gamma,x{:}\mathtt{S} \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta ; \mu_2\rangle :: \Gamma,x{:}\mathtt{S} \vdash_{\mathrm{S}} \mu\alpha.[\alpha]P : \neg\mathtt{C} \mid \Delta)$: Let

$$\mathcal{D}'_1 \;=\; \mathcal{D}_1\{\mathcal{D}_0/x{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N/x\} : \neg\mathtt{C} \mid \alpha{:}\mathtt{C},\Delta.$$

Then $\mathcal{D}\{\mathcal{D}_0/x{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1 ; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\alpha.[\alpha]P)\{N/x\} : \neg\mathtt{C} \mid \Delta.$

$(\langle \mathcal{D}_1 ;\ldots; \mathcal{D}_n ; \cap\rangle :: \Gamma,x{:}\mathtt{S} \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta)$: Let, for all $i \in \underline{n}$,

$$\mathcal{D}'_i \;=\; \mathcal{D}_i\{\mathcal{D}_0/x{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} M\{N/x\} : \mathtt{A}_i \mid \Delta,$$

then $\mathcal{D}\{\mathcal{D}_0/x{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1 ;\ldots; \mathcal{D}'_n ; \cap\rangle :: \Gamma \vdash_{\mathrm{S}} M\{N/x\} : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta.$

Notice that, by the last case,

$$(\langle\cap\rangle :: \Gamma,x{:}\mathtt{S} \vdash_{\mathrm{S}} M : \omega \mid \Delta)\{\mathcal{D}_0/x{:}\mathtt{S}\} \;=\; \langle\cap\rangle :: \Gamma \vdash_{\mathrm{S}} M\{N/x\} : \omega \mid \Delta.$$

Similarly, consider a derivation for the redex $(\mu\alpha.[\beta]P)Q$, shaped like (where $\mathtt{S} = \cap_{\underline{m}}\mathtt{A}_j$, and $\Delta' = \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\Delta$):

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\mathcal{D}_1^1\\ \Gamma' \vdash_{\mathrm{S}} R_1 : \neg(\mathtt{T}_1\times\mathtt{D}_1) \mid \delta_1{:}\mathtt{C}'_1,\Delta'\end{array}}{\Gamma' \vdash_{\mathrm{S}} \mu\delta_1.[\alpha]R_1 : \neg\mathtt{C}'_1 \mid \Delta'}\,(\mu_1) \quad \cdots \quad \dfrac{\begin{array}{c}\mathcal{D}_1^n\\ \Gamma' \vdash_{\mathrm{S}} R_n : \neg(\mathtt{T}_n\times\mathtt{D}_n) \mid \delta_n{:}\mathtt{C}'_n,\Delta'\end{array}}{\Gamma' \vdash_{\mathrm{S}} \mu\delta_n.[\alpha]R_n : \neg\mathtt{C}'_n \mid \Delta'}\,(\mu_1)\\ \mathcal{D}_1\\ \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\Delta\end{array}}{\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]P : \neg(\mathtt{S}\times\mathtt{C}) \mid \beta{:}\mathtt{C}',\Delta}\,(\mu_1) \quad \dfrac{\begin{array}{c}\mathcal{D}_2^i\\ \Gamma \vdash_{\mathrm{S}} Q : \mathtt{A}_i \mid \beta{:}\mathtt{C}',\Delta \quad (\forall i \in \underline{m})\end{array}}{\Gamma \vdash_{\mathrm{S}} Q : \cap_{\underline{m}}\mathtt{A}_j \mid \beta{:}\mathtt{C}',\Delta}\,(\cap)}{\Gamma \vdash_{\mathrm{S}} (\mu\alpha.[\beta]P)Q : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta}\,(App)$$

Then $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$, and $\mathtt{S}\times\mathtt{C} \leq_{\mathrm{S}} \mathtt{T}_i\times\mathtt{D}_i$ for all $i \in \underline{n}$; so $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}_i$ and also $\cap_{\underline{m}}\mathtt{A}_j = \mathtt{S} \leq_{\mathrm{S}} \mathtt{T}_i$, for all $i \in \underline{n}$, so there are $\mathtt{B}^i_1,\ldots,\mathtt{B}^i_k$ such that $\mathtt{T}_i = \cap_{\underline{k}}\mathtt{B}^i_l$, and $\{\mathtt{B}^i_1,\ldots,\mathtt{B}^i_k\} \subseteq \{\mathtt{A}_1,\ldots,\mathtt{A}_n\}$; in particular, by $(\cap)$, $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{T}_i \mid \beta{:}\mathtt{C}',\Delta'$, for all $i \in \underline{n}$, so by weakening also $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{T}_i \mid \delta_i{:}\mathtt{C}'_i,\gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta$. Let $\mathcal{D}_2 = \langle \mathcal{D}_2^1 ;\cdots; \mathcal{D}_2^m ; \cap\rangle :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{S} \mid \beta{:}\mathtt{C}',\Delta$. Derivation reduction will insert $\mathcal{D}_2$ into $\mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\Delta$, and construct (where now $\Delta' = \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta$):

$$\dfrac{\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1^1\\ \Gamma \vdash_{\mathrm{S}} R_1 : \neg(\mathtt{T}_1\times\mathtt{D}_1) \mid \delta_1{:}\mathtt{C}'_1,\Delta'\end{array} \quad \Gamma \vdash_{\mathrm{S}} Q : \mathtt{T}_1 \mid \delta_1{:}\mathtt{C}'_1,\Delta'}{\Gamma \vdash_{\mathrm{S}} R_1Q : \neg\mathtt{D}_1 \mid \delta_1{:}\mathtt{C}'_1,\Delta'}\,(App)}{\Gamma \vdash_{\mathrm{S}} \mu\delta_1.[\gamma]R_1Q : \neg\mathtt{C}'_1 \mid \Delta'}\,(\mu_1) \quad \cdots \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1^n\\ \Gamma \vdash_{\mathrm{S}} R_n : \neg(\mathtt{T}_n\times\mathtt{D}_n) \mid \delta_n{:}\mathtt{C}'_n,\Delta'\end{array} \quad \Gamma \vdash_{\mathrm{S}} Q : \mathtt{T}_n \mid \delta_n{:}\mathtt{C}'_n,\Delta'}{\Gamma \vdash_{\mathrm{S}} R_nQ : \neg\mathtt{D}_n \mid \delta_n{:}\mathtt{C}'_n,\Delta'}\,(App)}{\Gamma \vdash_{\mathrm{S}} \mu\delta_n.[\gamma]R_nQ : \neg\mathtt{C}'_n \mid \Delta'}\,(\mu_1)\\ \vdots\\ \mathcal{D}_1\\ \Gamma \vdash_{\mathrm{S}} P\{Q{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta\end{array}}{\Gamma \vdash_{\mathrm{S}} \mu\gamma.[\beta]P\{Q{\cdot}\gamma/\alpha\} : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta}\,(\mu_1)$$

This leads to the notion of derivation substitution that deals with structural substitution; since this does not actually replace existing structure but rather reorganises it, following what we suggested above, we call it *insertion*.

Definition 5.3 (Derivation insertion) For $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{T} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta$, and $\mathcal{D}_0 :: \Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta$, the derivation

$$\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} M\{N{\cdot}\gamma/\alpha\} : \mathtt{T} \mid \gamma{:}\mathtt{C},\Delta,$$

the result of *inserting* $\mathcal{D}_0$ *at* $\alpha{:}\mathtt{S}$ *in* $\mathcal{D}$, is defined by induction on the structure of derivations by:

$(\langle Ax\rangle :: \Gamma \vdash_{\mathrm{S}} x : \mathtt{T} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta)$: Then $\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} \;\triangleq\; \langle Ax\rangle :: \Gamma \vdash_{\mathrm{S}} x : \mathtt{T} \mid \gamma{:}\mathtt{C},\Delta.$

$(\langle \mathcal{D}_1 :: \Gamma, y{:}\mathtt{R} \vdash_{\mathrm{S}} P : \neg \mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta; Abs\rangle :: \Gamma \vdash_{\mathrm{S}} \lambda y.P : \neg(\mathtt{R}\times\mathtt{D}) \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta)$: Let

$$\mathcal{D}' \;=\; \mathcal{D}_1\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma, y{:}\mathtt{R} \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\Delta.$$

Then $\mathcal{D}\{\mathcal{D}_0/x{:}\mathtt{A}\} \;\triangleq\; \langle \mathcal{D}'; Abs\rangle :: \Gamma \vdash_{\mathrm{S}} (\lambda y.P)\{N{\cdot}\gamma/\alpha\} : \neg(\mathtt{R}\times\mathtt{D}) \mid \gamma{:}\mathtt{C},\Delta$.

$(\langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{R}\times\mathtt{D}) \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta; \mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{R} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta; App\rangle :: \Gamma \vdash_{\mathrm{S}} PQ : \neg\mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta)$: Let

$$\begin{array}{rcl} \mathcal{D}'_1 &=& \mathcal{D}_1\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\} : \neg(\mathtt{R}\times\mathtt{D}) \mid \gamma{:}\mathtt{C},\Delta, \text{ and} \\ \mathcal{D}'_2 &=& \mathcal{D}_2\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} Q\{N{\cdot}\gamma/\alpha\} : \mathtt{R} \mid \gamma{:}\mathtt{C},\Delta, \end{array}$$

then $\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1, \mathcal{D}'_2; App\rangle :: \Gamma \vdash_{\mathrm{S}} (PQ)\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\Delta$.

$(\langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\beta.[\delta]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{S}\times\mathtt{C},\delta{:}\mathtt{C}'',\Delta \textit{ with } \delta \neq \alpha,\ \delta \neq \beta)$: Let

$$\mathcal{D}'_1 \;=\; \mathcal{D}_1\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta.$$

Then $\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\beta.[\delta]P)\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{C}' \mid \gamma{:}\mathtt{C},\delta{:}\mathtt{C}'',\Delta$.

$(\langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{R}\times\mathtt{D}) \mid \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\beta.[\alpha]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{S}\times\mathtt{C},\delta{:}\mathtt{C}'',\Delta)$: Let

$$\mathcal{D}'_1 \;=\; \mathcal{D}_1\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\} : \neg(\mathtt{R}\times\mathtt{D}) \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta.$$

Since $\mathtt{S} \leq_{\mathrm{S}} \mathtt{R}$, by Lemma 3.9 there exists $\mathcal{D}'_0 :: \Gamma \vdash_{\mathrm{S}} N : \mathtt{R} \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta$. Then

$$\begin{array}{rcl} \mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} &\triangleq& \langle\langle \mathcal{D}', \mathcal{D}'_0; App\rangle :: \Gamma \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\}N : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\delta{:}\mathtt{C}'',\Delta; \mu_1\rangle :: \\ && \qquad \Gamma \vdash_{\mathrm{S}} \mu\beta.[\gamma]P\{N{\cdot}\gamma/\alpha\}N : \neg\mathtt{C}' \mid \gamma{:}\mathtt{C},\delta{:}\mathtt{C}'',\Delta. \end{array}$$

$(\langle \mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{S}\times\mathtt{C},\beta{:}\mathtt{C}',\Delta; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\beta.[\beta]P : \neg\mathtt{C}' \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta)$: Let

$$\mathcal{D}'_1 \;=\; \mathcal{D}_1\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} P\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta.$$

Then $\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\beta.[\beta]P)\{N{\cdot}\gamma/\alpha\} : \neg\mathtt{C}' \mid \gamma{:}\mathtt{C},\Delta$.

$(\langle \mathcal{D}_1; \ldots; \mathcal{D}_n; \cap\rangle :: \Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \alpha{:}\mathtt{S}\times\mathtt{C},\Delta)$: Let, for all $i \in \underline{n}$,

$$\mathcal{D}'_i \;=\; \mathcal{D}_i\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\mathrm{S}} M\{N{\cdot}\gamma/\alpha\} : \mathtt{A}_i \mid \gamma{:}\mathtt{C},\Delta,$$

then $\mathcal{D}\{\mathcal{D}_0{\cdot}\gamma/\alpha{:}\mathtt{S}\} \;\triangleq\; \langle \mathcal{D}'_1; \ldots; \mathcal{D}'_n; \cap\rangle :: \Gamma \vdash_{\mathrm{S}} M\{N{\cdot}\gamma/\alpha\} : \cap_{\underline{n}}\mathtt{A}_i \mid \gamma{:}\mathtt{C},\Delta$.

Below we will allow ourselves the notation $\mathcal{D}\{\overrightarrow{\mathcal{D}}{\cdot}\gamma/\alpha{:}\mathtt{C}\}$, with $\mathtt{C} = \mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \Omega$, which stands for

$$\mathcal{D}\{\mathcal{D}_1{\cdot}\gamma_1/\alpha{:}\mathtt{S}_1\}\{\mathcal{D}_2{\cdot}\gamma_2/\gamma_1{:}\mathtt{S}_2\}\cdots\{\mathcal{D}_n{\cdot}\gamma/\gamma_{n-1}{:}\mathtt{S}_n\}.$$

Before coming to the definition of derivation reduction, we need to define the notion of 'position of a sub-derivation in a derivation'. This is needed in Definition 5.5 to make sure that, when contracting a redex in one sub-derivation (branch) in a derivation ending with $(\cap)$, all its 'siblings' in neighbouring branches are contracted as well.

Definition 5.4 Let $\mathcal{D}$ be a derivation, and $\mathcal{D}'$ be a sub-derivation of $\mathcal{D}$. The position p of $\mathcal{D}'$ in $\mathcal{D}$, an element of the set of strings over $\{1,2\}$, is defined by:

i) If $\mathcal{D}' = \mathcal{D}$, then $p = \varepsilon$, the empty string.

ii) If the position of $\mathcal{D}'$ in $\mathcal{D}_1$ is q, and $\mathcal{D} = \langle \mathcal{D}_1; Abs\rangle$, $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2; App\rangle$, $\mathcal{D} = \langle \mathcal{D}_1; \mu_1\rangle$, or $\mathcal{D} = \langle \mathcal{D}_1; \mu_2\rangle$, then $p = 1q$.

iii) If the position of $\mathcal{D}'$ in $\mathcal{D}_2$ is q, and $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2; App\rangle$, then $p = 2q$.

iv) If the position of $\mathcal{D}'$ in $\mathcal{D}_i$ ($i \in \underline{n}$) is q, and $\mathcal{D} = \langle \mathcal{D}_1; \ldots; \mathcal{D}_n; \cap\rangle$, then $p = q$.

We can now define a notion of reduction on derivations; notice that this reduction corresponds to contracting a redex in the term involved only if that redex appears in the derivation in a sub-derivation with type different from ω.

Definition 5.5 (Derivation reduction) We say that the derivation $\mathcal{D} :: \Gamma \vdash_{\rm S} M : \mathtt{S} \mid \Delta$ *reduces at position* p *with redex* $\mathtt{R}$ *to* $\mathcal{D}' :: \Gamma \vdash_{\rm S} N : \mathtt{S} \mid \Delta$, if and only if:

$(\mathtt{S} = \neg\mathtt{C})$: $(M = (\lambda x.P)Q)$: Let $\mathcal{D}_1 :: \Gamma, x{:}\mathtt{T} \vdash_{\rm S} P : \neg\mathtt{C} \mid \Delta$ and $\mathcal{D}_2 :: \Gamma \vdash_{\rm S} Q : \mathtt{T} \mid \Delta$, then

$$\mathcal{D} \;=\; \langle\langle \mathcal{D}_1 ; Abs\rangle :: \Gamma \vdash_{\rm S} \lambda x.P : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta ; \mathcal{D}_2 ; App\rangle :: \Gamma \vdash_{\rm S} (\lambda x.P)Q : \neg\mathtt{C} \mid \Delta$$

reduces at position ε with redex $(\lambda x.P)Q$ to $\mathcal{D}_1\,\{\mathcal{D}_2/x{:}\mathtt{S}\} :: \Gamma \vdash_{\rm S} P\{Q/x\} : \neg\mathtt{C} \mid \Delta$.

$(M = (\mu\alpha.[\alpha]P)Q)$: Let $\mathcal{D}_1 :: \Gamma \vdash_{\rm S} P : \neg(\mathtt{T}\times\mathtt{D}) \mid \alpha{:}\mathtt{R}\times\mathtt{C}, \Delta$ and $\mathcal{D}_2 :: \Gamma \vdash_{\rm S} Q : \mathtt{R} \mid \Delta$ with $\mathtt{R} \leq_{\rm S} \mathtt{T}$ (notice that then there exists $\mathcal{D}'_2 :: \Gamma \vdash_{\rm S} Q : \mathtt{T} \mid \gamma{:}\mathtt{C}, \Delta$ with $\mathcal{D}_2 \leq_{\rm S} \mathcal{D}'_2$) and $\mathtt{C} \leq_{\rm S} \mathtt{D}$, then

$$\mathcal{D} \;=\; \langle\langle \mathcal{D}_1 ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]P : \neg(\mathtt{R}\times\mathtt{C}) \mid \Delta ; \mathcal{D}_2 ; App\rangle :: \Gamma \vdash_{\rm S} (\mu\alpha.[\alpha]P)Q : \neg\mathtt{C} \mid \Delta$$

reduces at position ε with redex $(\mu\alpha.[\alpha]P)Q$ to

$$\begin{array}{l}\langle\langle \mathcal{D}_1\,\{\mathcal{D}_2{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\rm S} P\{Q{\cdot}\gamma/\alpha\} : \neg(\mathtt{T}\times\mathtt{D}) \mid \gamma{:}\mathtt{C}, \Delta ; \mathcal{D}'_2 :: \Gamma \vdash_{\rm S} Q : \mathtt{T} \mid \gamma{:}\mathtt{C}, \Delta ; App\rangle :: \\ \qquad\qquad \Gamma \vdash_{\rm S} P\{Q{\cdot}\gamma/\alpha\}Q : \neg\mathtt{D} \mid \gamma{:}\mathtt{C}, \Delta ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\gamma]P\{Q{\cdot}\gamma/\alpha\}Q : \neg\mathtt{C} \mid \Delta\end{array}$$

$(M = (\mu\alpha.[\beta]P)Q$, with $\alpha \neq \beta)$: Let $\mathcal{D}_1 :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{R}\times\mathtt{C}, \beta{:}\mathtt{C}', \Delta$ and $\mathcal{D}_2 :: \Gamma \vdash_{\rm S} Q : \mathtt{R} \mid \beta{:}\mathtt{C}', \Delta$ with $\mathtt{C}' \leq_{\rm S} \mathtt{D}$, then

$$\mathcal{D} \;=\; \langle\langle \mathcal{D}_1 ; \mu_1\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\beta]P : \neg(\mathtt{R}\times\mathtt{C}) \mid \beta{:}\mathtt{C}', \Delta ; \mathcal{D}_2 ; App\rangle :: \Gamma \vdash_{\rm S} (\mu\alpha.[\beta]P)Q : \neg\mathtt{C} \mid \beta{:}\mathtt{C}', \Delta$$

reduces to at position ε with redex $(\mu\alpha.[\beta]P)Q$ to

$$\langle \mathcal{D}_1\,\{\mathcal{D}_2{\cdot}\gamma/\alpha{:}\mathtt{S}\} :: \Gamma \vdash_{\rm S} P\{Q{\cdot}\gamma/\alpha\} : \neg\mathtt{D} \mid \gamma{:}\mathtt{C}, \beta{:}\mathtt{C}', \Delta ; \mu_1\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\beta]P\{Q{\cdot}\gamma/\alpha\} : \neg\mathtt{C} \mid \beta{:}\mathtt{C}', \Delta$$

$(M = \mu\alpha.[\alpha]\mu\gamma.[\gamma]P)$: Let $\mathcal{D} :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \gamma{:}\mathtt{C}'', \Delta$, with $\mathtt{C} \leq_{\rm S} \mathtt{C}'' \leq_{\rm S} \mathtt{D}$, then

$$\langle\langle \mathcal{D} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\gamma]P : \neg\mathtt{C}'' \mid \alpha{:}\mathtt{C}, \Delta ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]\mu\gamma.[\gamma]P : \neg\mathtt{C} \mid \Delta$$

reduces at position ε with redex $\mu\alpha.[\alpha]\mu\gamma.[\gamma]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]P\{\alpha/\gamma\} : \neg\mathtt{C} \mid \Delta.$$

$(M = \mu\alpha.[\alpha]\mu\gamma.[\alpha]P)$: Let $\mathcal{D} :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \gamma{:}\mathtt{C}'', \Delta$, with $\mathtt{C} \leq_{\rm S} \mathtt{D}$ and $\mathtt{C} \leq_{\rm S} \mathtt{C}'$, then

$$\langle\langle \mathcal{D} ; \mu_1\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\alpha]P : \neg\mathtt{C}'' \mid \alpha{:}\mathtt{C}, \Delta ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]\mu\gamma.[\alpha]P : \neg\mathtt{C} \mid \Delta$$

reduces at position ε with redex $\mu\alpha.[\alpha]\mu\gamma.[\alpha]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]P\{\alpha/\gamma\} : \neg\mathtt{C} \mid \Delta.$$

$(M = \mu\alpha.[\alpha]\mu\gamma.[\delta]P$, with $\delta \neq \alpha$ and $\delta \neq \gamma)$: Let $\mathcal{D} :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \gamma{:}\mathtt{C}'', \delta{:}\mathtt{D}', \Delta$, with $\mathtt{D}' \leq_{\rm S} \mathtt{D}$ and $\mathtt{C} \leq_{\rm S} \mathtt{C}'$, then

$$\langle\langle \mathcal{D} ; \mu_1\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\delta]P : \neg\mathtt{C}'' \mid \alpha{:}\mathtt{C}, \delta{:}\mathtt{D}', \Delta ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\alpha]\mu\gamma.[\delta]P : \neg\mathtt{C} \mid \delta{:}\mathtt{D}', \Delta$$

reduces at position ε with redex $\mu\alpha.[\alpha]\mu\gamma.[\delta]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\delta]P\{\alpha/\gamma\} : \neg\mathtt{C} \mid \delta{:}\mathtt{D}', \Delta.$$

$(M = \mu\alpha.[\beta]\mu\gamma.[\gamma]P$, with $\beta \neq \alpha)$: Let $\mathcal{D} :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \gamma{:}\mathtt{C}'', \Delta$, with $\mathtt{C}' \leq_{\rm S} \mathtt{C}'' \leq_{\rm S} \mathtt{D}$, then

$$\langle\langle \mathcal{D} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\gamma.[\gamma]P : \neg\mathtt{C}'' \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \Delta ; \mu_1\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\beta]\mu\gamma.[\gamma]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}', \Delta$$

reduces at position ε with redex $\mu\alpha.[\beta]\mu\gamma.[\gamma]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\rm S} \mu\alpha.[\gamma]P\{\beta/\gamma\} : \neg\mathtt{C} \mid \beta{:}\mathtt{C}', \Delta.$$

$(M = \mu\alpha.[\beta]\mu\gamma.[\alpha]P$, with $\beta \neq \alpha)$: Let $\mathcal{D} :: \Gamma \vdash_{\rm S} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \gamma{:}\mathtt{C}'', \Delta$, with $\mathtt{C}' \leq_{\rm S} \mathtt{D}$ and $\mathtt{C} \leq_{\rm S} \mathtt{C}''$, then

$$\langle\langle \mathcal{D} ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\gamma.[\alpha]P : \neg \mathtt{C}'' \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \Delta ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]\mu\gamma.[\alpha]P : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \Delta$$

reduces at position ε with redex $\mu\alpha.[\beta]\mu\gamma.[\alpha]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]P\{\beta/\gamma\} : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \Delta.$$

($M = \mu\alpha.[\beta]\mu\gamma.[\beta]P$, with $\beta \neq \alpha$): Let $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} P : \neg \mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \gamma{:}\mathtt{C}'', \Delta$, with $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$ and $\mathtt{C} \leq_{\mathrm{S}} \mathtt{C}''$, then

$$\langle\langle \mathcal{D} ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\gamma.[\beta]P : \neg \mathtt{C}'' \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \Delta ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]\mu\gamma.[\beta]P : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \Delta$$

reduces at position ε with redex $\mu\alpha.[\beta]\mu\gamma.[\beta]P$ to

$$\langle \mathcal{D}\{\alpha/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]P\{\beta/\gamma\} : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \Delta.$$

($M = \mu\alpha.[\beta]\mu\gamma.[\delta]P$, with $\alpha,\beta,\gamma,\delta$ all different): Let $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} P : \neg \mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \gamma{:}\mathtt{C}'', \delta{:}\mathtt{D}', \Delta$, with $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$ and $\mathtt{C} \leq_{\mathrm{S}} \mathtt{C}''$, then

$$\begin{array}{r}\langle\langle \mathcal{D} ; \mu_1\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\gamma.[\delta]P : \neg \mathtt{C}'' \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \delta{:}\mathtt{D}', \Delta ; \mu_1\rangle :: \\ \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]\mu\gamma.[\delta]P : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \delta{:}\mathtt{D}', \Delta\end{array}$$

reduces at position ε with redex $\mu\alpha.[\beta]\mu\gamma.[\delta]P$ to

$$\langle \mathcal{D}\{\beta/\gamma\} ; \mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\delta]P\{\beta/\gamma\} : \neg \mathtt{C} \mid \beta{:}\mathtt{C}', \delta{:}\mathtt{D}', \Delta.$$

($\mathtt{S} = \cap_{\underline{n}}\mathtt{A}_i$): If $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta$, then, for every $i \in \underline{n}$, there exists $\mathcal{D}_i :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta$ such that $\mathcal{D} = \langle \mathcal{D}_1 ; \ldots ; \mathcal{D}_n ; \cap\rangle$. If there is an $i \in \underline{n}$ such that $\mathcal{D}_i$ reduces to $\mathcal{D}'_i$ at position p with redex R, then, for all $1 \leq i \neq j \leq n$, either:

a) there is no redex at position p because there is no sub-derivation at that position. Since R is a sub-term of M, it has to be part of a term that is typed with ω in $\mathcal{D}_j$. Let $\mathsf{R} \rightarrow_{\beta\mu} \mathsf{R}'$ and $\mathcal{D}'_j = \mathcal{D}_j\{\mathsf{R}'/\mathsf{R}\}$ (*i.e.* $\mathcal{D}_j$ where each R is replaced by R'), or

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

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

(Inductive cases): If $\mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$ reduces at position p with redex R to $\mathcal{D}'_1 :: \Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta$, and

- $\mathtt{S} = \neg\mathtt{C}$, $\Gamma = \Gamma', x{:}\mathtt{T}$; then $\mathcal{D} = \langle \mathcal{D}_1 ; Abs\rangle :: \Gamma' \vdash_{\mathrm{S}} \lambda x.M : \neg(\mathtt{T}\times\mathtt{C}) \mid \Delta$ reduces at position $1p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}'_1 ; Abs\rangle :: \Gamma' \vdash_{\mathrm{S}} \lambda x.N : \neg(\mathtt{T}\times\mathtt{C}) \mid \Delta$.
- $\mathtt{S} = \neg(\mathtt{T}\times\mathtt{C})$ and $\mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} Q : \mathtt{T} \mid \Delta$; then $\mathcal{D} = \langle \mathcal{D}_1, \mathcal{D}_2 ; App\rangle :: \Gamma \vdash_{\mathrm{S}} MQ : \neg\mathtt{C} \mid \Delta$ reduces at position $1p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}'_1 ; \mathcal{D}_2 ; App\rangle :: \Gamma \vdash_{\mathrm{S}} NQ : \neg\mathtt{C} \mid \Delta$.
- Let $\mathcal{D}_2 :: \Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta$; then $\mathcal{D} = \langle \mathcal{D}_2, \mathcal{D}_1 ; App\rangle :: \Gamma \vdash_{\mathrm{S}} PM : \neg\mathtt{C} \mid \Delta$ reduces at position $2p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}_2 ; \mathcal{D}'_1 ; App\rangle :: \Gamma \vdash_{\mathrm{S}} PN : \neg\mathtt{C} \mid \Delta$.
- $\mathtt{S} = \neg\mathtt{C}$; then $\mathcal{D} = \langle \mathcal{D}_1 ; \mu_i\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]M : \neg\mathtt{C} \mid \beta{:}\mathtt{D}, \Delta$ (with perhaps $\alpha = \beta$) reduces at position $1p$ with redex R to $\mathcal{D}' = \langle \mathcal{D}'_1 ; \mu_i\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]N : \neg\mathtt{C} \mid \beta{:}\mathtt{D}, \Delta$.

We write $\mathcal{D} \rightarrow_{\mathrm{DER}} \mathcal{D}'$ if there exists a position p and redex R such that $\mathcal{D}$ reduces to $\mathcal{D}'$ at position p with redex R and use $\rightarrow^*_{\mathrm{DER}}$ for its transitive closure.

Example 5.6 • Let $M = \lambda x.f(xx)$ and $\mathtt{A} = \neg(\omega\times\Omega)$; for $\lambda f.MM$ we can construct:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f{:}\mathtt{A}, x{:}\omega \vdash_{\mathrm{S}} f : \neg(\omega\times\Omega) \mid \varnothing}\,(Ax) \quad \dfrac{}{f{:}\mathtt{A}, x{:}\omega \vdash_{\mathrm{S}} xx : \omega \mid \varnothing}\,(\cap)}{f{:}\mathtt{A}, x{:}\omega \vdash_{\mathrm{S}} f(xx) : \neg\Omega \mid \varnothing}\,(App)}{f{:}\mathtt{A} \vdash_{\mathrm{S}} M : \neg(\omega\times\Omega) \mid \varnothing}\,(Abs) \quad \dfrac{}{f{:}\mathtt{A} \vdash_{\mathrm{S}} M : \omega \mid \varnothing}\,(\cap)}{\dfrac{f{:}\mathtt{A} \vdash_{\mathrm{S}} MM : \neg\Omega \mid \varnothing}{\varnothing \vdash_{\mathrm{S}} \lambda f.MM : \neg(\mathtt{A}\times\Omega) \mid \varnothing}\,(Abs)}\,(App)$$

Notice that this derivation has a cut and reduces to a derivation for $\lambda f.(f(xx)\{M/x\}) = \lambda f.f(MM)$. Also, the derivation substitution $\{\langle\cap\rangle :: f{:}\mathsf{A} \vdash_{\mathsf{s}} M : \omega \mid \emptyset \,/\, x{:}\omega\}$ that is applied to $\mathcal{D} :: f{:}\mathsf{A}, x{:}\omega \vdash_{\mathsf{s}} f(xx) : \neg\Omega \mid \emptyset$ has no effect, other than replacing x by $M = \lambda x.f(xx)$, and creates:

$$\dfrac{\dfrac{\dfrac{}{f{:}\mathsf{A} \vdash_{\mathsf{s}} f : \neg(\omega\times\Omega) \mid \emptyset}\,(Ax) \qquad \dfrac{}{f{:}\mathsf{A} \vdash_{\mathsf{s}} MM : \omega \mid \emptyset}\,(\cap)}{f{:}\mathsf{A} \vdash_{\mathsf{s}} f(MM) : \neg\Omega \mid \emptyset}\,(App)}{\emptyset \vdash_{\mathsf{s}} \lambda f.f(MM) : \neg(\mathsf{A}\times\Omega) \mid \emptyset}\,(Abs)$$

- Take $\mathsf{A} = \neg(\omega\times\Omega)$, $\mathsf{B} = \neg(\neg\Omega\times\Omega)$, and $\Gamma = f{:}\mathsf{A}\cap\mathsf{B}$, then we can construct the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma,x{:}\mathsf{A} \vdash_{\mathsf{s}} f : \neg(\neg\Omega\times\Omega) \mid \emptyset}\,(Ax) \qquad \begin{matrix}\dfrac{\dfrac{}{\Gamma,x{:}\mathsf{A} \vdash_{\mathsf{s}} x : \neg(\omega\times\Omega) \mid \emptyset}\,(Ax) \quad \dfrac{}{\Gamma,x{:}\mathsf{A} \vdash_{\mathsf{s}} x : \omega \mid \emptyset}\,(\cap)}{\Gamma,x{:}\mathsf{A} \vdash_{\mathsf{s}} xx : \neg\Omega \mid \emptyset}\,(App) \\ \vdots\end{matrix}}{\Gamma,x{:}\mathsf{A} \vdash_{\mathsf{s}} f(xx) : \neg\Omega \mid \emptyset}\,(App)}{\Gamma \vdash_{\mathsf{s}} M : \neg(\neg(\omega\times\Omega)\times\Omega) \mid \emptyset}\,(Abs) \qquad \dfrac{\dfrac{\dfrac{}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} f : \neg(\omega\times\Omega) \mid \emptyset}\,(Ax) \quad \dfrac{}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} xx : \omega \mid \emptyset}\,(\cap)}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} f(xx) : \neg\Omega \mid \emptyset}\,(App)}{\Gamma \vdash_{\mathsf{s}} M : \neg(\omega\times\Omega) \mid \emptyset}\,(Abs)}{\Gamma \vdash_{\mathsf{s}} MM : \neg\Omega \mid \emptyset}\,(App)}{\emptyset \vdash_{\mathsf{s}} \lambda f.MM : \neg(\mathsf{A}\cap\mathsf{B}\times\Omega) \mid \emptyset}\,(Abs)$$

Again, there is one cut, and contracting it creates:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\mathsf{s}} f : \neg(\neg\Omega\times\Omega) \mid \emptyset}\,(Ax) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} f : \neg(\omega\times\Omega) \mid \emptyset}\,(Ax) \quad \dfrac{}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} xx : \omega \mid \emptyset}\,(\cap)}{\Gamma,x{:}\omega \vdash_{\mathsf{s}} f(xx) : \neg\Omega \mid \emptyset}\,(App)}{\Gamma \vdash_{\mathsf{s}} M : \neg(\omega\times\Omega) \mid \emptyset}\,(Abs) \quad \dfrac{}{\Gamma \vdash_{\mathsf{s}} M : \omega \mid \emptyset}\,(\cap)}{\Gamma \vdash_{\mathsf{s}} MM : \neg\Omega \mid \emptyset}\,(App)}{\Gamma \vdash_{\mathsf{s}} f(MM) : \neg\Omega \mid \emptyset}\,(App)}{\emptyset \vdash_{\mathsf{s}} \lambda f.f(MM) : \neg(\mathsf{A}\cap\mathsf{B}\times\Omega) \mid \emptyset}\,(Abs)$$

Notice that x gets replaced by $M = \lambda x.f(xx)$, and that the right-hand subderivation for $\Gamma \vdash_{\mathsf{s}} \lambda x.f(xx) : \mathsf{A} \mid \emptyset$ takes the place of the sub-derivation $\langle Ax\rangle :: \Gamma, x{:}\mathsf{A} \vdash_{\mathsf{s}} x : \mathsf{A} \mid \emptyset$.

The following lemma states that derivation reduction follows term reduction.

Lemma 5.7 Let $\mathcal{D} :: \Gamma \vdash_{\mathsf{s}} M : \mathsf{S} \mid \Delta$, and $\mathcal{D} \to^*_{\mathrm{DER}} \mathcal{D}' :: \Gamma \vdash_{\mathsf{s}} N : \mathsf{S} \mid \Delta$, then $M \to^*_{\beta\mu} N$.

Proof: Implied by Definition 5.5.

We say that $\mathcal{D}$ is *normalisable* if there exists a redex-free $\mathcal{D}'$ such that $\mathcal{D} \to^*_{\mathrm{DER}} \mathcal{D}'$, and that $\mathcal{D}$ is *strongly normalisable* ($SN(\mathcal{D})$) if all reduction sequences starting in $\mathcal{D}$ are of finite length.

The following states some standard properties of strong normalisation.

Lemma 5.8 *i) If $SN(\langle \mathcal{D}_1 ; \mathcal{D}_2 ; App\rangle)$, then $SN(\mathcal{D}_1)$ and $SN(\mathcal{D}_2)$.*

ii) If $SN(\mathcal{D}_1 :: \Gamma_1 \vdash_{\mathsf{s}} xM_1\cdots M_n : \neg(\mathsf{S}\times\mathsf{C}) \mid \Delta)$ and $SN(\mathcal{D}_2 :: \Gamma_2 \vdash_{\mathsf{s}} N : \mathsf{S} \mid \Delta)$, then $SN(\langle \mathcal{D}_1 ; \mathcal{D}_2 ; App\rangle :: \Gamma_1, \Gamma_2 \vdash_{\mathsf{s}} xM_1\cdots M_n N : \neg\mathsf{C} \mid \Delta)$.

iii) $\forall i \in \underline{n}\ [SN(\mathcal{D}_i :: \Gamma \vdash_{\mathsf{s}} M : \mathsf{A}_i \mid \Delta)]$ if and only if $SN(\langle \mathcal{D}_1 ; \ldots ; \mathcal{D}_n ; \cap\rangle :: \Gamma \vdash_{\mathsf{s}} M : \cap_{\underline{n}} \mathsf{A}_i \mid \Delta)$.

iv) If $SN(\langle \cdots \langle \mathcal{D}_1\{\mathcal{D}_2/y{:}\mathsf{T}\}\rangle \cdots ; App\rangle :: \Gamma \vdash_{\mathsf{s}} M\{N/x\}\vec{P} : \mathsf{S} \mid \Delta)$ and $SN(\mathcal{D}_2 :: \Gamma \vdash_{\mathsf{s}} N : \mathsf{T} \mid \Delta)$, then $SN(\langle \cdots \langle\langle \mathcal{D}_1 ; Abs\rangle ; \mathcal{D}_2 ; App\rangle \cdots ; App\rangle :: \Gamma \vdash_{\mathsf{s}} (\lambda y.M)N\vec{P} : \mathsf{S} \mid \Delta)$.

v) If $SN(\mathcal{D} :: \Gamma \vdash_{\mathsf{s}} P : \neg\mathsf{D} \mid \delta{:}\mathsf{C}, \Delta)$, and $\mathsf{C} \leq_{\mathsf{s}} \mathsf{D}$, then $SN(\mathcal{D} :: \Gamma \vdash_{\mathsf{s}} \mu\delta.[\delta]P : \neg\mathsf{C} \mid \Delta)$.

vi) If $SN(\mathcal{D} :: \Gamma \vdash_{\mathsf{s}} P : \neg\mathsf{D} \mid \delta{:}\mathsf{C}, \gamma{:}\mathsf{C}', \Delta)$ and $\mathsf{C} \leq_{\mathsf{s}} \mathsf{D}$, then $SN(\mathcal{D} :: \Gamma \vdash_{\mathsf{s}} \mu\gamma.[\delta]P : \neg\mathsf{C}' \mid \delta{:}\mathsf{C}, \Delta)$.

vii) If $SN(\langle \cdots \langle\langle \mathcal{D}_1\{\mathcal{D}_2{\cdot}\gamma/\alpha{:}\mathsf{T}\} ; \mathcal{D}'_2 ; App\rangle ; \mu_2\rangle \cdots ; App\rangle :: \Gamma \vdash_{\mathsf{s}} \mu\gamma.[\gamma]M\{N{\cdot}\gamma/\alpha\}N\vec{P} : \mathsf{S} \mid \Delta)$ such that $\mathcal{D}_2 \leq_{\mathsf{s}} \mathcal{D}'_2$ and $SN(\mathcal{D}_2 :: \Gamma \vdash_{\mathsf{s}} N : \mathsf{T} \mid \Delta)$, then $SN(\langle \cdots \langle\langle \mathcal{D}_1 ; \mu_2\rangle ; \mathcal{D}_2 ; App\rangle \cdots ; App\rangle :: \Gamma \vdash_{\mathsf{s}}$

$(\mu\alpha.[\alpha]M)N\vec{P} : \mathtt{S} \mid \Delta)$.

viii) *If* $SN(\langle \cdots \langle \mathcal{D}_1\{\mathcal{D}_2\cdot\gamma/\alpha{:}\mathtt{C}\}\,;\mu_2\rangle \cdots ;App\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\gamma.[\beta]M\{N\cdot\gamma/\alpha\}\vec{P} : \mathtt{S} \mid \Delta)$ *with* $\mathtt{C} = \mathtt{T}_1 \times \cdots \times \mathtt{T}_n \times \Omega$, *and* $SN(\mathcal{D}_2^i :: \Gamma \vdash_{\mathrm{S}} N_i : \mathtt{T}_i \mid \Delta)$ *for all* $i \in \underline{n}$, *then*

$$SN(\langle \cdots \langle \cdots \langle \langle \mathcal{D}_1\,;\mu_2\rangle\,;\mathcal{D}_2^1;App\rangle \cdots ;\mathcal{D}_2^n;App\rangle \cdots ;App\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\alpha.[\beta]M)\vec{N}\vec{P} : \mathtt{S} \mid \Delta).$$

ix) *If* $SN(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} P : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \gamma{:}\mathtt{C}'', \delta{:}\mathtt{D}', \Delta)$, *with* $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$ *and* $\mathtt{C} \leq_{\mathrm{S}} \mathtt{C}''$, *then* $SN(\langle\langle \mathcal{D};r_1\rangle\,;r_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]\mu\gamma.[\delta]P : \neg\mathtt{C} \mid \beta{:}\mathtt{C}', \delta{:}\mathtt{D}', \Delta)$, *where* $r_i \in \{\mu_1, \mu_2\}$ *for* $i \in \underline{2}$.

Proof: Easy, by Definition 5.5. □

6 Strong normalisation of derivation reduction

In this subsection, we will prove a strong normalisation result for derivation reduction. In order to prove that each derivation in '$\vdash_{\mathrm{S}}$' is strongly normalisable with respect to $\rightarrow_{\mathrm{DER}}$, a notion of computable derivation will be introduced. We will show that all computable derivations are strongly normalisable with respect to derivation reduction, and that all derivations in '$\vdash_{\mathrm{S}}$' are computable.

Definition 6.1 (COMPUTABILITY PREDICATE) $Comp(\mathcal{D})$ is defined inductively over the structure of types by:

$$\begin{array}{lcl}
Comp(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \neg\Omega \mid \Delta) & \Longleftrightarrow & SN(\mathcal{D}) \\
Comp(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta) & \Longleftrightarrow & \\
\quad \forall \mathcal{D}' \, [Comp(\mathcal{D}' :: \Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta) \Rightarrow Comp(\langle \mathcal{D};\mathcal{D}';App\rangle :: \Gamma \vdash_{\mathrm{S}} MN : \neg\mathtt{C} \mid \Delta)] & & \\
Comp(\langle \mathcal{D}_1;\ldots;\mathcal{D}_n;\cap\rangle :: \Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}}\mathtt{A}_i \mid \Delta) & \Longleftrightarrow & \forall i \in \underline{n} \; [Comp(\mathcal{D}_i :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta)]
\end{array}$$

By abuse of notation we will use the abbreviation $\Gamma \vdash_{\mathrm{S}} \vec{Q}_{\underline{n}} : \mathtt{C} \mid \Delta$, which stands for $\Gamma \vdash_{\mathrm{S}} Q_1 : \mathtt{S}_1 \mid \Delta, \ldots, \Gamma \vdash_{\mathrm{S}} Q_n : \mathtt{S}_n \mid \Delta$, if $\mathtt{C} = \mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \Omega$; we will also write $Comp(\vec{\mathcal{D}}_{\underline{n}} :: \Gamma \vdash_{\mathrm{S}} \vec{Q}_{\underline{n}} : \mathtt{C} \mid \Delta)$ to denote the sequence of statements $Comp(\mathcal{D}_1 :: \Gamma \vdash_{\mathrm{S}} Q_1 : \mathtt{S}_1 \mid \Delta), \ldots, Comp(\mathcal{D}_n :: \Gamma \vdash_{\mathrm{S}} Q_n : \mathtt{S}_n \mid \Delta)$.

Notice that, as a special case for the third rule, we get $Comp(\langle\cap\rangle :: \Gamma \vdash_{\mathrm{S}} M : \omega \mid \Delta)$. Moreover, we did not define computability for derivations for context switches $\mu\alpha.[\beta]M$, since the types involved in that step are not related by syntactic sub-typing. Rather, computability of context switches is an indirectly inferred property in the proof of the Replacement Theorem 6.5.

We can show that computability is closed for weakening and '$\leq_{\mathrm{S}}$':

Lemma 6.2 *i)* *If* $Comp(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta)$, *and* $\Gamma' \supseteq \Gamma$, $\Delta' \supseteq \Delta$, *then* $Comp(\mathcal{D} :: \Gamma' \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta')$.

ii) *If* $Comp(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta)$, $\Gamma' \leq_{\mathrm{S}} \Gamma$, $\Delta' \leq_{\mathrm{S}} \Delta$, $\mathtt{S} \leq_{\mathrm{S}} \mathtt{S}'$, *then* $Comp(\mathcal{D} :: \Gamma' \vdash_{\mathrm{S}} M : \mathtt{S}' \mid \Delta')$.

Proof: By straightforward induction on the structure of types. □

We will now prove 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.

Lemma 6.3 *i)* *If* $Comp(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta)$, *then* $SN(\mathcal{D})$.

ii) *If* $SN(\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} xM_1 \cdots M_m : \mathtt{S} \mid \Delta)$, *then* $Comp(\mathcal{D})$.

Proof: By simultaneous induction on the structure of types.

($\mathtt{S} = \neg\Omega$): Directly by Definition 6.1.

($\mathtt{S} = \neg(\mathtt{T}\times\mathtt{C})$): *i)* Let x be a variable not appearing in Γ and M, and $\mathcal{D}' = \langle \vec{\mathcal{D}_i}\,;\cap\rangle :: \Gamma, x{:}\mathtt{T} \vdash_{\mathrm{S}} x : \mathtt{T} \mid \Delta$, then, by induction (*ii*), $Comp(\mathcal{D}')$. Since $Comp(\mathcal{D})$, by Lemma 6.2, also

$Comp(\mathcal{D} :: \Gamma, x{:}\mathsf{T} \vdash_{\mathrm{s}} M : \neg(\mathsf{T} \times \mathsf{C}) \mid \Delta)$; then $Comp(\langle \mathcal{D}; \mathcal{D}'; App\rangle :: \Gamma, x{:}\mathsf{T} \vdash_{\mathrm{s}} Mx : \neg\mathsf{C} \mid \Delta)$ by Definition 6.1. Then, by induction (*i*), $SN(\langle \mathcal{D}; \mathcal{D}'; App\rangle)$, and $SN(\mathcal{D})$ follows by Lemma 5.8:(*i*).

ii) Assume $Comp(\mathcal{D}' :: \Gamma \vdash_{\mathrm{s}} N : \mathsf{T} \mid \Delta)$, then by induction (*i*), $SN(\mathcal{D}')$. By Lemma 5.8:(*ii*) we have $SN(\langle \mathcal{D}; \mathcal{D}'; App\rangle :: \Gamma \cap \Gamma' \vdash_{\mathrm{s}} xM_1 \cdots M_m N : \neg\mathsf{C} \mid \Delta)$. Then by induction (*ii*) we have $Comp(\langle \mathcal{D}; \mathcal{D}'; App\rangle)$, so by Definition 6.1, $Comp(\mathcal{D})$.

$(\mathsf{S} = \cap_{\underline{n}} \mathsf{A}_i)$: Easy, using Definition 6.1, Lemma 5.8:(*iii*), and induction. $\square$

The Replacement Theorem (6.5) shows that replacing sub-derivations for term variables by computable derivations, and supplying names with computable derivations through adding applications in a derivation, yields a computable derivation. Before coming to this result, first an auxiliary lemma has to be proved, that formulates that $Comp(\cdot)$ is closed for subject-expansion with respect to derivation reduction for β-redexes.

Lemma 6.4 If $Comp(\mathcal{D}_1[\mathcal{D}'/x{:}\mathsf{T}] :: \Gamma \vdash_{\mathrm{s}} M\{N/x\}\vec{P} : \mathsf{A} \mid \Delta)$ *and* $Comp(\mathcal{D}_2 :: \Gamma \vdash_{\mathrm{s}} N : \mathsf{S} \mid \Delta)$*, then* $Comp(\mathcal{D}_3 :: \Gamma \vdash_{\mathrm{s}} (\lambda x.M)N\vec{P} : \mathsf{A} \mid \Delta)$.

Proof: By induction on the structure of types.

$(\mathsf{A} = \neg\Omega)$: From $Comp(\mathcal{D}_1 :: \Gamma \vdash_{\mathrm{s}} M\{N/x\}\vec{P} : \neg\Omega \mid \Delta)$ by Definition 6.1 we have $SN(\mathcal{D}_1)$. Since also $Comp(\mathcal{D}_2 :: \Gamma \vdash_{\mathrm{s}} N : \mathsf{S} \mid \Delta)$, also $SN(\mathcal{D}_2)$. Then by Lemma 5.8:(*iv*) there exists $\mathcal{D}_3 :: \Gamma \vdash_{\mathrm{s}} (\lambda x.M)N\vec{P} : \neg\Omega \mid \Delta$ such that $SN(\mathcal{D}_3)$, so by Definition 6.1 $Comp(\mathcal{D}_3)$.

$(\mathsf{A} = \neg(\mathsf{S} \times \mathsf{C}))$: Assume $Comp(\mathcal{D}_4 :: \Gamma \vdash_{\mathrm{s}} Q : \mathsf{S} \mid \Delta)$; since we have $Comp(\mathcal{D}_1)$, by Definition 6.1 we get $Comp(\langle \mathcal{D}_1; \mathcal{D}_4; App\rangle :: \Gamma \vdash_{\mathrm{s}} M\{N/x\}\vec{P}Q : \neg\mathsf{C} \mid \Delta)$. Then by induction we have also $Comp(\langle \mathcal{D}_3; \mathcal{D}_4; App\rangle :: \Gamma \vdash_{\mathrm{s}} (\lambda x.M)N\vec{P}Q : \neg\mathsf{C} \mid \Delta)$, so by Definition 6.1 we get $Comp(\mathcal{D}_3 :: \Gamma \vdash_{\mathrm{s}} (\lambda x.M)N\vec{P} : \neg(\mathsf{S} \times \mathsf{C}) \mid \Delta)$.

$(\mathsf{A} = \cap_{\underline{n}} \mathsf{A}_i)$: By induction. $\square$

We will not need the counterpart of this result for μ-reduction.

We now come to the Replacement Theorem.

Theorem 6.5 (Replacement Theorem) *Let* $\Gamma_0 = x_1{:}\mathsf{R}_1, \ldots, x_m{:}\mathsf{R}_m$, $\Delta_0 = \alpha_1{:}\mathsf{C}_1, \ldots, \alpha_k{:}\mathsf{C}_k$*, and take* $\mathcal{D}^0 :: \Gamma_0 \vdash_{\mathrm{s}} M : \mathsf{S} \mid \Delta_0$. *Assume that for all* $i \in \underline{m}$ *there exist* $\mathcal{D}_i$, N_i *such that* $Comp(\mathcal{D}_i :: \Gamma \vdash_{\mathrm{s}} N_i : \mathsf{R}_i \mid \Delta)$*, and that for all* $l \in \underline{k}$ *there exists* $\vec{\mathcal{D}'_l}$, $\vec{Q_l}$ *such that* $Comp(\vec{\mathcal{D}'_l} :: \Gamma \vdash_{\mathrm{s}} \vec{Q_l} : \mathsf{C}_l \mid \Delta)$*. Then*

$$Comp(\mathcal{D}^0\{\overrightarrow{\mathcal{D}/x{:}\mathsf{R}_{\underline{m}}}\}\{\overrightarrow{\vec{\mathcal{D}'}{\cdot}\gamma/\alpha{:}\mathsf{C}_{\underline{k}}}\} :: \Gamma \vdash_{\mathrm{s}} M\{\overrightarrow{N/x_{\underline{m}}}\}\{\overrightarrow{\vec{Q}{\cdot}\gamma/\alpha_{\underline{k}}}\} : \mathsf{S} \mid \overrightarrow{\gamma{:}\Omega_{\underline{k}}}, \Delta).$$

Proof: By induction on the structure of derivations. For readability, we will write **DS** for $\{\overrightarrow{\mathcal{D}/x{:}\mathsf{R}_{\underline{m}}}\}\{\overrightarrow{\vec{\mathcal{D}'}{\cdot}\gamma/\alpha{:}\mathsf{C}_{\underline{k}}}\}$, and **S** for $\{\overrightarrow{N/x_{\underline{m}}}\}\{\overrightarrow{\vec{Q}{\cdot}\gamma/\alpha_{\underline{k}}}\}$.

(Ax): Then $M = x_j$ for some $j \in \underline{m}$, and $\mathsf{R}_j \leq_{\mathrm{s}} \mathsf{S}$, so $\mathsf{R}_j = \cap_{\underline{n}} \mathsf{A}_i$ and $\mathsf{S} = \mathsf{A}_i$ for some $i \in \underline{n}$. Since we have $\mathcal{D}_j :: \Gamma \vdash_{\mathrm{s}} N_j : \cap_{\underline{n}} \mathsf{A}_i \mid \Delta$, we know $\mathcal{D}_j = \langle \mathcal{D}^1_j; \ldots; \mathcal{D}^n_j; \cap\rangle$, and $\mathcal{D}^i_j :: \Gamma \vdash_{\mathrm{s}} N_j : \mathsf{A}_i \mid \Delta$, and $\mathcal{D}^0\mathbf{DS} = \mathcal{D}^i_j$ by Definition 5.2 and 5.3. Since $Comp(\mathcal{D}_j :: \Gamma \vdash_{\mathrm{s}} N_j : \mathsf{R}_j \mid \Delta)$, by Definition 6.1, $Comp(\mathcal{D}^i_j)$.

(Abs): Then $M = \lambda y.M'$, $\mathsf{S} = \neg(\mathsf{T} \times \mathsf{C})$, and

$$\mathcal{D}^0 = \langle \mathcal{D}^1 :: \Gamma', y{:}\mathsf{T} \vdash_{\mathrm{s}} M' : \neg\mathsf{C} \mid \Delta; Abs\rangle :: \Gamma' \vdash_{\mathrm{s}} \lambda y.M' : \neg(\mathsf{T} \times \mathsf{C}) \mid \Delta.$$

Assume $Comp(\mathcal{D}' :: \Gamma \vdash_{\mathrm{s}} Q : \mathsf{T} \mid \Delta)$, then, by induction,

$$Comp(\mathcal{D}^1\, \mathbf{DS}\, [\mathcal{D}'/y{:}\mathsf{T}] :: \Gamma \vdash_{\mathrm{s}} M\mathbf{S}\{Q/y\} : \neg\mathsf{C} \mid \overrightarrow{\gamma{:}\Omega}, \Delta).$$

Then by Lemma 6.4,

$$Comp(\langle\langle \mathcal{D}^1\, \mathbf{DS}; Abs\rangle, \mathcal{D}'; App\rangle :: \Gamma \vdash_{\mathrm{s}} (\lambda y.M\mathbf{S})Q : \neg\mathsf{C} \mid \overrightarrow{\gamma{:}\Omega}, \Delta),$$

so, by Definition 6.1, 5.2 and 5.3,

$$Comp\,(\langle \mathcal{D}^1 ;Abs\rangle\,\mathbf{DS} :: \Gamma \vdash_{\mathrm{s}} (\lambda y.M)\mathbf{S} : \neg(\mathtt{S}\times\mathtt{C}) \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

(App): Then $M \equiv PQ$, $\mathtt{S} = \neg\mathtt{C}$, and there are $\mathcal{D}^1$, $\mathcal{D}^2$, and $\mathtt{S}$ such that

$$\mathcal{D}^0 \;=\; \langle \mathcal{D}^1 :: \Gamma' \vdash_{\mathrm{s}} P : \neg(\mathtt{S}\times\mathtt{C}) \mid \Delta; \mathcal{D}^2 :: \Gamma' \vdash_{\mathrm{s}} Q : \mathtt{S} \mid \Delta;App\rangle.$$

Then, by induction,

$$Comp\,(\mathcal{D}^1\mathbf{DS} :: \Gamma \vdash_{\mathrm{s}} P\mathbf{S} : \neg(\mathtt{S}\times\mathtt{C}) \mid \overrightarrow{\gamma{:}\Omega},\Delta), \text{ and } Comp\,(\mathcal{D}^2\mathbf{DS} :: \Gamma \vdash_{\mathrm{s}} Q\mathbf{S} : \mathtt{S} \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

Then, by Definition 6.1, 5.2 and 5.3,

$$Comp\,(\langle \mathcal{D}_1, \mathcal{D}_2 ;App\rangle\mathbf{DS} :: \Gamma \vdash_{\mathrm{s}} (PQ)\mathbf{S} : \neg\mathtt{C} \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

(μ_1): Then $M \equiv \mu\beta.[\rho]P$. If $\mathcal{D}^0 :: \Gamma_0 \vdash_{\mathrm{s}} \mu\beta.[\rho]P : \neg\mathtt{C} \mid \Delta_0$, then there exists $\mathcal{D}^1$ and $\mathtt{C}',\mathtt{D},\Delta'_0$ such that $\mathtt{C}' \leq_{\mathrm{s}} \mathtt{D}$, $\Delta_0 = \rho{:}\mathtt{C}',\Delta'_0$, and

$$\mathcal{D}^0 = \langle \mathcal{D}^1 :: \Gamma_0 \vdash_{\mathrm{s}} P : \neg\mathtt{D} \mid \beta{:}\mathtt{C},\rho{:}\mathtt{C}',\Delta'_0 ;\mu_1\rangle :: \Gamma_0 \vdash_{\mathrm{s}} \mu\beta.[\rho]P : \neg\mathtt{C} \mid \rho{:}\mathtt{C}',\Delta'_0.$$

Assume $Comp\,(\overrightarrow{\mathcal{D}_\beta} :: \Gamma \vdash_{\mathrm{s}} \vec{R} : \mathtt{C} \mid \Delta)$ (notice that $\rho = \alpha_l$, for some $l \in \underline{k}$), then by induction

$$Comp\,(\mathcal{D}^1\;\mathbf{DS}\;\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\} :: \Gamma \vdash_{\mathrm{s}} P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\} : \neg\mathtt{D} \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta).$$

Let $(\overrightarrow{\mathcal{D}}_{\underline{k}} :: \Gamma \vdash_{\mathrm{s}} \vec{Q}_{\underline{k}} : \mathtt{C}' \mid \Delta) \cdot \gamma_j/\rho{:}\mathtt{C} \in \mathbf{DS}$, then by Lemma 6.2 also $Comp\,(\overrightarrow{\mathcal{D}}_{\underline{k}} :: \Gamma \vdash_{\mathrm{s}} \vec{Q}_{\underline{k}} : \mathtt{D} \mid \Delta)$. Then by Definition 6.1 (k times)

$$\begin{array}{l} Comp\,(\langle \cdots \langle \mathcal{D}^1\;\mathbf{DS}\;\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\} ; \mathcal{D}_1 ;App\rangle \cdots ; \mathcal{D}_k ;App\rangle :: \\ \qquad\qquad\qquad \Gamma \vdash_{\mathrm{s}} P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\}\vec{Q} : \neg\Omega \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

Then by Definition 6.1, this derivation is strongly normalising:

$$\begin{array}{l} SN(\langle \cdots \langle \mathcal{D}^1\;\mathbf{DS}\;\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\} ; \mathcal{D}_1 ;App\rangle \cdots ; \mathcal{D}_k ;App\rangle :: \\ \qquad\qquad\qquad \Gamma \vdash_{\mathrm{s}} P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\}\vec{Q} : \neg\Omega \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

By Barendregt's convention, we can assume that $\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\}$ does not affect $\overrightarrow{\mathcal{D}} :: \Gamma \vdash_{\mathrm{s}} \vec{Q} : \mathtt{D} \mid \Delta$, so also

$$\begin{array}{l} SN(\langle \cdots \langle \mathcal{D}^1 ; \mathcal{D}_1 ;App\rangle \cdots ; \mathcal{D}_k ;App\rangle\;\mathbf{DS}\;\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\} :: \\ \qquad\qquad\qquad \Gamma \vdash_{\mathrm{s}} P\vec{Q}\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\} : \neg\Omega \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

but then by Lemma 5.8, also

$$\begin{array}{l} SN(\langle\langle \cdots \langle \mathcal{D}^1 ; \mathcal{D}_1 ;App\rangle \cdots ; \mathcal{D}_k ;App\rangle\;\mathbf{DS}\;\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\} ;\mu_1\rangle :: \\ \qquad\qquad\qquad \Gamma \vdash_{\mathrm{s}} \mu\delta.[\gamma_j]P\vec{Q}\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

and again by Lemma 5.8 (k times), also

$$\begin{array}{l} SN(\langle \cdots \langle\langle \cdots \langle\langle \mathcal{D}^1 ; \mathcal{D}_1 ;App\rangle \cdots ; \mathcal{D}_k ;App\rangle ;\mu_1\rangle\;\mathbf{DS} ; \mathcal{D}^1_\beta ;App\rangle \cdots ; \mathcal{D}^{k'}_\beta ;App\rangle :: \\ \qquad\qquad\qquad\qquad\qquad \Gamma \vdash_{\mathrm{s}} (\mu\beta.[\gamma_j]P\vec{Q}\mathbf{S})\vec{R} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

Since $\vec{Q}{\cdot}\gamma_j/\rho{:}\mathtt{C} \in \mathbf{S}$, also

$$SN(\langle \cdots \langle\langle \mathcal{D}^1 ;\mu_1\rangle\;\mathbf{DS} ; \mathcal{D}^1_\beta ;App\rangle \cdots ; \mathcal{D}^{k'}_\beta ;App\rangle :: \Gamma \vdash_{\mathrm{s}} (\mu\beta.[\rho]P\mathbf{S})\vec{R} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

Then by Definition 6.1

$$Comp\,(\langle \cdots \langle\langle \mathcal{D}^1 ;\mu_1\rangle\;\mathbf{DS} ; \mathcal{D}^1_\beta ;App\rangle \cdots ; \mathcal{D}^{k'}_\beta ;App\rangle :: \Gamma \vdash_{\mathrm{s}} (\mu\beta.[\rho]P\mathbf{S})\vec{R} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

and again by Definition 6.1

$$Comp\,(\langle \mathcal{D}^1 ;\mu_1\rangle\mathbf{DS} :: \Gamma \vdash_{\mathrm{s}} (\mu\beta.[\rho]P)\mathbf{S} : \neg\mathtt{C} \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

so

$$Comp\,(\mathcal{D}^0\mathbf{DS} :: \Gamma \vdash_{\mathrm{S}} (\mu\beta.[\rho]P)\mathbf{S} : \neg\mathtt{C} \mid \overrightarrow{\gamma{:}\Omega},\Delta)$$

(μ_2): Then $M \equiv \mu\beta.[\beta]P$. If $\mathcal{D}^0 :: \Gamma_0 \vdash_{\mathrm{S}} \mu\beta.[\beta]P : \neg\mathtt{C} \mid \Delta_0$, then there exists $\mathcal{D}^1$, and $\mathtt{C}'$ such that $\mathtt{C} \leq_{\mathrm{S}} \mathtt{C}'$, and

$$\mathcal{D}^0 = \langle \mathcal{D}^1 :: \Gamma_0 \vdash_{\mathrm{S}} P : \neg\mathtt{C}' \mid \beta{:}\mathtt{C},\Delta_0\,;\mu_2\rangle :: \Gamma_0 \vdash_{\mathrm{S}} \mu\beta.[\beta]P : \neg\mathtt{C} \mid \Delta_0.$$

Assume $Comp\,(\overrightarrow{\mathcal{D}_\beta} :: \Gamma \vdash_{\mathrm{S}} \vec{R} : \mathtt{C} \mid \Delta)$, then by induction

$$Comp\,(\mathcal{D}^1\,\mathbf{DS}\,[\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}] :: \Gamma \vdash_{\mathrm{S}} P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\} : \neg\mathtt{C}' \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta).$$

Let $\mathtt{C} = \neg(\mathtt{T}_1 \times \cdots \times \mathtt{T}_k \times \Omega)$, then $\mathtt{C}' = \neg(\mathtt{T}'_1 \times \cdots \times \mathtt{T}'_{k'} \times \Omega)$ and $\mathtt{T}_i \leq_{\mathrm{S}} \mathtt{T}'_i$ for all $i \in \underline{k}$. By definition of the notation, from $Comp\,(\overrightarrow{\mathcal{D}_\beta} :: \Gamma \vdash_{\mathrm{S}} \vec{R} : \mathtt{C} \mid \Delta)$ we get $Comp\,(\mathcal{D}^i_\beta :: \Gamma \vdash_{\mathrm{S}} R_i : \mathtt{T}_i \mid \Delta)$, and by Lemma 6.2, also $Comp\,(\mathcal{D}^i_\beta :: \Gamma \vdash_{\mathrm{S}} R_i : \mathtt{T}'_i \mid \Delta)$, for all $i \in \underline{k}$. Then by Definition 6.1 (k times), we get

$$Comp\,(\langle \cdots \langle \mathcal{D}^1\,\mathbf{DS}\,[\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}]\,;\mathcal{D}^1_\beta;App\rangle \cdots ;\mathcal{D}^{k'}_\beta;App\rangle :: \Gamma \vdash_{\mathrm{S}} P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\}\vec{R} : \neg\Omega \mid \delta{:}\Omega,\overrightarrow{\gamma{:}\Omega},\Delta)$$

so by Definition 6.1, this derivation is strongly normalisable:

$$SN(\langle \cdots \langle \mathcal{D}^1\,\mathbf{DS}\,\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\}\,;\mathcal{D}^1_\beta;App\rangle \cdots ;\mathcal{D}^{k'}_\beta;App\rangle).$$

Then by Lemma 5.8 also is the following:

$$\begin{array}{l} SN(\langle\langle \cdots \langle \mathcal{D}^1\,\mathbf{DS}\,\{\overrightarrow{\mathcal{D}_\beta}{\cdot}\delta/\beta{:}\mathtt{C}\}\,;\mathcal{D}^1_\beta;App\rangle \cdots ;\mathcal{D}^{k'}_\beta;App\rangle\,;\mu_2\rangle :: \\ \qquad\qquad \Gamma \vdash_{\mathrm{S}} \mu\delta.[\delta](P\mathbf{S}\{\vec{R}{\cdot}\delta/\beta\})\vec{R} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta) \end{array}$$

Then again by Lemma 5.8, we get

$$SN(\langle \cdots \langle\langle \mathcal{D}^1\,\mathbf{DS}\,;\mu_2\rangle\,;\mathcal{D}^1_\beta;App\rangle \cdots ;\mathcal{D}^{k'}_\beta;App\rangle :: \Gamma \vdash_{\mathrm{S}} (\mu\beta.[\beta]P\mathbf{S})\vec{R} : \neg\Omega \mid \overrightarrow{\gamma{:}\Omega},\Delta)$$

and by Definition 6.1, we get $Comp\,(\langle \cdots \langle\langle \mathcal{D}^1\,\mathbf{DS}\,;\mu_2\rangle\,;\mathcal{D}^1_\beta;App\rangle \cdots ;\mathcal{D}^{k'}_\beta;App\rangle)$. Then by Definition 6.1 (k' times)

$$Comp\,(\langle \mathcal{D}^1\,\mathbf{DS}\,;\mu_2\rangle :: \Gamma \vdash_{\mathrm{S}} \mu\beta.[\beta]P\mathbf{S} : \neg\mathtt{C} \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

$(\cap)$: Then $\mathtt{S} = \cap_{\underline{n}}\mathtt{A}_i$, and, for all $i \in \underline{n}$, there exists $\mathcal{D}^i :: \Gamma' \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta$ such that $\mathcal{D}^0 = \langle \mathcal{D}^1;\ldots;\mathcal{D}^n,\cap\rangle$. Then, by induction, for all $i \in \underline{n}$, $Comp\,(\mathcal{D}^i\mathbf{DS} :: \Gamma \vdash_{\mathrm{S}} M\mathbf{S} : \mathtt{A}_i \mid \overrightarrow{\gamma{:}\Omega},\Delta)$, and, by Definition 6.1, $Comp\,(\mathcal{D}^0\mathbf{DS} :: \Gamma \vdash_{\mathrm{S}} M\mathbf{S} : \cap_{\underline{n}}\mathtt{A}_i \mid \overrightarrow{\gamma{:}\Omega},\Delta)$. ☐

Using this last result, we can now prove a strong normalisation result for derivation reduction in '$\vdash_{\mathrm{S}}$'.

Theorem 6.6 *If $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$, then $SN(\mathcal{D})$.*

Proof: Let $\Gamma = x_1{:}\mathtt{T}_1,\ldots,x_n{:}\mathtt{T}_n$, and $\Delta = \alpha_1{:}\mathtt{C}_1,\ldots,\alpha_m{:}\mathtt{C}_m$, Let, for $x_i{:}\mathtt{T}_i \in \Gamma$, $\mathcal{D}_i :: \Gamma \vdash_{\mathrm{S}} x_i : \mathtt{T}_i \mid \Delta$, and for $\alpha_j{:}\mathtt{C}_j \in \Delta$, $\overrightarrow{\mathcal{D}'_j} :: \Gamma \vdash_{\mathrm{S}} \vec{y_j} : \mathtt{C}_j \mid \Delta$. Then by Lemma 6.3:(*ii*), for all $i \in \underline{n}$, $Comp\,(\mathcal{D}_i)$, and for all $j \in \underline{m}$, $Comp\,(\mathcal{D}'_j)$. Then

$$Comp\,(\mathcal{D}\,\{\overrightarrow{\mathcal{D}_i/x_i{:}\mathtt{T}_i}\}\{\overrightarrow{\overrightarrow{\mathcal{D}'_j}{\cdot}\gamma/\alpha{:}\mathtt{C}}\} :: \Gamma \vdash_{\mathrm{S}} M\{\overrightarrow{x/x}\}\{\overrightarrow{\vec{y}{\cdot}\gamma/\alpha}\} : \mathtt{S} \mid \overrightarrow{\gamma{:}\Omega},\Delta).$$

so also $Comp\,(\mathcal{D}\,\{\overrightarrow{\overrightarrow{\mathcal{D}_y}{\cdot}\gamma/\alpha{:}\mathtt{C}}\} :: \Gamma \vdash_{\mathrm{S}} M\{\overrightarrow{\vec{y}{\cdot}\gamma/\alpha}\} : \mathtt{S} \mid \overrightarrow{\gamma{:}\Omega},\Delta)$. Then $SN(\mathcal{D}\,\{\overrightarrow{\overrightarrow{\mathcal{D}_y}{\cdot}\gamma/\alpha{:}\mathtt{C}}\})$ follows by Lemma 6.3:(*i*). By Lemma 5.8:(*viii*), also

$$\begin{array}{l} SN(\langle \ldots \langle\langle \cdots \langle \mathcal{D}\,;\mu_1\rangle \cdots ;\mu_1\rangle;\mathcal{D}'_1;App\rangle \cdots ;\mathcal{D}'_k;App\rangle :: \\ \qquad\qquad \Gamma \vdash_{\mathrm{S}} (\mu\alpha_1.[\beta_1]\cdots\mu\alpha_k.[\beta_k]M)\vec{y_1}\cdots\vec{y_k} : \mathtt{S} \mid \overrightarrow{\gamma{:}\Omega},\Delta). \end{array}$$

Then also $SN(\mathcal{D})$. ☐

We will show below how this result leads to all the characterisation properties; we first prepare the characterisation of approximation by introducing the notion of approximation semantics.

7 Approximation semantics for $\lambda\mu$

Following the approach of Wadsworth [40], we now define an *approximation semantics* for $\lambda\mu$ with respect to $\rightarrow_{\beta\mu}$. We will use this notion in the next section to show an approximation result for '$\vdash_S$'. Approximation for $\Lambda\mu$ has been studied by Saurin [37] and de'Liguoro [31]; *weak* approximants for $\lambda\mu$ are studied by [15].

Essentially, approximants are partially evaluated expressions in which the locations of incomplete evaluation (*i.e.* where reduction *may* still take place) are explicitly marked by the element $\perp$; thus, they *approximate* the result of computations; intuitively, an approximant can be seen as a 'snapshot' of a computation, where we focus on that part of the resulting term which will no longer change.

Definition 7.1 (Approximation for $\lambda\mu$) *i*) We define $\lambda\mu\perp$ as an extension of $\lambda\mu$ by adding the term constant $\perp$.

$$M, N \;::=\; x \mid \lambda y.M \mid MN \mid \mu\alpha.[\beta]M \mid \perp$$

ii) The set of $\lambda\mu$'s *approximants* $\mathcal{A}$ with respect to $\rightarrow_{\beta\mu}$ is defined through the grammar:

$$\begin{array}{rcll} A & ::= & \perp \mid xA_1\cdots A_n & (n \geq 0) \\ & \mid & \lambda x.A & (A \neq \perp) \\ & \mid & \mu\alpha.[\beta]A & (A \neq \mu\gamma[\delta]A',\ A \neq \perp) \end{array}$$

iii) The relation $\sqsubseteq\, \subseteq \lambda\mu\perp^2$ is defined as the smallest pre-order that is the compatible extension of $\perp \sqsubseteq M$.

iv) The set of *approximants* of M, $\mathcal{A}(M)$, is defined as:

$$\mathcal{A}(M) \;\triangleq\; \{\, A \in \mathcal{A} \mid \exists N \in \lambda\mu\ [\, M \rightarrow^*_{\beta\mu} N \;\&\; A \sqsubseteq N \,]\,\}.$$

v) *Approximation equivalence*, $\sim_{\mathcal{A}}$, between terms is defined through:

$$M \sim_{\mathcal{A}} N \;\triangleq\; \mathcal{A}(M) = \mathcal{A}(N).$$

Note that all $\lambda\mu$-terms that are in normal form are approximants and coincide with $\perp$-free approximants.

Approximants are also the normal forms with respect to the notion of reduction on $\lambda\mu\perp$-terms that is the extension of '$\rightarrow_{\beta\mu}$' by adding the reduction rules:

$$\begin{array}{rcl} \lambda x.\perp & \rightarrow & \perp \\ \perp M & \rightarrow & \perp \\ \mu\alpha.[\beta]\perp & \rightarrow & \perp \end{array}$$

but this will play no role in this paper.

The relationship between the approximation relation and reduction is characterised by the following result.

Lemma 7.2 *i) If $A \sqsubseteq M$ and $M \rightarrow^*_{\beta\mu} N$, then $A \sqsubseteq N$.*

ii) $\boldsymbol{H}$ is a head-normal form if and only if there exists $A \in \mathcal{A}$ such that $A \sqsubseteq \boldsymbol{H}$ and $A \neq \perp$.

Proof: *i*) By induction on the structure of approximants.

$(A = \perp)$: Trivial, since $\perp \sqsubseteq N$.

$(A = xA_1 \cdots A_n)$: If $xA_1 \cdots A_n \sqsubseteq M$, then $M \equiv xM_1 \cdots M_n$, with $A_i \sqsubseteq M_i$ for all $i \in \underline{n}$. If $M \rightarrow^*_{\beta\mu} N$, then $N = xN_1 \cdots N_n$ with $M_i \rightarrow^*_{\beta\mu} N_i$, for all $i \in \underline{n}$ (notice that the reduction can take place in many sub-terms, and need not take place in all). Then, by induction, $A_i \sqsubseteq N_i$ for all $i \in \underline{n}$, so $A \sqsubseteq N$.

$(A = \lambda x.A', A' \neq \bot)$: If $\lambda x.A' \sqsubseteq M$, then $M \equiv \lambda x.M'$, with $A' \sqsubseteq M'$. If $M \rightarrow^*_{\beta\mu} N$, then $N = \lambda x.N'$ with $M' \rightarrow^*_{\beta\mu} N'$. Then, by induction, $A' \sqsubseteq N'$, so $A \sqsubseteq N$.

$(A = \mu\alpha.[\beta]A', A' \neq \mu\gamma[\delta]A'', A' \neq \bot)$: If $\mu\alpha.[\beta]A' \sqsubseteq M$, then $M \equiv \mu\alpha.[\beta]M'$, with $A' \sqsubseteq M'$. Since $A' \neq \mu\gamma[\delta]A''$, $M \neq \mu\alpha.[\beta]\mu\gamma[\delta]M''$, so any reduction in M takes place inside M'. So if $M \rightarrow^*_{\beta\mu} N$, then $N = \mu\alpha.[\beta]N'$ with $M' \rightarrow^*_{\beta\mu} N'$. Then, by induction, $A' \sqsubseteq N'$, so $A \sqsubseteq N$.

ii) (*only if*): By induction on the structure of head-normal forms:

$(\boldsymbol{H} = xM_1 \cdots M_n)$: Take $A = x\bot \cdots \bot$.

$(\boldsymbol{H} = \lambda x.\boldsymbol{H}')$: By induction, there exists $A \neq \bot$ such that $A \sqsubseteq \boldsymbol{H}'$. Then $\lambda x.A \sqsubseteq \lambda x.\boldsymbol{H}'$; notice that, since $A \neq \bot$, also $\lambda x.A \in \mathcal{A}$.

$(\boldsymbol{H} = \mu\alpha.[\beta]\boldsymbol{H}', \boldsymbol{H}' \neq \mu\gamma.[\delta]\boldsymbol{H}'')$: By induction, there exists $A \neq \bot$ such that $A \sqsubseteq \boldsymbol{H}'$. Then $\mu\alpha.[\beta]A \sqsubseteq \mu\alpha.[\beta]\boldsymbol{H}'$; notice that, since $A \neq \mu\gamma.[\delta]A'$ and $A \neq \bot$, also $\mu\alpha.[\beta]A \in \mathcal{A}$.

(*if*): If there exists $A \in \mathcal{A}$ such that $A \sqsubseteq M$ and $A \neq \bot$, then either:

$(A = xA_1 \cdots A_n)$: If $xA_1 \cdots A_n \sqsubseteq M$, then $M \equiv xM_1 \cdots M_n$, so M is in head-normal form.

$(A = \lambda x.A', A' \neq \bot)$: If $\lambda x.A' \sqsubseteq M$, then $M \equiv \lambda x.M'$, with $A' \sqsubseteq M'$. Since $A' \neq \bot$, by induction M' is in head-normal form, so also $\lambda x.M'$ is in head-normal form.

$(A = \mu\alpha.[\beta]A', A' \neq \mu\gamma[\delta]A'', A' \neq \bot)$: If $\mu\alpha.[\beta]A' \sqsubseteq M$, then $M \equiv \mu\alpha.[\beta]M'$, with $A' \sqsubseteq M'$. Since $A' \neq \bot$, by induction M' is in head-normal form; since $A' \neq \mu\gamma[\delta]A''$, also $M' \neq \mu\gamma[\delta]M''$, so also $\mu\alpha.[\beta]M'$ is in head-normal form.

The following definition introduces the notion of compatibility between terms through an operation of join on $\lambda\mu\bot$-terms. Terms are compatible when they are syntactically equal, except for positions where $\bot$ occurs.

Definition 7.3 (JOIN, COMPATIBLE TERMS) *i*) On $\lambda\mu\bot$, the partial mapping *join*, $\sqcup : \lambda\mu\bot^2 \rightarrow \lambda\mu\bot$, is defined by:

$$\begin{array}{rcll} \bot \sqcup M & \equiv & M \sqcup \bot & \equiv \; M \\ & & x \sqcup x & \equiv \; x \\ & & (\lambda x.M) \sqcup (\lambda x.N) & \equiv \; \lambda x.(M \sqcup N) \\ & & (\mu\alpha.[\beta]M) \sqcup (\mu\alpha.[\beta]N) & \equiv \; \mu\alpha.[\beta](M \sqcup N) \\ & & (M_1M_2) \sqcup (N_1N_2) & \equiv \; (M_1 \sqcup N_1)(M_2 \sqcup N_2) \end{array}$$

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

It is easy to show that $\sqcup$ is associative and commutative; we will use $\sqcup_{\underline{n}} M_i$ for the term $M_1 \sqcup \cdots \sqcup M_n$. Note that $\bot$ can be defined as the empty join: $\sqcup_{\underline{0}} M_i \triangleq \bot$.

The last alternative in the definition of $\sqcup$ defines the join on applications in a more general way than Scott's [27], that would state that

$$(M_1M_2) \sqcup (N_1N_2) \;\sqsubseteq\; (M_1 \sqcup N_1)(M_2 \sqcup N_2),$$

since it is not always certain if a join of two arbitrary terms exists. Since we will use our more general definition only on terms that are compatible, there is no real conflict.

The following lemma shows that the join acts as least upper bound of compatible terms.

Lemma 7.4 *i) If $P \sqsubseteq M$, and $Q \sqsubseteq M$, then $P \sqcup Q$ is defined, and: $P \sqsubseteq P \sqcup Q$, $Q \sqsubseteq P \sqcup Q$, and $P \sqcup Q \sqsubseteq M$.*

ii) If A_1, $A_2 \in \mathcal{A}(M)$, then A_1 and A_2 are compatible.

Proof: *i*) By easy induction on the definition of '$\sqsubseteq$'.

a) If $P \equiv \bot$, then $P \sqcup Q \equiv Q$, so $P \sqsubseteq P \sqcup Q, Q \sqsubseteq P \sqcup Q$, and $P \sqcup Q \sqsubseteq Q \sqsubseteq M$. (The case $Q \equiv \bot$ goes similarly.)

b) If $P \equiv x$, then $M \equiv x$, and either $Q = \bot$ or also $Q \equiv x$. The first case has been dealt with in part (*i.a*), and for the other: $P \sqcup Q \equiv x$. Obviously, $x \sqsubseteq x \sqcup x$, $x \sqsubseteq x \sqcup x$, and $x \sqcup x \sqsubseteq x$.

c) If $P \equiv \lambda x.N_1$, then $M \equiv \lambda x.N$, $N_1 \sqsubseteq N$, and either $Q = \bot$ or $Q \equiv \lambda x.N_2$. The first case has been dealt with in part (*i.a*), and for the other: then $N_2 \sqsubseteq N$. Then, by induction, $N_1 \sqsubseteq N_1 \sqcup N_2$, $N_2 \sqsubseteq N_1 \sqcup N_2$, and $N_1 \sqcup N_2 \sqsubseteq N$. Then also $\lambda x.N_1 \sqsubseteq \lambda x.N_1 \sqcup N_2$, $\lambda x.N_2 \sqsubseteq \lambda x.N_1 \sqcup N_2$, and $\lambda x.N_1 \sqcup N_2 \sqsubseteq \lambda x.N$. Notice that $\lambda x.N_1 \sqcup N_2 \equiv (\lambda x.N_1) \sqcup (\lambda x.N_2)$.

d) If $P \equiv P_1 Q_1$, then $M \equiv PQ$, $P_1 \sqsubseteq P$, $Q_1 \sqsubseteq Q$, and either $Q = \bot$ or $Q \equiv P_2 Q_2$. The first case has been dealt with in part (*i.a*), and for the other: then $P_2 \sqsubseteq P$, $Q_2 \sqsubseteq Q$. By induction, we know $P_1 \sqsubseteq P_1 \sqcup P_2$, $P_2 \sqsubseteq P_1 \sqcup P_2$, and $P_1 \sqcup P_2 \sqsubseteq P$, as well as $Q_1 \sqsubseteq Q_1 \sqcup Q_2$, $Q_2 \sqsubseteq Q_1 \sqcup Q_2$, and $Q_1 \sqcup Q_2 \sqsubseteq Q$. Then also $P_1 Q_1 \sqsubseteq (P_1 \sqcup P_2)(Q_1 \sqcup Q_2)$, $P_2 Q_2 \sqsubseteq (P_1 \sqcup P_2)(Q_1 \sqcup Q_2)$, and $(P_1 \sqcup P_2)(Q_1 \sqcup Q_2) \sqsubseteq PQ$. Notice that $(P_1 \sqcup P_2)(Q_1 \sqcup Q_2) \equiv (P_1 Q_1) \sqcup (P_2 Q_2)$.

ii) If A_1, $A_2 \in \mathcal{A}(M)$, then there exist N_1, N_2 such that $M \rightarrow^*_{\beta\mu} N_i$ and $A_i \sqsubseteq N_i$, for $i = 1,2$. Since '$\rightarrow_{\beta\mu}$' is confluent, there exists P such that $N_i \rightarrow^*_{\beta\mu} P$; then by Lemma 7.2, also $A_i \sqsubseteq P$, for $i = 1,2$. Then, by part (*i*), A_1 and A_2 are compatible. □

We can also define $\ulcorner M \lrcorner = \sqcup \{ A \mid A \in \mathcal{A}(M) \}$ (which by the previous lemma is well defined); then $\ulcorner \cdot \lrcorner$ corresponds to (a $\lambda\mu$-variant of) Böhm trees [19, 17].

As is standard in other settings, interpreting a $\lambda\mu$-term M through its set of approximants $\mathcal{A}(M)$ gives a semantics.

Theorem 7.5 (Approximation semantics for $\lambda\mu$) *If $M =_{\beta\mu} N$, then $M \sim_{\mathcal{A}} N$.*

Proof: By induction on the definition of '$=_{\beta\mu}$', of which we only show the case $M \rightarrow^*_{\beta\mu} N$.

($\mathcal{A}(M) \subseteq \mathcal{A}(N)$): If $A \in \mathcal{A}(M)$, then there exists L such that $M \rightarrow^*_{\beta\mu} L$ and $A \sqsubseteq L$. Since '$\rightarrow_{\beta\mu}$' is confluent, there exists R such that $L \rightarrow^*_{\beta\mu} R$ and $N \rightarrow^*_{\beta\mu} R$, so also $M \rightarrow^*_{\beta\mu} R$. Then by Lemma 7.2, $A \sqsubseteq R$, and since $N \rightarrow^*_{\beta\mu} R$, we have $A \in \mathcal{A}(N)$.

($\mathcal{A}(N) \subseteq \mathcal{A}(M)$): If $A \in \mathcal{A}(N)$, then there exists L such that $N \rightarrow^*_{\beta\mu} L$ and $A \sqsubseteq L$. But then also $M \rightarrow^*_{\beta\mu} L$, so $A \in \mathcal{A}(M)$.

The reverse implication of this result does not hold, since terms without head-normal form (which have only $\bot$ as approximant) are not all related by reduction; so approximation semantics is not fully abstract.

8 The approximation and head normalisation results for $\vdash_{\mathrm{S}}$

In this section we will show an approximation result that states that every derivation for a term M characterises one of its approximants, *i.e.* for every M, Γ, S, and Δ such that $\Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$, there exists an $A \in \mathcal{A}(M)$ such that $\Gamma \vdash_{\mathrm{S}} A : S \mid \Delta$. From this result, the well-known characterisation of (head-)normalisation of $\lambda\mu$-terms using intersection types follows easily, *i.e.* all terms having a head-normal form are typeable in '$\vdash_{\mathrm{S}}$' and all terms that have a normal form are typeable with a type without ω-occurrences.

First we give some auxiliary definitions and results.

The rules of the system '$\vdash_{\mathrm{S}}$' are generalised to terms containing $\bot$; therefore, if $\bot$ occurs in a term M and there exists Γ, S and Δ such that $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$, in that derivation $\bot$ has to appear in a position where the rule $(\cap)$ is used with $n = 0$, *i.e.* in a sub-term typed with ω. Notice that the terms $\lambda x.\bot$, $\bot M_1 \cdots M_n$, and $\mu\alpha.[\beta]\bot$ are typeable by ω only.

We show that '$\vdash_{\mathrm{S}}$' is closed for $\sqsubseteq$.

Lemma 8.1 $\Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$ *and* $M \sqsubseteq N$ *then* $\Gamma \vdash_{\mathrm{S}} N : S \mid \Delta$.

Proof: By easy induction on the definition of $\sqsubseteq$; the base case, $\bot \sqsubseteq N$, follows from the fact that then $S = \omega$. □

Next we define a notion of type assignment that is similar to that of Definition 3.4, but differs in that it assigns ω only to the term $\bot$. It is defined as '$\vdash_{\mathrm{S}}$' in Definition 3.4, but for rule $(\cap)$ that gets replaced by rule $(\cap_\bot)$.

Definition 8.2 $\bot$-*type assignment* and $\bot$-*derivations* are defined by the following natural deduction system:

$$(Ax): \frac{}{\Gamma, x{:}S \vdash x : A \mid \Delta}\ (S \leq_{\mathrm{S}} A) \qquad (\cap_\bot): \frac{\Gamma \vdash M_i : A_i \mid \Delta \quad (\forall i \in \underline{n})}{\Gamma \vdash \sqcup_{\underline{n}} M_i : \cap_{\underline{n}} A_i \mid \Delta}\ (n = 0 \vee n \geq 2)$$

$$(Abs): \frac{\Gamma, x{:}S \vdash M : \neg C \mid \Delta}{\Gamma \vdash \lambda x.M : \neg(S \times C) \mid \Delta}\ (x \notin \Gamma) \qquad (\mu_1): \frac{\Gamma \vdash M : \neg D \mid \alpha{:}C, \beta{:}C', \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \neg C \mid \beta{:}C', \Delta}\ (\beta \neq \alpha \notin \Delta, C' \leq_{\mathrm{S}} D)$$

$$(App): \frac{\Gamma \vdash M : \neg(S \times C) \mid \Delta \quad \Gamma \vdash N : S \mid \Delta}{\Gamma \vdash MN : \neg C \mid \Delta} \qquad (\mu_2): \frac{\Gamma \vdash M : \neg D \mid \alpha{:}C, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \neg C \mid \Delta}\ (\alpha \notin \Delta, C \leq_{\mathrm{S}} D)$$

We write $\Gamma \vdash_\bot M : S \mid \Delta$ if this statement is derivable using a $\bot$-derivation.

Notice that, by $(\cap_\bot)$, $\Gamma \vdash_\bot \bot : \omega \mid \Delta$, and that this is the only way to assign ω to a term. Moreover, in that rule, the terms M_j need to be compatible (otherwise their join would not be defined).

Lemma 8.3 *i) If* $\mathcal{D} :: \Gamma \vdash_\bot M : S \mid \Delta$, *then* $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$.

ii) If $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : S \mid \Delta$, *then there exists* $M' \sqsubseteq M$ *such that* $\mathcal{D} :: \Gamma \vdash_\bot M' : S \mid \Delta$.

Proof: *i*) By induction on the structure of derivations in $\vdash_\bot$.

(Ax): Immediate.

$(\cap_\bot)$: Then $S = \cap_{\underline{n}} A_i$, $M = \sqcup_{\underline{n}} M_i$, and, for every $i \in \underline{n}$, $\Gamma \vdash_\bot M_i : A_i \mid \Delta$. Then, by induction, for every $i \in \underline{n}$, $\Gamma \vdash_{\mathrm{S}} M_i : A_i \mid \Delta$. Since, by Lemma 7.4, $M_i \sqsubseteq M$ for all $i \in \underline{n}$, by Lemma 8.1, for every $i \in \underline{n}$, $\Gamma \vdash_{\mathrm{S}} M : A_i \mid \Delta$, so by $(\cap)$, $\Gamma \vdash_{\mathrm{S}} M : \cap_{\underline{n}} A_i \mid \Delta$.

(Abs): Then $M \equiv \lambda x.N$, and $S = \neg(T \times C)$, and $\Gamma, x{:}T \vdash_\bot N : \neg C \mid \Delta$. Then, by induction, $\Gamma, x{:}T \vdash_{\mathrm{S}} N : \neg C \mid \Delta$, so by (Abs), $\Gamma \vdash_{\mathrm{S}} \lambda x.N : \neg(T \times C) \mid \Delta$.

(App): Then $M \equiv PQ$, $S = \neg C$, and there exists T such that $\Gamma \vdash_\bot P : \neg(T \times C) \mid \Delta$, and $\Gamma \vdash_\bot Q : T \mid \Delta$. Then, by induction, $\Gamma \vdash_{\mathrm{S}} P : \neg(T \times C) \mid \Delta$, and $\Gamma \vdash_{\mathrm{S}} Q : T \mid \Delta$, so by (App), $\Gamma \vdash_{\mathrm{S}} PQ : S \mid \Delta$.

(μ_1): Then $M \equiv \mu\alpha.[\beta]N$, $S = \neg C$, $\Delta = \beta{:}C', \Delta'$, and $\Gamma \vdash_\bot N : \neg D \mid \alpha{:}C, \beta{:}C', \Delta'$ with $C' \leq_{\mathrm{S}} D$. By induction, $\Gamma \vdash_{\mathrm{S}} N : \neg D \mid \alpha{:}C, \beta{:}C', \Delta'$, so by (μ_1), also $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\beta]N : \neg C \mid \beta{:}C', \Delta'$.

(μ_2): Then $M \equiv \mu\alpha.[\alpha]N$, $S = \neg C$, and $\Gamma \vdash_\bot N : \neg D \mid \alpha{:}C, \Delta$ with $C \leq_{\mathrm{S}} D$. By induction, $\Gamma \vdash_{\mathrm{S}} N : \neg D \mid \alpha{:}C, \Delta$, so by (μ_2), also $\Gamma \vdash_{\mathrm{S}} \mu\alpha.[\alpha]N : \neg C \mid \Delta$.

ii) By induction on the structure of derivations in '$\vdash_{\mathrm{S}}$'.

(Ax): Immediate.

$(\cap)$: Then $\mathtt{S} = \cap_{\underline{n}}\mathtt{A}_i$ and, for every $i \in \underline{n}$, $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta$; by induction, for every $i \in \underline{n}$ there exists $M_i \sqsubseteq M$ such that $\Gamma \vdash_{\perp} M_i : \mathtt{A}_i \mid \Delta$ (notice that then these M_i are compatible). Then, by $(\cap_{\perp})$, we have $\Gamma \vdash_{\perp} \sqcup_{\underline{n}} M_i : \mathtt{A}_i \mid \Delta$. Notice that, by Lemma 7.4, $\sqcup_{\underline{n}} M_i \sqsubseteq M$.

(Abs): Then $M \equiv \lambda x.P$, and $\mathtt{S} = \neg(\mathtt{T} \times \mathtt{C})$, and $\Gamma, x{:}\mathtt{T} \vdash_{\mathrm{S}} P : \neg\mathtt{C} \mid \Delta$. So, by induction, there exists $P' \sqsubseteq P$ such that $\Gamma, x{:}\mathtt{T} \vdash_{\perp} P' : \neg\mathtt{C} \mid \Delta$. Then, by (Abs) we obtain $\Gamma \vdash_{\perp} \lambda x.P' : \neg(\mathtt{T} \times \mathtt{C}) \mid \Delta$. Notice that $\lambda x.P' \sqsubseteq \lambda x.P$.

(App): Then $M \equiv PQ$, $\mathtt{S} = \neg\mathtt{C}$, and there is a $\mathtt{T}$ such that $\Gamma \vdash_{\mathrm{S}} P : \neg(\mathtt{T} \times \mathtt{C}) \mid \Delta$, and $\Gamma \vdash_{\mathrm{S}} Q : \mathtt{T} \mid \Delta$. Then, by induction, there are $P' \sqsubseteq P$, and $Q' \sqsubseteq Q$, such that $\Gamma \vdash_{\perp} P' : \neg(\mathtt{T} \times \mathtt{C}) \mid \Delta$, and $\Gamma \vdash_{\perp} Q' : \mathtt{T} \mid \Delta$. Then, by (App), $\Gamma \vdash_{\perp} P'Q' : \mathtt{S} \mid \Delta$. Notice that $P'Q' \sqsubseteq PQ$.

(μ_1): Then $M \equiv \mu\alpha.[\beta]N$, $\mathtt{S} = \neg\mathtt{C}$, $\Delta = \beta{:}\mathtt{C}',\Delta'$, and $\Gamma \vdash_{\mathrm{S}} N : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$ with $\mathtt{C}' \leq_{\mathrm{S}} \mathtt{D}$. By induction, there exists $N' \sqsubseteq N$, and $\Gamma \vdash_{\perp} N' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{C}',\Delta'$, so by (μ_1), also $\Gamma \vdash_{\perp} \mu\alpha.[\beta]N' : \neg\mathtt{C} \mid \beta{:}\mathtt{C}',\Delta'$. Notice that $\mu\alpha.[\beta]N' \sqsubseteq \mu\alpha.[\beta]N$.

(μ_2): Then $M \equiv \mu\alpha.[\alpha]N$, $\mathtt{S} = \neg\mathtt{C}$, and $\Gamma \vdash_{\mathrm{S}} N : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta$ with $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}$. By induction, there exists $N' \sqsubseteq N$, and $\Gamma \vdash_{\perp} N' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta$, so by (μ_2), also $\Gamma \vdash_{\perp} \mu\alpha.[\alpha]N' : \neg\mathtt{C} \mid \Delta$. Notice that $\mu\alpha.[\alpha]N' \sqsubseteq \mu\alpha.[\alpha]N$. □

Notice that the case $\mathtt{S} = \omega$ is present in the case $(\cap_{\perp})$ of the proof. Then $n = 0$, and $\sqcup_{\underline{n}} M_i = \perp$. Moreover, since M' need not be the same as M, the second derivation in part (*ii*) is not exactly the same; however, it has the same structure in terms of applied derivation rules.

Using Theorem 6.6 and Lemma 8.3, as for the system of [18] (see [36]) and the system of [3], the relation between types assignable to a $\lambda\mu$-term and those assignable to its approximants can be formulated as follows:

Theorem 8.4 (Approximation) $\Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta \iff \exists A \in \mathcal{A}(M)\ [\Gamma \vdash_{\mathrm{S}} A : \mathtt{S} \mid \Delta]$.

Proof: $(\Rightarrow)$: If $\mathcal{D} :: \Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$, then, by Theorem 6.6, $SN(\mathcal{D})$. Let $\mathcal{D}' :: \Gamma \vdash_{\mathrm{S}} N : \mathtt{S} \mid \Delta$ be a normal form of $\mathcal{D}$ with respect to $\rightarrow_{\mathrm{DER}}$, then by Lemma 5.7, $M \rightarrow^{*}_{\beta\mu} N$ and, by Lemma 8.3:(*ii*), there exists $N' \sqsubseteq N$ such that $\mathcal{D}' :: \Gamma \vdash_{\perp} N' : \mathtt{S} \mid \Delta$. So, in particular, N' contains no redexes (no derivation redexes since $\mathcal{D}'$ is in normal form, and none untyped since only $\perp$ can be typed with ω), so $N' \in \mathcal{A}$, and therefore $N' \in \mathcal{A}(M)$.

$(\Leftarrow)$: Let $A \in \mathcal{A}(M)$ be such that $\Gamma \vdash_{\mathrm{S}} A : \mathtt{S} \mid \Delta$. Since $A \in \mathcal{A}(M)$, there exists an M' such that $M \rightarrow^{*}_{\beta\mu} M'$ and $A \sqsubseteq M'$. Then, by Lemma 8.1, $\Gamma \vdash_{\mathrm{S}} M' : \mathtt{S} \mid \Delta$, and, by Theorem 4.5, also $\Gamma \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta$. □

Using this last result, the characterisation of head-normalisation becomes easy to show.

Using this last result, the characterisation of head-normalisation becomes easy to show.

Theorem 8.5 (Head-normalisation) *There exists Γ, $\mathtt{A}$, and Δ such that $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$, if and only if M has a head normal form.*

Proof: (*only if*): If $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$, then, by Theorem 8.4, there exists an $A \in \mathcal{A}(M)$ such that $\Gamma \vdash_{\perp} A : \mathtt{A} \mid \Delta$. Then, by Definition 7.1, there exists N such that $M \rightarrow^{*}_{\beta\mu} N$ and $A \sqsubseteq N$. Since $\mathtt{A} \neq \omega$, $A \not\equiv \perp$, so we know that A is either x, $\lambda x.A'$, $xA_1 \cdots A_n$, or $\mu\alpha.[\beta]A'$ with $A' \neq \mu\gamma.[\delta]A''$. Since $A \sqsubseteq N$, N is either $xM_1 \cdots M_n$ $(n \geq 0)$, $\lambda x.P$, or $\mu\alpha.[\beta]P$ with $P \neq \mu\gamma.[\delta]Q$. Then N is in head-normal from and M has a head-normal form.

(*if*): If M has a head-normal form, then there exists N such that $M \rightarrow^{*}_{\beta\mu} N$ and either:

$(N \equiv xM_1 \cdots M_n)$: Take $\Gamma = x{:}\neg(\omega \times \cdots \times \omega \times \Omega)$ (with n times ω) and $\mathtt{A} = \neg\Omega$.

$(N \equiv \lambda x.P)$: Since P is in head-normal form, by induction there are Γ', $\mathtt{C}$, and Δ' such that $\Gamma' \vdash_{\mathrm{S}} P : \neg\mathtt{C} \mid \Delta'$. If $x{:}\mathtt{S} \in \Gamma'$, take $\Gamma = \Gamma' \backslash x$, and $\mathtt{A} = \neg(\mathtt{S} \times \mathtt{C})$; otherwise take $\Gamma = \Gamma'$

and $A = \neg(\omega \times C)$. In either case, by (Abs), $\Gamma \vdash_S \lambda x.P : A \mid \Delta'$

$(N = \mu\alpha.[\alpha]P)$: Since P is in head-normal form, by induction there are Γ', C, D, and Δ' such that $\Gamma' \vdash_S P : \neg D \mid \alpha{:}C, \Delta'$. Take $C' = C \cap D$[6], then by Lemma 3.9 also $\Gamma' \vdash_S P : \neg D \mid \alpha{:}C', \Delta'$, and since $C' \leq_S D$, by (μ_2) we get $\Gamma' \vdash_S \mu\alpha.[\alpha]P : \neg C' \mid \Delta'$.

$(N = \mu\alpha.[\beta]P$, *with* $\alpha \neq \beta)$: Since P is in head-normal form, by induction there are C, C', D such that $\Gamma' \vdash_S P : \neg D \mid \alpha{:}C, \beta{:}C', \Delta$ and $C' \leq_S D$. Take $C'' = C' \cap D$, then by Lemma 3.9 also $\Gamma' \vdash_S P : \neg D \mid \alpha{:}C, \beta{:}C'', \Delta$, and since $C'' \leq_S D$, we get $\Gamma' \vdash_S \mu\alpha.[\beta]P : \neg C' \mid \beta{:}C'', \Delta'$ from (μ_1).

Notice that in all cases, $\Gamma \vdash_S N : A \mid \Delta$, for some A, and by Theorem 4.5, $\Gamma \vdash_S M : A \mid \Delta$. □

9 Type assignment for (strong) normalisation

In this section we show the characterisation of strong normalisation, for which we first define a notion of derivability obtained from '$\vdash_S$' by restricting the use of the type assignment rule $(\cap)$ to at least two sub-derivations, thereby eliminating the possibility to assign ω to a term. Apart from the elimination of ω in the type language, as with '$\vdash_\perp$', the only change lies in rule $(\cap)$.

Definition 9.1 (SN type assignment) *i*) We define the ω-free types by the grammar:

$$\begin{array}{rcl} A,B & ::= & \neg C \\ R,S,T & ::= & A_1 \cap \cdots \cap A_n \quad (n \geq 1) \\ C,D & ::= & \Omega \mid S \times C \end{array}$$

ii) *SN type assignment* is defined by the following natural deduction system (where all types are ω-free):

$$(Ax): \frac{}{\Gamma, x{:}S \vdash x : A \mid \Delta}\ (S \leq_S A) \qquad (\cap): \frac{\Gamma \vdash M : A_i \mid \Delta \quad (\forall i \in \underline{n})}{\Gamma \vdash M : \cap_{\underline{n}} A_i \mid \Delta}\ (n \geq 2)$$

$$(Abs): \frac{\Gamma, x{:}S \vdash M : \neg C \mid \Delta}{\Gamma \vdash \lambda x.M : \neg(S \times C) \mid \Delta}\ (x \notin \Gamma) \qquad (\mu_1): \frac{\Gamma \vdash M : \neg D \mid \alpha{:}C, \beta{:}C', \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \neg C \mid \beta{:}C', \Delta}$$

$$(App): \frac{\Gamma \vdash M : \neg(S \times C) \mid \Delta \quad \Gamma \vdash N : S \mid \Delta}{\Gamma \vdash MN : \neg C \mid \Delta} \qquad (\mu_2): \frac{\Gamma \vdash M : \neg D \mid \alpha{:}C, \Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \neg C \mid \Delta}\ (\alpha \notin \Delta, C \leq_S D)$$

We write $\Gamma \vdash_{SN} M : S \mid \Delta$ if this judgement is derivable using these rules.

Notice that the only real change in the system compared to '$\vdash_S$' is that ω is no longer an intersection type, so in $(\cap)$ the empty intersection ω is excluded.[7]

In the proofs of Lemma 9.7 and 9.11 we will use the following notation.

Definition 9.2 Given two contexts Γ_1 and Γ_2, we define the context $\Gamma_1 \cap \Gamma_2$ as follows:

$$\begin{array}{rcl} \Gamma_1 \cap \Gamma_2 & \triangleq & \{ x{:}S_1 \cap S_2 \mid x{:}S_1 \in \Gamma_1 \ \&\ x{:}S_2 \in \Gamma_2 \} \cup \\ & & \{ x{:}S \mid x{:}S \in \Gamma_1 \ \&\ x \notin \Gamma_2 \} \cup \\ & & \{ x{:}S \mid x{:}S \in \Gamma_2 \ \&\ x \notin \Gamma_1 \} \end{array}$$

and write $\cap_{\underline{n}} \Gamma_i$ for $\Gamma_1 \cap \cdots \cap \Gamma_n$. The notions $\Delta_1 \cap \Delta_2$ and $\cap_{\underline{n}} \Delta_i$ are defined similarly.

The following properties hold:

[6] This is the first time we need the operation of intersection on continuation types.

[7] In terms of the characterisation of strong normalisation, it would have sufficed to only restrict $(\cap)$; we restrict the set of types as well in order to be able to characterise normalisation as well.

Lemma 9.3 *i) If $S \leq_{S} T$, then $S = \cap_I A_i$, $T = \cap_J B_j$, and for every $j \in J$ there exists $i \in I$ such that $A_i = B_j$.*

ii) $\Gamma \vdash_{SN} M : S \cap T \mid \Delta$, if and only if $\Gamma \vdash_{SN} M : S \mid \Delta$ and $\Gamma \vdash_{SN} M : T \mid \Delta$.

iii) $\Gamma, x{:}S \vdash_{SN} x : T \mid \Delta$, if and only if $S \leq_{S} T$.

iv) $\Gamma \vdash_{SN} M : S \mid \alpha{:}C, \beta{:}C, \Delta$ if and only if $\Gamma \vdash_{SN} M\{\alpha/\beta\} : S \mid \alpha{:}C, \Delta$.

v) If $\Gamma \vdash_{SN} M : S \mid \Delta$, then $\{x{:}T \in \Gamma \mid x \in fv(M)\} \vdash_{SN} M : S \mid \{\alpha{:}C \in \Delta \mid \alpha \in fn(M)\}$.

vi) If $\Gamma \vdash_{SN} M : S \mid \Delta$ & $\Gamma' \subseteq \Gamma$ & $\Delta' \subseteq \Delta$, then $\Gamma' \vdash_{SN} M : S \mid \Delta'$.

vii) If $\mathcal{D} :: \Gamma \vdash_{SN} M : S \mid \Delta$, then $\mathcal{D} :: \Gamma \vdash_{S} M : S \mid \Delta$.

Proof: Straightforward. □

As for '$\vdash_S$', we can show that $(\leq_S)$ is an admissible rule in '$\vdash_{SN}$'.

Lemma 9.4 *If $\Gamma \vdash_{SN} M : S \mid \Delta$, and Γ', T, and Δ' are all ω-free and satisfy $\Gamma' \leq_S \Gamma$, $\Delta' \leq_S \Delta$, and $S \leq_S T$, then $\Gamma' \vdash_{SN} M : T \mid \Delta'$.*

Proof: Much the same as the proof for Theorem 3.9. □

The following lemma shows a (limited) subject expansion result for '$\vdash_{SN}$': it states that if a contraction of a redex is typeable, then so is the redex, provided that the operand N is typeable in its own right; since N might not appear in the contractum, and ω is missing, we need to assume that separately. Notice that we demand that N is typeable in the *same contexts* as the contractum itself; this property would not hold once we consider contextual closure (in particular, when the reduction takes place under an abstraction); it might be that free names or variables in N get bound.

Lemma 9.5 (Term substitution lemma for '$\vdash_{SN}$') *If $\Gamma \vdash_{SN} M\{N/x\} : T \mid \Delta$ and $\Gamma \vdash_{SN} N : B \mid \Delta$, then there exists S such that $\Gamma, x{:}S \vdash_{SN} M : T \mid \Delta$ and $\Gamma \vdash_{SN} N : S \mid \Delta$.*

Proof: As the proof for Lemma 4.2, with the exception of:

$(M \equiv y \neq x)$: We have $y\{N/x\} \equiv y$ and $x \notin fv(y)$; by Lemma 9.3 we have $\Gamma, x{:}B \vdash_{SN} y : \neg C$.

Lemma 9.6 (Structural substitution lemma for '$\vdash_{SN}$') *If $\Gamma \vdash_{SN} M\{N{\cdot}\gamma/\alpha\} : T \mid \gamma{:}C, \Delta$ and $\Gamma \vdash_{SN} N : B \mid \Delta$, then there exists S such that $\Gamma \vdash_{SN} M : T \mid \alpha{:}S \times C, \Delta$ and $\Gamma \vdash_{SN} N : S \mid \Delta$.*

Proof: As the proof for Lemma 4.1, with the exception of:

$(M \equiv x)$: Then $x\{N{\cdot}\gamma/\alpha\} = x$. Take $S = B$, then by Lemma 9.3, also $\Gamma \vdash_{SN} x : \neg C' \mid \alpha{:}S \times C, \Delta$.

To prepare the characterisation of terms by their assignable types, we first prove that an approximant is typeable in $\vdash_{SN}$, if and only if it does not contain $\perp$. This forms the basis for the result that all normalisable terms are typeable without ω. Notice that the first result is stated for '$\vdash_S$'.

Lemma 9.7 *i) If $\Gamma \vdash_S A : A \mid \Delta$, and Γ, A, and Δ are ω-free, then A is $\perp$-free.*

ii) If A is $\perp$-free, then there are Γ, A, and Δ, such that $\Gamma \vdash_{SN} A : A \mid \Delta$.

Proof: By induction on the structure of approximate normal forms.

i) $(A \equiv x)$: Immediate.

$(A \equiv \perp)$: Impossible, by inspecting the rules of '$\vdash_S$'.

$(A \equiv \lambda x.A')$: By (Abs), $A = \neg(T \times C)$, and $\Gamma, x{:}T \vdash_S A' : \neg C \mid \Delta$. Of course also $\Gamma, x{:}T$, and $\neg C$ are ω-free, so by induction, A' is $\perp$-free, so also $\lambda x.A'$ is $\perp$-free.

$(A \equiv x A_1 \cdots A_n)$: Then by (App) and (Ax), $\Gamma \vdash_S A_i : S_i \mid \Delta$, and $x{:}\cap_{\underline{m}} B_i \in \Gamma$, and, for some $j \in m$, $B_j = \neg(S_1 \times S_2 \times \cdots \times S_n \times C)$ and $A = \neg C$. Since each S_i occurs in B_j, which occurs in Γ, all are ω-free, so by induction each A_i is $\bot$-free. Then also $xA_1\cdots A_n$ is $\bot$-free.

$(A \equiv \mu\alpha.[\beta]A'$, *with* $\alpha \neq \beta$ *and* $A' \neq \mu\gamma.[\delta]A'')$: Then $\mathtt{A} = \neg\mathtt{C}$, and by (μ_1) there exists $\mathtt{D}$, $\mathtt{D}'$ such that $\Delta = \beta{:}\mathtt{D}',\Delta'$, $\mathtt{D}' \leq_{\mathrm{S}} \mathtt{D}$, and $\Gamma \vdash_{\mathrm{S}} A' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{D}',\Delta'$. Since $\mathtt{D}' \leq_{\mathrm{S}} \mathtt{D}$, and $\mathtt{D}'$ is ω-free, so is $\mathtt{D}$; then, by induction, A' is $\bot$-free, so so is $\mu\alpha.[\beta]A'$.

$(A \equiv \mu\alpha.[\alpha]A'$, *with* $A' \neq \mu\gamma.[\delta]A'')$: Then $\mathtt{A} = \neg\mathtt{C}$, and by (μ_2) there exists $\mathtt{D}$ such that $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}$, and $\Gamma \vdash_{\mathrm{S}} A' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta$. Since $\mathtt{C} \leq_{\mathrm{S}} \mathtt{D}$, and $\mathtt{C}$ is ω-free, so is $\mathtt{D}$; then, by induction, A' is $\bot$-free, so so is $\mu\alpha.[\alpha]A'$.

ii) $(A \equiv x)$: Take $x{:}\neg\Omega \vdash_{\mathrm{SN}} x : \neg\Omega \mid$.

$(A \equiv \lambda x.A')$: By induction there exists Γ, Δ, and $\mathtt{C}$ such that $\Gamma \vdash_{\mathrm{SN}} A' : \neg\mathtt{C} \mid \Delta$. If x does not occur in Γ, take an ω-free $\mathtt{T}$; otherwise, there exist $x{:}\mathtt{T} \in \Gamma$ and $\mathtt{T}$ is ω-free. In either case, by (Abs) we obtain $\Gamma\backslash x \vdash_{\mathrm{SN}} \lambda x.A' : \neg(\mathtt{T} \times \mathtt{C}) \mid \Delta$.

$(A \equiv xA_1\cdots A_n)$: By induction, for every $i \in \underline{n}$ there are $\mathtt{A}_i$, Γ_i, and Δ_i such that $\Gamma_i \vdash_{\mathrm{SN}} A_i : \mathtt{A}_i \mid \Delta_i$; take $\Gamma = \cap_n \Gamma_i$ and $\Delta = \cap_n \Delta_i$, then by weakening also $\Gamma \vdash_{\mathrm{SN}} A_i : \mathtt{A}_i \mid \Delta$, for every $i \in \underline{n}$. Then $\Gamma \cap \{\, x{:}\neg(\mathtt{A}_1 \times \cdots \times \mathtt{A}_n \times \Omega) \,\} \vdash_{\mathrm{SN}} xA_1\cdots A_n : \neg\Omega \mid \Delta$.

$(A \equiv \mu\alpha.[\beta]A'$, *with* $\alpha \neq \beta$ *and* $A' \neq \mu\gamma.[\delta]A'')$: By induction there exists Γ, Δ, $\mathtt{C}$, $\mathtt{C}'$, and $\mathtt{D}$ such that $\Gamma \vdash_{\mathrm{SN}} A' : \neg\mathtt{C} \mid \alpha{:}\mathtt{C}',\beta{:}\mathtt{D},\Delta$. Then by weakening $\Gamma \vdash_{\mathrm{SN}} A' : \neg\mathtt{C} \mid \alpha{:}\mathtt{C}',\beta{:}\mathtt{D}\cap\mathtt{C},\Delta$, and by (μ_1) we have $\Gamma \vdash_{\mathrm{SN}} \mu\alpha.[\beta]A' : \neg\mathtt{C}' \mid \beta{:}\mathtt{D}\cap\mathtt{C},\Delta$.

$(A \equiv \mu\alpha.[\alpha]A'$, *with* $A' \neq \mu\gamma.[\delta]A'')$: By induction $\Gamma \vdash_{\mathrm{SN}} A' : \neg\mathtt{C} \mid \alpha{:}\mathtt{D},\Delta$ for some Γ, Δ, $\mathtt{C}$, and $\mathtt{D}$. Then by weakening $\Gamma \vdash_{\mathrm{SN}} A' : \neg\mathtt{C} \mid \beta{:}\mathtt{D}\cap\mathtt{C},\Delta$, so by (μ_2), $\Gamma \vdash_{\mathrm{SN}} \mu\alpha.[\alpha]A' : \mathtt{D}\cap\neg\mathtt{C} \mid \Delta$. $\square$

We are now in the position to characterise normalisable terms.

Theorem 9.8 (Characterisation of Normalisation) *There exists ω-free Γ, Δ, and $\mathtt{A}$ such that $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$, if and only if M has a normal form.*

Proof: $(\Rightarrow)$: If $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$, by Theorem 8.4, there exists $A \in \mathcal{A}(M)$ such that $\Gamma \vdash_{\mathrm{S}} A : \mathtt{A} \mid \Delta$. Since $\Gamma,\mathtt{A}$ are ω-free, by Lemma 9.7:(*i*), this A is $\bot$-free. By Definition 7.1 there exists $M' =_{\beta\mu} M$ such that $A \sqsubseteq M'$. Since A contains no $\bot$, $A \equiv M'$, so M' is a normal form, so, especially, M has a normal form.

$(\Leftarrow)$: If M' is the normal form of M, then it is a $\bot$-free approximate normal form. Then by Lemma 9.7:(*ii*) there are Γ, $\mathtt{A}$, and Δ such that $\Gamma \vdash_{\mathrm{SN}} M' : \mathtt{S} \mid \Delta$. Then, by Theorem 4.5, $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$, and Γ, $\mathtt{A}$, and Δ are ω-free. $\square$

(Notice that, in the second part, in general, the property that ω is not used at all, is lost.)

The following lemma shows that type assignment is preserved in the ω-free system for the expansion of redexes (notice that the result is not stated for arbitrary reduction steps, but only for terms that are proper redexes).

Lemma 9.9 i) If $\Gamma \vdash_{\mathrm{SN}} M\{N/x\} : \mathtt{A} \mid \Delta$ and $\Gamma \vdash_{\mathrm{SN}} N : \mathtt{B} \mid \Delta$, then $\Gamma \vdash_{\mathrm{SN}} (\lambda x.M)N : \mathtt{A} \mid \Delta$.

ii) If $\Gamma \vdash_{\mathrm{SN}} \mu\gamma.[\gamma]M\{N{\cdot}\gamma/\alpha\}N : \mathtt{A} \mid \Delta$ and $\Gamma \vdash_{\mathrm{SN}} N : \mathtt{B} \mid \Delta$, then $\Gamma \vdash_{\mathrm{SN}} (\mu\alpha.[\alpha]M)N : \mathtt{A} \mid \Delta$.

iii) If $\Gamma \vdash_{\mathrm{SN}} \mu\gamma.[\beta]M\{N{\cdot}\gamma/\alpha\} : \mathtt{A} \mid \Delta$ (with $\gamma \neq \beta$) and $\Gamma \vdash_{\mathrm{SN}} N : \mathtt{B} \mid \Delta$, then $\Gamma \vdash_{\mathrm{SN}} (\mu\alpha.[\beta]M)N : \mathtt{A} \mid \Delta$.

iv) If $\Gamma \vdash_{\mathrm{SN}} \mu\alpha.([\delta]M)\{\beta/\gamma\} : \mathtt{A} \mid \Delta$, then $\Gamma \vdash_{\mathrm{SN}} \mu\alpha.[\beta]\mu\gamma.[\delta]M : \mathtt{A} \mid \Delta$.

Proof: *i*) Like that for Theorem 4.4, but using Lemma 9.5.

ii) - *iii*) Like that for Lemma 4.1 and Theorem 4.4, but for the fact that the additional assumption $\Gamma \vdash_{\mathrm{SN}} N : \mathtt{B} \mid \Delta$ is used when $\alpha \notin fn(M)$.

iv) As in the proof of Theorem 4.4, but using Lemma 9.6. $\square$

Theorem 9.12 below shows that the set of strongly normalisable terms is exactly the set of terms typeable in '$\vdash_{\text{SN}}$'. The proof goes by induction on the leftmost outermost reduction path. First we will introduce the notion of leftmost, outermost reduction.

Definition 9.10 An occurrence of a redex $\mathtt{R}$ in a term M is called the *leftmost outermost redex of M* ($lor(M)$), if:

i) There is no redex $\mathtt{R}'$ in M such that $\mathtt{R}' = \mathtt{C}[\mathtt{R}]$ with $\mathtt{C}[-] \neq [-]$ (*outermost*) ;

ii) There is no redex $\mathtt{R}'$ in M such that $M = \mathtt{C}_0[\mathtt{C}_1[\mathtt{R}']\mathtt{C}_2[\mathtt{R}]]$ (*leftmost*).

We write $M \rightarrow_{lor} N$ is used to indicate that M reduces to N by contracting $lor(M)$.

The following lemma formulates a subject expansion result for '$\vdash_{\text{SN}}$' with respect to leftmost outermost reduction.

Lemma 9.11 Assume $M \rightarrow_{lor} N$, and $\Gamma \vdash_{\text{SN}} N : \neg\mathtt{C} \mid \Delta$; if $lor(M) = PQ$ also assume that $\Gamma_0 \vdash_{\text{SN}} Q : \mathtt{B} \mid \Delta_0$. Then there exists $\Gamma',\Delta',\mathtt{C}'$ such that $\Gamma' \vdash_{\text{SN}} M : \neg\mathtt{C}' \mid \Delta'$.

Proof: We reason by induction on the structure of terms:

($M \equiv VP_1 \cdots P_n$): We distinguish two cases:

a) V is a $\rightarrow_{\beta\mu}$-redex, and $N \equiv V'P_1 \cdots P_n$, where V' is the result of contracting V. From the fact that $\Gamma \vdash_{\text{SN}} V'P_1 \cdots P_n : \neg\mathtt{C} \mid \Delta$, we know there are $\mathtt{S}_1,\ldots,\mathtt{S}_n$ such that $\Gamma \vdash_{\text{SN}} V' : \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \mid \Delta$, and $\Gamma \vdash_{\text{SN}} P_i : \mathtt{S}_i \mid \Delta$ for all $i \in \underline{n}$. By weakening we have both $\Gamma \cap \Gamma_0 \vdash_{\text{SN}} V' : \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \mid \Delta \cap \Delta_0$ and $\Gamma \cap \Gamma_0 \vdash_{\text{SN}} Q : \mathtt{B} \mid \Delta \cap \Delta_0$, and also $\Gamma \cap \Gamma_0 \vdash_{\text{SN}} P_i : \mathtt{S}_i \mid \Delta \cap \Delta_0$ for all $i \in \underline{n}$. Then by Lemma 9.9, $\Gamma \cap \Gamma_0 \vdash_{\text{SN}} V : \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \mid \Delta \cap \Delta_0$, so also $\Gamma \cap \Gamma_0 \vdash_{\text{SN}} VP_1 \cdots P_n : \neg\mathtt{C} \mid \Delta \cap \Delta_0$.

b) $V \equiv y$, so there exists $j \in \underline{n}$ such that $lor(M) = lor(P_j)$, $P_j \rightarrow_{lor} P'_j$, and $N \equiv yP_1 \cdots P' \cdots P_n$. From $\Gamma \vdash_{\text{SN}} yP_1 \cdots P'_j \cdots P_n : \neg\mathtt{C} \mid \Delta$, we know there are $\mathtt{S}_1,\ldots,\mathtt{S}_n$ such that $\Gamma \vdash_{\text{SN}} P_i : \mathtt{S}_i \mid \Delta$ for all $i \neq j \in \underline{n}$ and $\Gamma \vdash_{\text{SN}} P'_j : \mathtt{S}_j \mid \Delta$, and $\Gamma \vdash_{\text{SN}} y : \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \mid \Delta$. Notice that then there exists $y{:}\mathtt{T} \in \Gamma$ such that $\mathtt{T} \leq_{\text{S}} \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C})$. Then, by induction, there are Γ_j, Δ_j, and $\mathtt{B}$ such that $\Gamma_j \vdash_{\text{SN}} P_j : \mathtt{B} \mid \Delta_j$. Then

$$\Gamma \cap \Gamma_j \cap \{ y{:}\neg(\mathtt{S}_1 \times \cdots \mathtt{B} \cdots \times \mathtt{S}_n \times \mathtt{C}) \} \vdash_{\text{SN}} yP_1 \cdots P_j \cdots P_n : \neg\mathtt{C} \mid \Delta_j \cap \Delta.$$

($M \equiv \lambda y.M'$): If $M \rightarrow_{lor} N$, then $N = \lambda y.N'$ and $M' \rightarrow_{lor} N'$. Then there exists $\mathtt{S},\mathtt{D}$ such that $\Gamma,y{:}\mathtt{S} \vdash_{\text{SN}} N' : \neg\mathtt{D} \mid \Delta$ and $\mathtt{C} = \mathtt{S} \times \mathtt{D}$. By induction, there exists Γ', Δ', $\mathtt{S}'$, and $\mathtt{D}'$ such that $\Gamma',y{:}\mathtt{S}' \vdash_{\text{SN}} M' : \neg\mathtt{D}' \mid \Delta'$. Then, by (Abs), $\Gamma' \vdash_{\text{SN}} \lambda y.M' : \neg(\mathtt{S}' \times \mathtt{D}') \mid \Delta'$.

($M \equiv \mu\alpha.[\beta]M'$ *with* $\alpha \neq \beta$): Then $N = \mu\alpha.[\beta]N'$ and $M' \rightarrow_{lor} N'$. Since $\Gamma \vdash_{\text{SN}} \mu\alpha.[\beta]N' : \neg\mathtt{C} \mid \Delta$, there are Δ_1, $\mathtt{E}$, $\mathtt{D}$ such that $\Delta = \beta{:}\mathtt{E},\Delta_1$, $\mathtt{E} \leq_{\text{S}} \mathtt{D}$, and $\Gamma \vdash_{\text{SN}} N' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\beta{:}\mathtt{E},\Delta_1$. Then by induction there exist Γ', $\mathtt{C}'$, $\mathtt{E}'$, $\mathtt{D}'$, and Δ' such that $\Gamma' \vdash_{\text{SN}} N' : \neg\mathtt{D}' \mid \alpha{:}\mathtt{C}',\beta{:}\mathtt{E}',\Delta'$. By Lemma 9.4 we have $\Gamma' \vdash_{\text{SN}} N' : \neg\mathtt{D}' \mid \alpha{:}\mathtt{C}',\beta{:}\mathtt{E}' \cap \mathtt{D}',\Delta'$ and $\Gamma' \vdash_{\text{SN}} \mu\alpha.[\beta]N' : \neg\mathtt{C}' \mid \beta{:}\mathtt{E}' \cap \mathtt{D}',\Delta'$ follows by (μ_1).

($M \equiv \mu\alpha.[\alpha]M'$): Then $N = \mu\alpha.[\alpha]N'$ and $M' \rightarrow_{lor} N'$. Since $\Gamma \vdash_{\text{SN}} \mu\alpha.[\alpha]N' : \neg\mathtt{C} \mid \Delta$, there exists $\mathtt{D}$ such that $\mathtt{C} \leq_{\text{S}} \mathtt{D}$, and $\Gamma \vdash_{\text{SN}} N' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C},\Delta_0$. Then by induction there exist Γ', $\mathtt{C}'$, $\mathtt{D}'$, and Δ' such that $\Gamma' \vdash_{\text{SN}} N' : \neg\mathtt{D}' \mid \alpha{:}\mathtt{C}',\Delta'$. By Lemma 9.4 $\Gamma' \vdash_{\text{SN}} N' : \neg\mathtt{D}' \mid \alpha{:}\mathtt{C}' \cap \mathtt{D}',\Delta'$ and then by (μ_2) we get $\Gamma' \vdash_{\text{SN}} \mu\alpha.[\alpha]N' : \neg(\mathtt{C}' \cap \mathtt{D}') \mid \Delta'$. $\square$

We can now show that all strongly normalisable terms are exactly those typeable in '$\vdash_{\text{SN}}$'.

Theorem 9.12 (Characterisation of strong normalisation) *There exists Γ, Δ, and $\mathtt{A}$ such that $\Gamma \vdash_{\text{SN}} M : \mathtt{A} \mid \Delta$ if and only if M is strongly normalisable with respect to $\rightarrow_{\beta\mu}$.*

Proof: ($\Rightarrow$): If $\mathcal{D} :: \Gamma \vdash_{\text{SN}} M : \mathtt{A} \mid \Delta$, then by Lemma 9.3:(*vii*) also $\mathcal{D} :: \Gamma \vdash_{\text{S}} M : \mathtt{A} \mid \Delta$. Then, by Theorem 6.6, $\mathcal{D}$ is strongly normalisable with respect to $\rightarrow_{\text{Der}}$. Since $\mathcal{D}$ contains no ω, all redexes in M correspond to redexes in $\mathcal{D}$, a property that is preserved by derivation reduction (it does not introduce ω). So also M is strongly normalisable with respect to $\rightarrow_{\beta\mu}$.

$(\Leftarrow)$: By induction on the maximum of the lengths of reduction sequences for a strongly normalisable term M to its normal form (denoted by $\#M$).

a) If $\#M = 0$, then M is in normal form, and by Lemma 9.7:(*ii*), there exist Γ, Δ and A such that $\Gamma \vdash_{\text{SN}} M : A \mid \Delta$.

b) If $\#M \geq 1$, so M contains a redex, then let $M \rightarrow_{lor} N$ by contracting the redex PQ. Then $\#N < \#M$, and $\#Q < \#M$ (since Q is a proper sub-term of a redex in M), so by induction, for some Γ, Γ', Δ, Δ', A, and B, we have $\Gamma \vdash_{\text{SN}} N : A \mid \Delta$ and $\Gamma' \vdash_{\text{SN}} Q : B \mid \Delta'$. Then, by Lemma 9.11, there exist Γ_1, Δ_1, C such that $\Gamma_1 \vdash_{\text{SN}} M : C \mid \Delta_1$. If the redex is $\mu\alpha.[\beta]\mu\gamma.[\delta]M$, then $\#\mu\alpha.[\beta]\mu\gamma.[\delta]M > \#\mu\alpha.([\delta]M)\{\beta/\gamma\}$, so the result follows by induction. $\square$

10 The relation between $\vdash_{\text{S}}$ and $\vdash_{\wedge}$

We will now establish a direct relation between the notion of strict negated intersection type assignment '$\vdash_{\text{S}}$' and that of intersection type assignment '$\vdash_{\wedge}$' as defined in [12].

To be able to express the relation between '$\vdash_{\wedge}$' and '$\vdash_{\text{S}}$', we need to reason through approximants. In Theorem 8.4 we have shown the approximation result for '$\vdash_{\text{S}}$'; a similar result has been shown for '$\vdash_{\wedge}$' by de'Liguoro [31].

Theorem 10.1 ([31]) $\Gamma \vdash_{\wedge} M : \delta \mid \Delta \iff \exists A \in \mathcal{A}(M)\ [\Gamma \vdash_{\wedge} M : \delta \mid \Delta]$.

Using these two results, we will now establish the relation between '$\vdash_{\wedge}$' and '$\vdash_{\text{S}}$'. We first state some properties of '$\vdash_{\wedge}$', as shown in [12].

Lemma 10.2 ([12]) *If $\Gamma' \leq_{\wedge} \Gamma$, $\Delta' \leq_{\wedge} \Delta$, $\sigma \leq_{\wedge} \tau$, and $\Gamma \vdash_{\wedge} T : \sigma \mid \Delta$, then $\Gamma' \vdash_{\wedge} T : \tau \mid \Delta'$.*

Lemma 10.3 (Generation Lemma for '$\vdash_{\wedge}$' [12]) *Let $\delta \not\sim_{\wedge} \omega$ and $\kappa \not\sim_{\wedge} \omega$:*

$$\begin{array}{rcl}
\Gamma \vdash_{\wedge} x : \delta \mid \Delta & \iff & \exists x{:}\delta \in \Gamma\ [\delta' \leq_{\wedge} \delta] \\
\Gamma \vdash_{\wedge} \lambda x.M : \delta \mid \Delta & \iff & \exists I \forall i \in I\ \exists \delta_i,\kappa_i,\rho_i\ [\Gamma, x{:}\delta_i \vdash_{\wedge} M : \kappa_i \rightarrow \rho_i \mid \Delta\ \&\ \wedge_I \delta_i \times \kappa_i \rightarrow \rho_i \leq_{\wedge} \delta] \\
\Gamma \vdash_{\wedge} MN : \delta \mid \Delta & \iff & \\
 & & \exists I \forall i \in I\ \exists \delta_i,\kappa_i,\rho_i\ [\Gamma \vdash_{\wedge} M : \delta_i \times \kappa_i \rightarrow \rho_i \mid \Delta\ \&\ \Gamma \vdash_{\wedge} N : \delta_i \mid \Delta\ \&\ \wedge_I \kappa_i \rightarrow \rho_i \leq_{\wedge} \delta] \\
\Gamma \vdash_{\wedge} [\alpha]M : \kappa \mid \Delta & \iff & \exists \kappa', I \forall i \in I\ \exists \delta_i\ [\Gamma \vdash_{\wedge} M : \delta_i \mid \Delta\ \&\ \alpha{:}\kappa' \in \Delta\ \&\ \wedge_I \delta_i \times \kappa' \leq_{\wedge} \kappa] \\
\Gamma \vdash_{\wedge} \mu\alpha.\mathsf{C} : \delta \mid \Delta & \iff & \exists I \forall i \in I\ \exists \kappa_i,\rho_i,\kappa_i'\ [\Gamma \vdash_{\wedge} \mathsf{C} : (\kappa_i \rightarrow \rho_i) \times \kappa_i \mid \alpha{:}\kappa_i' \in \Delta\ \&\ \wedge_I \kappa_i' \rightarrow \rho_i \leq_{\wedge} \delta]
\end{array}$$

We can show that when restricting '$\vdash_{\wedge}$' to $\lambda\mu$, the equivalent of rules (μ_2) and (μ_1) of '$\vdash_{\text{S}}$' are inherently present.

Lemma 10.4 *i) If $\Gamma \vdash_{\wedge} \mu\alpha.[\alpha]P : \kappa \rightarrow \rho \mid \Delta$, then there exists κ' such that $\kappa \leq_{\wedge} \kappa'$ and $\Gamma \vdash_{\wedge} P : \kappa' \rightarrow \rho \mid \alpha{:}\kappa, \Delta$.*

ii) If $\Gamma \vdash_{\wedge} \mu\alpha.[\beta]P : \kappa \rightarrow \rho \mid \beta{:}\kappa', \Delta$, then there exists κ'' such that $\kappa' \leq_{\wedge} \kappa''$ and $\Gamma \vdash_{\wedge} P : \kappa'' \rightarrow \rho \mid \alpha{:}\kappa, \beta{:}\kappa', \Delta$.

Proof: *i*) By Lemma 10.3, there exists I such that for all $i \in I$ there are κ_i, κ_i', ρ_i, and M_i such that for every $m \in M_i$ there is a δ_i^m, such that:

$$\begin{array}{l}
\forall i \in I\,[\forall m \in M_i\ [\Gamma \vdash_{\wedge} A' : \delta_i^m \mid \alpha{:}\kappa_i', \Delta]\ \& \\
\qquad \wedge_{M_i} \delta_i^m \times \kappa_i' \leq_{\wedge} (\kappa_i \rightarrow \rho_i) \times \kappa_i]\ \&\ \wedge_I \kappa_i' \rightarrow \rho_i \leq_{\wedge} \kappa \rightarrow \rho.
\end{array}$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash_\wedge A' : \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m) \mid \alpha{:}\kappa_i,\beta{:}\kappa',\Delta}{\Gamma \vdash_\wedge [\beta]A' : \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)\times\kappa' \mid \alpha{:}\kappa_i,\beta{:}\kappa',\Delta}\,(Cmd) \quad (\forall m \in M_i)}{\Gamma \vdash_\wedge [\beta]A' : \wedge_{M_i}\wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)\times\kappa' \mid \alpha{:}\kappa_i,\beta{:}\kappa',\Delta}\,(\wedge)}{\Gamma \vdash_\wedge [\beta]A' : (\kappa'_i\rightarrow\rho_i)\times\kappa'_i \mid \alpha{:}\kappa_i,\beta{:}\kappa',\Delta}\,(\leq_\wedge)\ \ (\dagger)}{\Gamma \vdash_\wedge \mu\alpha.[\beta]A' : \kappa_i\rightarrow\rho_i \mid \beta{:}\kappa',\Delta}\,(\mu) \quad (\forall i \in I)}{\dfrac{\Gamma \vdash_\wedge \mu\alpha.[\beta]A' : \wedge_I\kappa_i\rightarrow\rho_i \mid \beta{:}\kappa',\Delta}{\Gamma \vdash_\wedge \mu\alpha.[\beta]A' : \kappa{\rightarrow}\rho \mid \beta{:}\kappa',\Delta}\,(\leq_\wedge)\ \ (\ddagger)}\,(\wedge)
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash P : \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m) \mid \alpha{:}\kappa'_i,\Delta}{\Gamma \vdash [\alpha]P : \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)\times\kappa'_i \mid \alpha{:}\kappa'_i,\Delta}\,(Cmd) \quad (\forall m \in M_i)}{\Gamma \vdash [\alpha]P : \wedge_{M_i}\wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)\times\kappa'_i \mid \alpha{:}\kappa'_i,\Delta}\,(\wedge)}{\Gamma \vdash [\alpha]P : (\kappa_i\rightarrow\rho_i)\times\kappa_i \mid \alpha{:}\kappa'_i,\Delta}\,(\leq_\wedge)\ \ (\dagger)}{\Gamma \vdash \mu\alpha.[\alpha]P : \kappa'_i\rightarrow\rho_i \mid \Delta}\,(\mu) \quad (\forall i \in I)}{\dfrac{\Gamma \vdash \mu\alpha.[\alpha]P : \wedge_I(\kappa'_i\rightarrow\rho_i) \mid \Delta}{\Gamma \vdash \mu\alpha.[\alpha]P : \kappa\rightarrow\rho \mid \Delta}\,(\leq_\wedge)\ \ (\ddagger)}\,(\wedge)
$$

Figure 4: Two derivations for the proof of Lemma 10.4

Let $\delta_i^m = \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)$, then (without loss of generality) by Lemma 10.3 the derivation for $\Gamma \vdash_\wedge \mu\alpha.[\alpha]P : \kappa\rightarrow\rho \mid \Delta$ is shaped as the left-hand derivation in Figure 4.

Notice that there exists some $a \in I$ such that $\kappa \leq_\wedge \kappa'_a$ and $\rho_a \leq_\wedge \rho$ by step (‡), and $\kappa'_a \leq_\wedge \kappa_a$, $\kappa_a \leq_\wedge \eta_{a,c}^b$ and $\rho_{a,c}^b \leq_\wedge \rho_a$, for some $b \in M_a$ and $c \in N_a^b$ by step (†). Then, by Lemma 10.2, $\Gamma \vdash_\wedge P : \eta_{a,c}^b{\rightarrow}\rho_{a,c}^b \mid \alpha{:}\kappa,\Delta$. Take $\kappa' = \eta_{a,c}^b$; notice that $\rho_{a,c}^b \leq_\wedge \rho_a \leq_\wedge \rho$, so by applying $(\leq_\wedge)$ we get $\Gamma \vdash_\wedge P : \kappa'{\rightarrow}\rho \mid \alpha{:}\kappa,\Delta$.

ii) Then by Lemma 10.3 there exists I such that for all $i \in I$ there are κ_i, κ'_i, ρ_i, and M_i such that for every $m \in M_i$ there are δ_i^m such that:

$$
\begin{array}{c}
\forall i \in I\,[\,\forall m \in M_i\ \ [\,\Gamma \vdash_\wedge A' : \delta_i^m \mid \alpha{:}\kappa_i,\beta{:}\kappa',\Delta\,]\ \& \\
\wedge_{J_i}\delta_i^m \times \kappa' \leq_\wedge (\kappa_i\rightarrow\rho_i)\times\kappa_i\,]\ \&\ \wedge_I \kappa'_i\rightarrow\rho_i \leq_\wedge \delta.
\end{array}
$$

Let $\delta_i^m = \wedge_{N_i^m}(\eta_{i,n}^m{\rightarrow}\rho_{i,n}^m)$, then the derivation for $\Gamma \vdash_\wedge \mu\alpha.[\beta]P : \kappa\rightarrow\rho \mid \beta{:}\kappa',\Delta$ is shaped (without loss of generality) as the right-hand derivation in Figure 4.

Notice that, by step (‡), there exists some $a \in I$ such that $\kappa \leq_\wedge \kappa_a$ and $\rho_a \leq_\wedge \rho$ and $\kappa' \leq_\wedge \kappa'_a$, $\kappa'_a \leq_\wedge \eta_{a,c}^b$, by step (†), and $\rho_{a,c}^b \leq_\wedge \rho_a$, for some $b \in M_a$ and $c \in N_a^b$. Then, by Lemma 10.2, $\Gamma \vdash_\wedge P : \eta_{a,c}^b{\rightarrow}\rho_{a,c}^b \mid \alpha{:}\kappa,\beta{:}\kappa',\Delta$. Take $\kappa'' = \eta_{a,c}^b$; notice that $\rho_{a,c}^b \leq_\wedge \rho_a \leq_\wedge \rho$, and by applying $(\leq_\wedge)$ we get $\Gamma \vdash_\wedge P : \kappa''\rightarrow\rho \mid \alpha{:}\kappa,\beta{:}\kappa',\Delta$.

A direct result of this lemma, in combination with Lemma 2.7, is that the type assignment system for '$\vdash_\wedge$', restricted to $\lambda\mu$, can be defined as follows (replacing rules (Cmd) and (μ) of '$\vdash_\wedge$'):

$$
\begin{array}{rlrl}
(Ax): & \dfrac{}{\Gamma,x{:}\delta \vdash x : \delta \mid \Delta} & (\omega): & \dfrac{}{\Gamma \vdash M : \omega \mid \Delta} \\[3ex]
(Abs): & \dfrac{\Gamma,x{:}\delta \vdash M : \kappa{\rightarrow}\rho \mid \Delta}{\Gamma \vdash \lambda x.M : \delta\times\kappa{\rightarrow}\rho \mid \Delta}\ (x \notin \Gamma) & (\mu_1): & \dfrac{\Gamma \vdash M : \kappa''{\rightarrow}\rho \mid \alpha{:}\kappa,\beta{:}\kappa',\Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \kappa{\rightarrow}\rho \mid \beta{:}\kappa',\Delta}\ (\alpha \notin \Delta, \kappa' \leq_\wedge \kappa'') \\[3ex]
(App): & \dfrac{\Gamma \vdash M : \delta\times\kappa{\rightarrow}\rho \mid \Delta \quad \Gamma \vdash N : \delta \mid \Delta}{\Gamma \vdash MN : \kappa{\rightarrow}\rho \mid \Delta} & (\mu_2): & \dfrac{\Gamma \vdash M : \kappa'{\rightarrow}\rho \mid \alpha{:}\kappa,\Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \kappa{\rightarrow}\rho \mid \Delta}\ (\alpha \notin \Delta, \kappa \leq_\wedge \kappa') \\[3ex]
(\wedge): & \dfrac{\Gamma \vdash M : \delta_1 \mid \Delta \quad \cdots \quad \Gamma \vdash M : \delta_n \mid \Delta}{\Gamma \vdash M : \delta_1 \cap \cdots \cap \delta_n \mid \Delta} & (\leq_\wedge): & \dfrac{\Gamma \vdash M : \delta \mid \Delta \quad \delta \leq_\wedge \delta'}{\Gamma \vdash M : \delta' \mid \Delta}
\end{array}
$$

bringing it closer to '$\vdash_{\mathrm{S}}$'.

We will also need the following result, which states that strict types are representatives of equivalence classes of types in '$\vdash_\wedge$' under $\sim_\wedge$. Remember that our strict negated types are a subset of the full intersection types, $\neg \mathtt{C} = \mathtt{C}\to v$, and that we read Ω as ω. Also, to simplify the following proof, we will assume that all $v \in \mathcal{T}_R$ are equivalent under $\sim_\wedge$.

Lemma 10.5 For every δ there exists $\mathtt{S}$ (called δ^*) such that $\delta \sim_\wedge \mathtt{S}$, and for every κ there exists $\mathtt{C}$ (called κ^*) such that $\kappa \sim_\wedge \mathtt{C}$.

Proof: By simultaneous induction on the structure of types.

$(\delta = \kappa\to\rho) : (\rho = v)$: By induction, there exists $\mathtt{C}$ such that $\kappa \sim_\wedge \mathtt{C}$; then $\neg\mathtt{C} \sim_\wedge \kappa\to v$.

$(\rho = \omega)$: Notice that $\omega \leq_\wedge \omega\to\omega \leq_\wedge \kappa\to\omega \leq_\wedge \omega$, so $\kappa\to\omega \sim_\wedge \omega$; take $\mathtt{S} = \omega$.

$(\rho = \rho_1 \wedge \rho_2)$: Notice that $\kappa\to\rho_1 \wedge \rho_2 \sim_\wedge (\kappa\to\rho_1) \wedge (\kappa\to\rho_2)$; by induction, there exist $\mathtt{S}_1$, $\mathtt{S}_2$ such that $\mathtt{S}_1 \sim_\wedge \kappa\to\rho_1$ and $\mathtt{S}_2 \sim_\wedge \kappa\to\rho_2$, so $\mathtt{S}_1 \cap \mathtt{S}_2 \sim_\wedge (\kappa\to\rho_1) \wedge (\kappa\to\rho_2) \sim_\wedge \kappa\to\rho_1 \wedge \rho_2$.

$(\delta = \omega)$: Take $\mathtt{S} = \omega$.

$(\delta = \delta_1 \wedge \delta_2)$: By induction, there exists $\mathtt{S}_1$, $\mathtt{S}_2$ such that $\mathtt{S}_1 \sim_\wedge \delta_1$ and $\mathtt{S}_2 \sim_\wedge \delta_2$. Take $\mathtt{S} = \mathtt{S}_1 \cap \mathtt{S}_2$.

$(\kappa = \delta \times \kappa)$: By induction, there exist $\mathtt{S}$, $\mathtt{D}$ such that $\mathtt{S} \sim_\wedge \delta$ and $\mathtt{D} \sim_\wedge \kappa$. Take $\mathtt{C} = \mathtt{S} \times \mathtt{D}$.

$(\kappa = \omega)$: Take $\mathtt{C} = \Omega$.

$(\kappa = \kappa_1 \wedge \kappa_2)$: By induction, there exists $\mathtt{C}_1$, $\mathtt{C}_2$ such that $\kappa_1 \sim_\wedge \mathtt{C}_1$ and $\kappa_2 \sim_\wedge \mathtt{C}_2$. Take $\mathtt{C} = \mathtt{C}_1 \cap \mathtt{C}_2$. □

We will also use the notation Γ^* and Δ^*; their intended meaning should be clear.

We now show that we can relate the systems '$\vdash_\wedge$' and '$\vdash_{\mathrm{S}}$' for approximants.

Lemma 10.6 $\Gamma \vdash_\wedge A : \delta \mid \Delta$ if and only if there exists $\mathtt{S}$, Γ', Δ' such that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, $\mathtt{S} \leq_\wedge \delta$, and $\Gamma' \vdash_{\mathrm{S}} A : \mathtt{S} \mid \Delta'$.

Proof: $(\Rightarrow)$: By induction on the height of derivations; notice that whenever a result is derived in a proper sub-derivation, its height is strictly smaller.

(Ax): Then $A = x$ and $x{:}\delta \in \Gamma$. By Lemma 3.8 we can derive $\Gamma^* \vdash_{\mathrm{S}} x : \delta^* \mid \Delta^*$. By Lemma 10.5 we have $\delta^* \sim_\wedge \delta$, so also $\delta^* \leq_\wedge \delta$; since $\Gamma^* \sim_\wedge \Gamma$ and $\Delta^* \sim_\wedge \Delta$, we also have $\Gamma \leq_\wedge \Gamma^*$ and $\Delta \leq_\wedge \Delta^*$.

(Abs): Then $A = \lambda x.A'$, $\delta = \delta_0 \times \kappa\to\rho$ and $\Gamma, x{:}\delta_0 \vdash_\wedge A' : \kappa\to\rho \mid \Delta$ in a sub-derivation. By induction, there exists $\mathtt{S}'$, Γ', δ_0', and Δ' such that $\Gamma, x{:}\delta_0 \leq_\wedge \Gamma', x{:}\delta_0'$, $\Delta \leq_\wedge \Delta'$, $\mathtt{S}' \leq_\wedge \kappa\to\rho$ and $\Gamma', x{:}\delta_0' \vdash_{\mathrm{S}} A' : \mathtt{S}' \mid \Delta'$.
Since $\mathtt{S}' \leq_\wedge \kappa\to\rho$, by Proposition 3.3:(*ii*) we can assume $\mathtt{S}' = \mathtt{A} = \neg\mathtt{C} = \mathtt{C}\to v \leq_\wedge \kappa\to\rho$. Then $\Gamma', x{:}\delta_0' \vdash_{\mathrm{S}} A' : \neg\mathtt{C} \mid \Delta'$ and by (Abs) also $\Gamma' \vdash_{\mathrm{S}} \lambda x.A' : \neg(\delta_0' \times \mathtt{C}) \mid \Delta'$. Observe that from $\Gamma, x{:}\delta_0 \leq_\wedge \Gamma', x{:}\delta_0'$ we have $\Gamma \leq_\wedge \Gamma'$ and $\delta_0 \leq_\wedge \delta_0'$, so also $\neg(\delta_0' \times \mathtt{C}) = \delta_0' \times \mathtt{C}\to v \leq_\wedge \delta_0 \times \kappa\to\rho \leq_\wedge \delta$.

(App): Then $A = xA_1 \cdots A_n$, $\delta = \kappa\to\rho$. By Lemma 10.3 there are $\delta_1, \ldots, \delta_n$ such that $\Gamma \vdash_\wedge A_i : \delta_i \mid \Delta$ in a sub-derivation, for all $i \in \underline{n}$, and $\Gamma \leq_\wedge \{ x{:}\delta_1 \times \cdots \times \delta_n \times \kappa\to\rho \}$. By induction, for all $i \in \underline{n}$ there exists $\mathtt{S}_i$, Γ_i, Δ_i such that $\Gamma \leq_\wedge \Gamma_i$, $\Delta \leq_\wedge \Delta_i$, $\mathtt{S}_i \leq_\wedge \delta_i$ and $\Gamma_i \vdash_{\mathrm{S}} A_j : \mathtt{S}_i \mid \Delta_i$. By Lemma 10.5 there exist $\mathtt{C}$ such that $\neg\mathtt{C} \sim_\wedge \kappa\to\rho$. By Lemma 3.8 we can derive $\cap_{i\in\underline{n}}\Gamma_i \cap \{ x{:}\neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \} \vdash_{\mathrm{S}} xA_1 \cdots A_n : \neg\mathtt{C} \mid \cap_{i\in\underline{n}}\Delta_i$. Since $\delta_1 \times \cdots \times \delta_n \times \kappa\to\rho \leq_\wedge \neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C})$ and $\Gamma \leq_\wedge \Gamma_i$, we also have $\Gamma \leq_\wedge \cap_n\Gamma_i \cap \{ x{:}\neg(\mathtt{S}_1 \times \cdots \times \mathtt{S}_n \times \mathtt{C}) \}$; also, $\neg\mathtt{C} \leq_\wedge \kappa\to\rho = \delta$, and $\Delta \leq_\wedge \cap_{\underline{n}}\Delta_i$.

$(\cap I)$: Then $\delta = \delta_1 \cap \cdots \cap \delta_n$, and, for every $i \in \underline{n}$, $\Gamma \vdash_\wedge A : \delta_i \mid \Delta$ in a sub-derivation. Then by induction there are $\mathtt{S}_i$, Γ_i, Δ_i such that $\Gamma \leq_\wedge \Gamma_i$, $\Delta \leq_\wedge \Delta_i$, $\mathtt{S}_i \leq_\wedge \delta_i$, and $\Gamma_i \vdash_{\mathrm{S}} A : \mathtt{S}_i \mid \Delta_i$. But by Lemma 3.8 then also $\cap_n\Gamma_i \vdash_{\mathrm{S}} A : \cap_n\mathtt{S}_i \mid \cap_n\Delta_i$; notice that $\Gamma \leq_\wedge \cap_n\Gamma_i$, $\Delta \leq_\wedge \cap_n\Delta_i$, and $\cap_n \mathtt{S}_i \leq \cap_n \delta_i = \delta$.

(ω): Take $\Gamma' = \emptyset = \Delta'$, and $\mathtt{S} = \omega$; notice that $\emptyset \vdash_{\mathrm{S}} A : \omega \mid \emptyset$ and that $\Gamma \leq_\wedge \emptyset$ and $\Delta \leq_\wedge \emptyset$.

(μ_1): Then $A = \mu\alpha.[\beta]A'$, with $\alpha \neq \beta$, $\delta = \kappa{\rightarrow}\rho$ and there exists κ', κ'' such that $\kappa' \leq_\wedge \kappa''$ and $\Gamma \vdash_\wedge A' : \kappa'' {\rightarrow} \rho \mid \alpha{:}\kappa, \beta{:}\kappa', \Delta$ in a sub-derivation. Then, by induction, there exist $\mathtt{S}'$, Γ', $\mathtt{C}$, $\mathtt{C}'$, and Δ' such that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, $\kappa \leq_\wedge \mathtt{C}$, $\kappa' \leq_\wedge \mathtt{C}'$, $\mathtt{S}' \leq_\wedge \kappa''{\rightarrow}\rho$, and $\Gamma' \vdash_{\mathrm{S}} A' : \mathtt{S}' \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}', \Delta'$.
Since $\mathtt{S}' \leq_\wedge \kappa''{\rightarrow}\rho$, we can assume $\mathtt{S}' = \mathtt{A} = \neg\mathtt{D} = \mathtt{D}{\rightarrow}\upsilon \leq_\wedge \kappa''{\rightarrow}\rho$, so $\kappa'' \leq_\wedge \mathtt{D}$, and $\upsilon \leq_\wedge \rho$. By Lemma 3.9 $\Gamma' \vdash_{\mathrm{S}} A' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C}, \beta{:}\mathtt{C}' \cap \mathtt{D}, \Delta'$, so by (μ_1) we obtain $\Gamma' \vdash_{\mathrm{S}} \mu\alpha.[\beta]A' : \neg\mathtt{C} \mid \beta{:}\mathtt{C}' \cap \mathtt{D}, \Delta'$. Notice that $\Gamma \leq_\wedge \Gamma'$, $\beta{:}\kappa', \Delta \leq_\wedge \beta{:}\mathtt{C}' \cap \mathtt{D}, \Delta'$, since both $\kappa' \leq_\wedge \mathtt{C}'$ and $\kappa' \leq_\wedge \kappa'' \leq_\wedge \mathtt{D}$, and $\neg\mathtt{C} \leq_\wedge \kappa{\rightarrow}\rho$, since $\kappa \leq_\wedge \mathtt{C}$ and $\upsilon \leq_\wedge \rho$.

(μ_2): Then $A = \mu\alpha.[\alpha]A'$ and $\delta = \kappa{\rightarrow}\rho$ and there exists κ' such that $\kappa \leq_\wedge \kappa'$ and $\Gamma \vdash_\wedge A' : \kappa' {\rightarrow} \rho \mid \alpha{:}\kappa, \Delta$ in a sub-derivation. Then, by induction, there exist $\mathtt{S}'$, Γ', $\mathtt{C}$, and Δ' such that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, $\kappa \leq_\wedge \mathtt{C}$, $\mathtt{S}' \leq_\wedge \kappa'{\rightarrow}\rho$, and $\Gamma' \vdash_{\mathrm{S}} A' : \mathtt{S}' \mid \alpha{:}\mathtt{C}, \Delta'$. As above, since $\mathtt{S}' \leq_\wedge \kappa'{\rightarrow}\rho$, we can assume $\mathtt{S}' = \mathtt{A} = \neg\mathtt{D} \leq_\wedge \kappa'{\rightarrow}\rho$, so $\kappa' \leq_\wedge \mathtt{D}$, and $\upsilon \leq_\wedge \rho$. By Lemma 3.9 we also have $\Gamma' \vdash_{\mathrm{S}} A' : \neg\mathtt{D} \mid \alpha{:}\mathtt{C} \cap \mathtt{D}, \Delta'$, so by (μ_2) we get $\Gamma' \vdash_{\mathrm{S}} \mu\alpha.[\alpha]A' : \neg(\mathtt{C} \cap \mathtt{D}) \mid \Delta'$. Notice that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, and $\neg(\mathtt{C} \cap \mathtt{D}) \leq_\wedge \kappa{\rightarrow}\rho$, since $\kappa \leq_\wedge \mathtt{C}$ and $\kappa \leq_\wedge \kappa' \leq_\wedge \mathtt{D}$, and $\upsilon \leq_\wedge \rho$.

$(\leq_\wedge)$: Then there exists δ' such that $\delta' \leq_\wedge \delta$ and $\Gamma \vdash A : \delta' \mid \Delta$ in a sub-derivation. Then, by induction, there exist $\mathtt{S}'$, Γ', and Δ' such that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, $\mathtt{S}' \leq_\wedge \delta'$, and $\Gamma' \vdash_{\mathrm{S}} A : \mathtt{S}' \mid \Delta'$. Since $\delta' \leq_\wedge \delta$, also $\mathtt{S}' \leq_\wedge \delta$.

$(\Leftarrow)$: Immediate, since '$\vdash_{\mathrm{S}}$' is a subsystem of '$\vdash_\wedge$', and Lemma 10.2.

We can now state the exact relation between '$\vdash_\wedge$' and '$\vdash_{\mathrm{S}}$'.

Theorem 10.7 $\Gamma \vdash_\wedge M : \delta \mid \Delta \iff \exists\, \mathtt{S}, \Gamma', \Delta'\ [\Gamma \leq_\wedge \Gamma' \;\&\; \Delta \leq_\wedge \Delta' \;\&\; \mathtt{S} \leq_\wedge \delta \;\&\; \Gamma' \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta]$.

Proof:
$$
\begin{array}{lcl}
\Gamma \vdash_\wedge M : \delta \mid \Delta & \iff & (10.1) \\
\exists A \in \mathcal{A}(M)\ [\Gamma \vdash_\wedge A : \delta \mid \Delta] & \iff & (10.6) \\
\exists\, \mathtt{S}, \Gamma', \Delta', A \in \mathcal{A}(M)\ [\Gamma \leq_\wedge \Gamma' \;\&\; \Delta \leq_\wedge \Delta' \;\&\; \mathtt{S} \leq_\wedge \delta \;\&\; \Gamma' \vdash_{\mathrm{S}} A : \mathtt{S} \mid \Delta'] & \iff & (8.4) \\
\exists\, \mathtt{S}, \Gamma', \Delta'\ [\Gamma \leq_\wedge \Gamma' \;\&\; \Delta \leq_\wedge \Delta' \;\&\; \mathtt{S} \leq_\wedge \delta \;\&\; \Gamma' \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta'] & &
\end{array}
$$

Notice that this result states that a derivation in '$\vdash_\wedge$' can be represented by one in '$\vdash_{\mathrm{S}}$' (which is a derivation in '$\vdash_\wedge$' as well), with the rule $(\leq_\wedge)$ applied only to term variables or as the last step. So in a sense, '$\vdash_{\mathrm{S}}$' is the 'kernel' of '$\vdash_\wedge$'.

It is now straightforward to show the following characterisation results for '$\vdash_\wedge$' as well.

Theorem 10.8 *i) If $\Gamma \vdash_\wedge M : \delta \mid \Delta$ and $M =_{\beta\mu} N$, then $\Gamma \vdash_\wedge N : \delta \mid \Delta$.*

ii) There exists Γ, $\delta \neq \omega$, and Δ such that $\Gamma \vdash_\wedge M : \delta \mid \Delta$, if and only if M has a head normal form.

iii) There exists ω-free Γ, Δ, and δ such that $\Gamma \vdash_\wedge M : \delta \mid \Delta$, if and only if M has a normal form.

Proof: i) By Theorem 10.7 and the corresponding result for '$\vdash_{\mathrm{S}}$', Theorem 4.5 and 8.4, respectively.

ii)(*only if*): If $\Gamma \vdash_\wedge M : \delta \mid \Delta$, then, by Theorem 10.7, there exists $\mathtt{S}$, Γ', Δ' such that $\Gamma \leq_\wedge \Gamma'$, $\Delta \leq_\wedge \Delta'$, $\mathtt{S} \leq_\wedge \delta$, and $\Gamma' \vdash_{\mathrm{S}} M : \mathtt{S} \mid \Delta'$. Since $\delta \neq \omega$, there exist $\mathtt{A}_1, \ldots, \mathtt{A}_n$ with $n > 0$ such that $\mathtt{S} = \cap_{\underline{n}} \mathtt{A}_i$ and $\Gamma' \vdash_{\mathrm{S}} M : \mathtt{A}_i \mid \Delta'$ for all $i \in \underline{n}$. Then by Theorem 8.5, M has a head normal form.

(*if*): If M has a head normal form then by Theorem 8.5 there exist Γ, Δ, and $\mathtt{A}$ such that $\Gamma \vdash_{\mathrm{S}} M : \mathtt{A} \mid \Delta$. Since every derivation in '$\vdash_{\mathrm{S}}$' corresponds to one in '$\vdash_\wedge$', the result follows directly.

iii) Much the same as the previous point. Notice that, by construction of the proofs of

Lemma 10.5 and 10.6, ω is never selected as a type to construct Γ', Δ or S, and therefore for any T and δ used in either, if $T \leq_{\wedge} \delta$ or $\delta \leq_{\wedge} T$ and δ is ω-free, then so is T. Observe that, as already remarked after the proof of Theorem 9.8, ω might appear inside the derivation.

The equivalent of the approximation result for '$\vdash_{\wedge}$' was already shown by deLiguoro [31] (cited above in Definition 10.1), and is needed to show Theorem 10.7. The equivalent of the characterisation of strong normalisation, Theorem 9.12, for '$\vdash_{\wedge}$' was shown in [12]. Since ω might be introduced inside a derivation constructed in the proof of Theorem 4.5 (that is used in the proof of Theorem 8.4, which is used in the proof of Theorem 10.7), this result does not follow directly from Theorem 10.7.

11 On type variables

We should point out that, since reflecting directly the structure of the domain, the notion of type in this paper (and that of [12]) is rather non-standard, in that types are defined without type variables. As far as the construction of a filter model is concerned, this creates no problems, but it is now impossible to define a notion of principal types in the traditional way, *i.e.* based around the operation of type-substitution, that replaces type variables by types. Moreover, it is now not clear how to relate this notion of type assignment to the more familiar ones as appeared in [18, 2, 33].

When adding type variables to the negated types we have considered so far, we would aim to derive, for example,

$$\dfrac{\dfrac{\dfrac{}{x{:}\neg(\neg\varphi\times\varphi')\cap\neg\varphi \vdash x : \neg(\neg\varphi\times\varphi') \mid \emptyset} \qquad \dfrac{}{x{:}\neg(\neg\varphi\times\varphi')\cap\neg\varphi \vdash x : \neg\varphi \mid \emptyset}}{x{:}\neg(\neg\varphi\times\varphi')\cap\neg\varphi \vdash xx : \neg\varphi' \mid \emptyset}}{\emptyset \vdash \lambda x.xx : \neg((\neg(\neg\varphi\times\varphi')\cap\neg\varphi)\times\varphi') \mid \emptyset}$$

(notice the absence of Ω). This example suggests that a (non-negated) type variable should be a continuation type, and comes at the end. So adding type variables seems to lead to the following definition of types:

$$\begin{array}{rcll} A,B & ::= & \neg\mathtt{C} & \\ R,S,T & ::= & \omega \mid A_1\cap\cdots\cap A_n & (n\geq 1) \\ \mathtt{C},\mathtt{D} & ::= & \varphi \mid S\times\mathtt{C} & \end{array}$$

Now the problem is that we can no longer see $\mathtt{C}\cap\mathtt{D}$ as a short hand of the 'zipped up' version of $\mathtt{C}$ and $\mathtt{D}$ that we considered above. Take for example:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x{:}\neg\varphi \vdash x : \neg\varphi \mid \alpha{:}\varphi',\gamma{:}(\neg\varphi\cap\varphi')\times\varphi}\,(Ax)}{\emptyset \vdash \lambda x.x : \neg(\neg\varphi\times\varphi) \mid \alpha{:}\varphi',\gamma{:}(\neg\varphi\cap\varphi')\times\varphi}\,(Abs)}{\emptyset \vdash \mu\alpha.[\gamma]\lambda x.x : \neg\varphi' \mid \gamma{:}(\neg\varphi\cap\varphi')\times\varphi}\,(\mu_1)\quad ((\neg\varphi\cap\varphi')\times\varphi \leq_{\mathrm{s}} \neg\varphi\times\varphi)}{\emptyset \vdash \mu\gamma.[\gamma]\mu\alpha.[\gamma]\lambda x.x : \neg((\neg\varphi\cap\varphi')\times\varphi) \mid \emptyset}\,(\mu_2)\quad ((\neg\varphi\cap\varphi')\times\varphi \leq_{\mathrm{s}} \varphi')$$

Notice that the 'zipped' version of the two types for γ, $\neg\varphi\times\varphi$ and φ', *i.e.* $(\neg\varphi\cap\varphi')\times\varphi$ as used in the derivation above, is not a continuation type: the intersection of $\neg\varphi$ and φ' is not an intersection of negated continuation types.

This forces us to add the intersection of continuation types explicitly and has an impact on the inference rules: since a continuation type can be an intersection, a basic type can be of the shape $\neg(\mathtt{C}\cap\mathtt{D})$, so we no longer have that a term in the left-hand side of an application has a type of the shape $\neg(S \times C)$. In fact, it will now have a type of the shape $\neg((S_1 \times C_1) \cap \cdots \cap (S_n \times C_n))$, and we can safely apply this only to a term that has *all the types* S_i $(i \in \underline{n})$. This yields:

Definition 11.1 *i*) We define strict negated intersection types with type variables through the grammar:

$$\begin{array}{rcll} A,B & ::= & \neg\mathbf{C} & & (\textit{negated types}) \\ R,S,T & ::= & \omega \mid A_1 \cap \cdots \cap A_n & (n \geq 1) & (\textit{intersection types}) \\ C,D & ::= & \varphi \mid S \times C & & (\textit{strict continuation types}) \\ \mathbf{C},\mathbf{D} & ::= & C_1 \cap \cdots \cap C_n & (n \geq 1) & (\textit{continuation types}) \end{array}$$

ii) The type inclusion relation is defined by:

$$\frac{}{\cap_I A_i \leq \cap_J A_j}\ (J \subseteq I) \qquad \frac{S \leq A_i \quad (\forall i \in I)}{S \leq \cap_I A_i}$$

$$\frac{}{\varphi \leq \varphi} \qquad \frac{}{\cap_I C_i \leq \cap_J C_j}\ (J \subseteq I) \qquad \frac{\mathbf{C} \leq C_i \quad (\forall i \in I)}{\mathbf{C} \leq \cap_I C_i}\ (I \neq \emptyset) \qquad \frac{S_i \leq_{\mathrm{S}} T_i \quad (\forall i \in \underline{m})}{S_1 \times \cdots \times S_n \leq_{\mathrm{S}} T_1 \times \cdots \times T_m}\ (n \geq m)$$

iii) Type assignment is defined through the rules:

$$(Ax):\ \frac{}{\Gamma,x{:}S \vdash x : A \mid \Delta}\ (S \leq_{\mathrm{S}} A) \qquad (\cap):\ \frac{\Gamma \vdash M : A_i \mid \Delta \quad (\forall i \in I)}{\Gamma \vdash M : \cap_I A_i \mid \Delta}\ (I = \emptyset \vee |I| \geq 2)$$

$$(Abs):\ \frac{\Gamma,x{:}S \vdash M : \neg C \mid \Delta}{\Gamma \vdash \lambda x.M : \neg(S \times C) \mid \Delta}\ (x \notin \Gamma) \qquad (\mu_1):\ \frac{\Gamma \vdash M : \neg\mathbf{D} \mid \alpha{:}\mathbf{C},\beta{:}\mathbf{C}',\Delta}{\Gamma \vdash \mu\alpha.[\beta]M : \neg\mathbf{C} \mid \beta{:}\mathbf{C}',\Delta}\ \begin{array}{l}(\beta \neq \alpha \notin \Delta, \\ \mathbf{C}' \leq_{\mathrm{S}} \mathbf{D})\end{array}$$

$$(App):\ \frac{\Gamma \vdash M : \neg(\cap_n (S_i \times C_i)) \mid \Delta \quad \Gamma \vdash N : S_i \mid \Delta\ (\forall i \in \underline{n})}{\Gamma \vdash MN : \neg(\cap_n C_i) \mid \Delta} \qquad (\mu_2):\ \frac{\Gamma \vdash M : \neg\mathbf{D} \mid \alpha{:}\mathbf{C},\Delta}{\Gamma \vdash \mu\alpha.[\alpha]M : \neg\mathbf{C} \mid \Delta}\ (\alpha \notin \Delta, \mathbf{C} \leq_{\mathrm{S}} \mathbf{D})$$

We use '$\vdash_{\mathrm{S}}^{\varphi}$' for derivable judgements in this system.

Apart from the fact that negated types are used, and intersection is used to group continuation types, we can show that this system is actually that one defined in [6], which we will now quickly review.

In [6] a notion of intersection and union type assignment was presented for $\lambda\mu$, inspired by similar notions for $\mathcal{X}$ [8] and $\lambda\mu\tilde{\mu}$ [5]. The main result shown there is that this system is closed for conversion.

The point of view of [6] is to see $\lambda\mu$'s context variables α as names for possible continuations that in the philosophy of intersection types need not all be typed with the same type; therefore multiple types are allowed for context variables in the environment Δ. As in [5, 8], these types are grouped using a new type constructor, called union mainly for historical reasons. This union type constructor is not the standard one, since the system has no 'normal' rules for for union, traditionally formulated (as in [16], for example) via

$$(\cup I):\ \frac{\Gamma \vdash M : A \mid \Delta}{\Gamma \vdash M : A \cup B \mid \Delta} \qquad (\cup E):\ \frac{\Gamma \vdash N : A \cup B \mid \Delta \quad \Gamma,x{:}A \vdash M : C \mid \Delta \quad \Gamma,x{:}B \vdash M : C \mid \Delta}{\Gamma \vdash M\{N/x\} : C \mid \Delta}$$

These create the subject-reduction problem dealt with in that paper by considering parallel reduction.

Binding a context variable as in $\mu\alpha.[\beta]M$ then naturally has a union type $\cup_{\underline{n}} A_i$; reduction of the term $(\mu\alpha.[\beta]M)N$ will bring the operand N to each of the (pseudo) sub-terms in M of the shape $[\alpha]Q$, where Q has type A_i; since N gets placed behind Q, this implies that $A_i = C_i \rightarrow B_i$ and that therefore the type for α should be $\cup_{\underline{n}}(C_i \rightarrow B_i)$; this then also implies that N should have all the types $C_i\ (\forall i \in \underline{n})$; $(\rightarrow E)$ as below (Definition 11.3) expresses exactly that. The only 'functionality' we need for union types therefore is the ability to choose a collection of types for α amongst those stored in Δ; this is represented by $(\cup E)$.

Definition 11.2 (THE SYSTEM $\vdash_{\cap\cup}$) *i*) The set of types for intersection-union type assignment is defined by the grammar:

$$\begin{array}{rcll} A, B & ::= & \varphi \mid B_1 \cup \cdots \cup B_m \mid \sigma \rightarrow B & (m \geq 0) \quad (\textit{strict types}) \\ \sigma & ::= & A_1 \cap \cdots \cap A_n & (n \geq 0) \end{array}$$

As above, we call $A_1 \cap \cdots \cap A_n$ (with $n \geq 0$) an intersection type, and call $B_1 \cup \cdots \cup B_m$ (with $m \geq 0$) a *union* type; we use $\top$ for the empty intersection type, and $\bot$ for the empty union type.

ii) The relation '$\leq$' is defined as:

$$\begin{array}{rcll} A_1 \cap \cdots \cap A_n & \leq & A_i, & (\text{for all } i \in \underline{n},\ n \geq 1) \\ \sigma \leq A_i, \text{ for all } i \in \underline{n} & \Rightarrow & \sigma \leq A_1 \cap \cdots \cap A_n, & (n \geq 0) \\ B_j & \leq & B_1 \cup \cdots \cup B_m, & (\text{for all } j \in \underline{m},\ m \geq 1) \\ B_j \leq \sigma, \text{ for all } j \in \underline{m} & \Rightarrow & B_1 \cup \cdots \cup B_m \leq \sigma, & (m \geq 0) \end{array}$$

iii) A *left environment* Γ is a partial mapping from term variables to intersections of strict types, and we write $x{:}\sigma \in \Gamma$ if $\Gamma(x) = \sigma$. Similarly, a *right environment* Δ contains only strict types, which can be union types.

Notice that we consider union types to be strict as well; this implies that we allow an intersection of union types, a union of union types, but not a union of intersection types.

Definition 11.3 (THE SYSTEM '$\vdash_{\cap\cup}$') Intersection-union type assignment for $\lambda\mu$ is defined via the inference rules:

$$(\cap E): \frac{}{\Gamma, x{:}\cap_{\underline{n}} A_i \vdash x : A_i \mid \Delta}\ (i \in \underline{n}) \qquad (\cap I): \frac{\Gamma \vdash M : A_i \mid \Delta \quad (\forall i \in \underline{n})}{\Gamma \vdash M : \cap_{\underline{n}} A_i \mid \Delta}\ (n \geq 0, n \neq 1)$$

$$(\rightarrow I): \frac{\Gamma, x{:}\sigma \vdash M : B \mid \Delta}{\Gamma \vdash \lambda x.M : \sigma \rightarrow B \mid \Delta}\ (x \notin \Gamma) \qquad (\rightarrow E): \frac{\Gamma \vdash M : \cup_J (\sigma_j \rightarrow B_j) \mid \Delta \quad \Gamma \vdash N : \sigma_j \mid \Delta \ (\forall j \in J)}{\Gamma \vdash MN : \cup_J B_j \mid \Delta}\ (n \geq 1)$$

$$(\cup E): \frac{\Gamma \vdash M : \cup_{\underline{m}} B_j \mid \beta{:}\cup_{\underline{n}} A_i, \alpha{:}B, \Delta}{\Gamma \vdash \mu\alpha.[\beta]M : B \mid \beta{:}\cup_{\underline{n}} A_i, \Delta}\ (\cup_{\underline{m}} B_j \leq \cup_{\underline{n}} A_i) \qquad \frac{\Gamma \vdash M : \cup_{\underline{m}} B_j \mid \beta{:}\cup_{\underline{n}} A_i, \Delta}{\Gamma \vdash \mu\beta.[\beta]M : \cup_{\underline{n}} A_i \mid \Delta}\ (\cup_{\underline{m}} B_j \leq \cup_{\underline{n}} A_i)$$

We write $\Gamma \vdash_{\cap\cup} M : A \mid \Delta$ if this statement is derivable using these rules.

Notice that the traditional $(\rightarrow E)$ is obtained by taking $n = 1$ in the corresponding rule above. Moreover, all σ_j can be intersection types, so each can be $\top$; this is why that rule is not formulated using $\Gamma \vdash_{\cap\cup} N : \cap_J \sigma_j \mid \Delta$. Moreover, if $x{:}\cup_{\underline{m}} B_j \in \Gamma$, then it is only possible to derive $\Gamma \vdash_{\cap\cup} x : \cup_{\underline{m}} B_j \mid \Delta$, *i.e.* we have no way of eliminating a union assigned to a term variable. Since in '$\vdash_{\mathrm{S}}$' we do not allow real intersections of continuation types, our approach differs significantly from that of '$\vdash_{\cap\cup}$'; in a certain sense, in '$\vdash_{\mathrm{S}}$' we use streams of intersection types, whereas in '$\vdash_{\cap\cup}$' intersections of stream types (modelled using union) are used.

The main result shown in [6] is:

Theorem 11.4 (SUBJECT REDUCTION AND EXPANSION) *If $M \rightarrow_{\beta\mu} N$, then $\Gamma \vdash_{\cap\cup} M : \sigma \mid \Delta$ if and only if $\Gamma \vdash_{\cap\cup} N : \sigma \mid \Delta$.*

We can map '$\vdash_{\cap\cup}$' into '$\vdash_{\mathrm{S}}^{\varphi}$', for which we define an interpretation of types.

Definition 11.5 (CF. [29]) The interpretation of types of '$\vdash_{\cap\cup}$' to those of '$\vdash_{\mathrm{S}}^{\varphi}$' is defined by:

$$\begin{array}{rcl} A^- & \triangleq & \neg(A^+) \\ (A_1\cap\cdots\cap A_n)^- & \triangleq & A_1^-\cap\cdots\cap A_n^- \quad (n\geq 2) \\ \top^- & \triangleq & \omega \end{array} \qquad \begin{array}{rcl} \varphi_i^+ & \triangleq & \varphi_i \\ (A\to B)^+ & = & A^-\times B^+ \\ (A_1\cup\cdots\cup A_n)^+ & \triangleq & A_1^+\cap\cdots\cap A_n^+ \quad (n\geq 2) \\ \bot^+ & \triangleq & \omega \end{array}$$

and define $\Gamma^- \triangleq \{\, x{:}A^- \mid x{:}A \in \Gamma \,\}$ and $\Delta^+ \triangleq \{\, \alpha{:}A^+ \mid \alpha{:}A \in \Delta \}$.

We can now show:

Theorem 11.6 *If $\Gamma \vdash_{\cap\cup} M : A \mid \Delta$, then $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} M : A^- \mid \Delta^+$.*

Proof: By induction on the structure of derivations.

$(\cap E)$: Then $M \equiv x$, $\Gamma = \Gamma', x{:}A_1\cap\cdots\cap A_n$ with $n \geq 1$, and $A = A_i$, for some $i \in \underline{n}$. Then $\Gamma^- = \Gamma'^-, x{:}A_1^-\cap\cdots\cap A_n^-$; notice that $A_1^-\cap\cdots\cap A_n^- \leq_{\mathrm{s}} A_i^-$, so $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} x : A^- \mid \Delta^+$.

$(\cap I)$: Then $A = A_1\cap\cdots\cap A_n$ with $n \geq 2$, and $\Gamma \vdash_{\lambda\cap} M{:}A_i$, for all $i \in \underline{n}$. Then by induction $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} M : A_i^- \mid \Delta^+$, and $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} M : A^- \mid \Delta^+$ follows by $(\cap)$.

$(\to I)$: Then $M = \lambda x.N$, $A = B\to C$, and $\Gamma, x{:}B \vdash_{\lambda\cap} N{:}C$; by induction, $\Gamma^-, x{:}B^- \vdash_{\mathrm{s}}^{\varphi} N : C^- \mid \Delta^+$. Since $(B\to C)^- = \neg(B^-\times C^+)$, and $C^- = \neg(C^+)$, by applying (Abs) we get $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} \lambda x.N : B^- \times \neg(C^+) \mid \Delta^+$, so $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} \lambda x.N : A^- \mid \Delta^+$.

$(\to E)$: Then $M = PQ$, and there exists B such that $\Gamma \vdash_{\lambda\cap} P{:}B\to A$ and $\Gamma \vdash_{\lambda\cap} Q{:}B$. Then by induction $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} P : (B\to A)^- \mid \Delta^+$ and $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} Q : B^- \mid \Delta^+$. Since $(B\to A)^- = \neg(B^-\times A^+)$, by (App) we get $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} PQ : \neg(A^+) \mid \Delta^+$, so $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} PQ : A^- \mid \Delta^+$.

$(\cup E)$: We have two cases:

$(M = \mu\alpha.[\beta]N$, *with* $\alpha \neq \beta)$: Then $\Delta = \beta{:}\cup_{\underline{n}}A_i, \Delta'$, and $\Gamma \vdash_{\cap\cup} N : \cup_{\underline{m}}B_j \mid \beta{:}\cup_{\underline{n}}A_i, \alpha{:}A, \Delta'$ and $\cup_{\underline{m}}B_j \leq \cup_{\underline{n}}A_i$. Then, by induction, $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} N : (\cup_{\underline{m}}B_j)^- \mid \beta{:}(\cup_{\underline{n}}A_i)^+, \Delta'^+$, so

$$\Gamma^- \vdash_{\mathrm{s}}^{\varphi} N : \neg(B_1^+\cap\cdots\cap B_m^+) \mid \beta{:}A_1^+\cap\cdots\cap A_n^+, \alpha{:}A^+, \Delta'^+.$$

Notice that $\cap_{\underline{m}}B_j^+ \leq_{\mathrm{s}} \cap_{\underline{n}}A_i^+$, so by (μ_1) we get

$$\Gamma^- \vdash_{\mathrm{s}}^{\varphi} \mu\alpha.[\beta]N : \neg(A^+) \mid \beta{:}A_1^+\cap\cdots\cap A_n^+, \Delta'^+$$

so $\Gamma^- \vdash_{\cap\cup} \mu\alpha.[\beta]N : A^- \mid \beta{:}(\cup_{\underline{n}}A_i)^+, \Delta'^+$.

$(M = \mu\beta.[\beta]N)$: Then $A = \cup_{\underline{n}}A_i$, $\Gamma \vdash_{\cap\cup} N : \cup_{\underline{m}}B_j \mid \beta{:}\cup_{\underline{n}}A_i, \Delta$ and $\cup_{\underline{m}}B_j \leq \cup_{\underline{n}}A_i$. Then, by induction, $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} N : (\cup_{\underline{m}}B_j)^- \mid \beta{:}(\cup_{\underline{n}}A_i)^+, \Delta^+$, so $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} N : \neg(\cap_{\underline{m}}B_j^+) \mid \beta{:}\cap_{\underline{n}}A_i^+, \Delta^+$. Notice that $\cap_{\underline{m}}B_j^+ \leq_{\mathrm{s}} \cap_{\underline{n}}A_i^+$, so by (μ_2) we get $\Gamma^- \vdash_{\mathrm{s}}^{\varphi} \mu\beta.[\beta]N : \neg(\cap_{\underline{n}}A_i^+) \mid \Delta^+$ so $\Gamma^- \vdash_{\cap\cup} \mu\beta.[\beta]N : A^- \mid \Delta^+$. □

So the version of '$\vdash_{\mathrm{s}}$' extended with type variables corresponds to '$\vdash_{\cap\cup}$'; it seems obvious that it is possible to show all the characterisation results of this paper for '$\vdash_{\mathrm{s}}^{\varphi}$' as well, but will skip those results here.

Conclusions and future work

We have shown that a strict version of the intersection type system for $\lambda\mu$ of [12] is as expressive as the full version, by showing that it is closed under conversion. We have shown that derivation reduction (a kind of cut-elimination) is strongly normalisable, and that a number of characterisation properties follow from that as a direct consequence. We have shown that the system without the type constant ω characterises the strongly normalisable terms and that we can characterise normalisation as well. We have also shown an approximation theorem, and from that a characterisation of head normalisation.

We have investigated the relation between the full system of [12] and the one presented here, and shown, through the approximation result, that a derivation in the full system essentially contains a derivation in the strict system, with the rule $(\leq_\wedge)$ applied only on the outside. We also compared the strict system with that of [6], and found that the latter corresponds to the strict system, extended with type variables.

We will investigate the definition of a strict filter semantics for $\lambda\mu$, as well as the structure of the domain, if any, that corresponds to the intersection type theory defined here.

References

[1] Z.M. Ariola and H. Herbelin. Minimal Classical Logic and Control Operators. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, and G.J. Woeginger, editors, *Proceedings of Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 871–885. Springer Verlag, 2003.

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

[3] S. van Bakel. Intersection Type Assignment Systems. *Theoretical Computer Science*, 151(2):385–435, 1995.

[4] S. van Bakel. The Heart of Intersection Type Assignment; Normalisation proofs revisited. *Theoretical Computer Science*, 398:82–94, 2008.

[5] S. van Bakel. Completeness and Partial Soundness Results for Intersection & Union Typing for $\lambda\mu\tilde{\mu}$. *Annals of Pure and Applied Logic*, 161:1400–1430, 2010.

[6] S. van Bakel. Sound and Complete Typing for $\lambda\mu$. In *Proceedings of 5th International Workshop* Intersection Types and Related Systems *(ITRS'10), Edinburgh, Scotland*, volume 45 of *Electronic Proceedings in Theoretical Computer Science*, pages 31–44, 2010.

[7] S. van Bakel. Strict intersection types for the Lambda Calculus. *ACM Computing Surveys*, 43:20:1–20:49, April 2011.

[8] S. van Bakel. Completeness and Soundness results for $\mathcal{X}$ with Intersection and Union Types. *Fundamenta Informaticae*, 121:1–41, 2012.

[9] S. van Bakel. Approximation and (Head) Normalisation for $\lambda\mu$ using Strict Intersection Types. volume 242 of *Electronic Proceedings in Theoretical Computer Science*, 2016.

[10] S. van Bakel, F. Barbanera, and U. de'Liguoro. A Filter Model for $\lambda\mu$. In L. Ong, editor, *Proceedings of* 10th International Conference on Typed Lambda Calculi and Applications *(TLCA'11)*, volume 6690 of *Lecture Notes in Computer Science*, pages 213–228. Springer Verlag, 2011.

[11] S. van Bakel, F. Barbanera, and U. de'Liguoro. Characterisation of Strongly Normalising $\lambda\mu$-Terms. In *Proceedings of 6th International Workshop* Intersection Types and Related Systems *(ITRS'12), Dubrovnik, Croatia, June 29th*, volume 121 of *Electronic Proceedings in Theoretical Computer Science*, pages 31–44, 2012.

[12] S. van Bakel, F. Barbanera, and U. de'Liguoro. Intersection types for $\lambda\mu$. *Logical Methods in Computer Science*, 2017. To appear.

[13] S. van Bakel, S. Lengrand, and P. Lescanne. The language $\mathcal{X}$: Circuits, Computations and Classical Logic. In M. Coppo, E. Lodi, and G.M. Pinna, editors, *Proceedings of Ninth Italian Conference on Theoretical Computer Science* (ICTCS'05), *Siena, Italy*, volume 3701 of *Lecture Notes in Computer Science*, pages 81–96. Springer Verlag, 2005.

[14] S. van Bakel and P. Lescanne. Computation with Classical Sequents. *Mathematical Structures in Computer Science*, 18:555–609, 2008.

[15] S. van Bakel and M.G. Vigliotti. A fully abstract semantics of $\lambda\mu$ in the π-calculus. In *Proceedings of Sixth International Workshop on Classical Logic and Computation 2014* (CL&C'14), *Vienna, Austria*, volume 164 of *Electronic Proceedings in Theoretical Computer Science*, pages 33–47, 2014.

[16] F. Barbanera, M. Dezani-Ciancaglini, and U. de'Liguoro. Intersection and Union Types: Syntax and Semantics. *Information and Computation*, 119(2):202–230, 1995.

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

[18] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic*, 48(4):931–940, 1983.

[19] C. Böhm. Alcune propietá delle forme $\beta\eta$-normali nel λk-calcolo. *Pubblicazioni 696, Instituto Nazionale per le Applicazioni del Calcolo. Roma*, 1968.

[20] A. Church. A Note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41, 1936.

[21] M. Coppo and M. Dezani-Ciancaglini. A New Type Assignment for λ-Terms. *Archiv für Mathematische Logic und Grundlagen Forschung*, 19:139–156, 1978.

[22] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, and G. Longo. Extended type structures and filter lambda models. In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium 82*, pages 241–262, Amsterdam, the Netherlands, 1984. North-Holland.

[23] P.-L. Curien and H. Herbelin. The Duality of Computation. In *Proceedings of the 5th ACM SIGPLAN International Conference on Functional Programming (ICFP'00)*, volume 35.9 of *ACM Sigplan Notices*, pages 233–243. ACM, 2000.

[24] H.B. Curry and R. Feys. *Combinatory Logic*, volume 1. North-Holland, Amsterdam, 1958.

[25] G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift*, 39(2):176–210 and 405–431, 1935.

[26] Ph. de Groote. On the Relation between the $\lambda\mu$-Calculus and the Syntactic Theory of Sequential Control. In *Proceedings of 5th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'94)*, volume 822 of *Lecture Notes in Computer Science*, pages 31–43. Springer Verlag, 1994.

[27] C.A. Gunter and D.S. Scott. Semantic domains. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 633–674. North-Holland, 1990.

[28] J.R. Hindley. *Basic Simple Type Theory*. Cambridge University Press, 1997.

[29] K. Kikuchi and T. Sakurai. A translation of intersection and union types for the $\lambda\mu$-calculus. In Jacques Garrigue, editor, *Programming Languages and Systems - 12th Asian Symposium, APLAS 2014, Singapore, November 17-19, 2014, Proceedings*, volume 8858 of *Lecture Notes in Computer Science*, pages 120–139. Springer Verlag, 2014.

[30] S. Lengrand. Call-by-value, call-by-name, and strong normalization for the classical sequent calculus. In B. Gramlich and S. Lucas, editors, *3rd Workshop on Reduction Strategies in Rewriting and Programming (WRS 2003)*, volume 86 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

[31] U. de'Liguoro. The Approximation Theorem for the $\Lambda\mu$-Calculus. *Mathematical Structures in Computer Science*, FirstView:1–21, 2016.

[32] C.-H.L. Ong and C.A. Stewart. A Curry-Howard foundation for functional computation with control. In *Proceedings of the 24th Annual ACM Symposium on Principles Of Programming Languages*, pages 215–227, 1997.

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

[34] G. Pottinger. A Type Assignment for the Strongly Normalizable λ-terms. In J.P. Seldin and J.R. Hindley, editors, *To H. B. Curry, Essays in Combinatory Logic, Lambda-Calculus and Formalism*, pages 561–577. Academic press, New York, 1980.

[35] W. Py. *Confluence en $\lambda\mu$-calcul*. Thèse de doctorat, Université de Savoie, 1998.

[36] S. Ronchi Della Rocca and B. Venneri. Principal type schemes for an extended type theory. *Theoretical Computer Science*, 28:151–169, 1984.

[37] A. Saurin. Standardization and Böhm Trees for $\lambda\mu$-calculus. In M. Blume, N. Kobayashi, and G. Vidal, editors, *Functional and Logic Programming, 10th International Symposium, (FLOPS'10), Sendai, Japan*, volume 6009 of *Lecture Notes in Computer Science*, pages 134–149. Springer Verlag, 2010.

[38] Th. Streicher and B. Reus. Classical logic: Continuation Semantics and Abstract Machines. *Journal of Functional Programming*, 11(6):543–572, 1998.

[39] W. Tait. Intensional Interpretations of Functionals of Finite Type I. *Journal of Symbolic Logic*, 32(2):198–212, 1967.

[40] C.P. Wadsworth. The Relation Between Computational and Denotational Properties for Scott's D_∞-Models of the Lambda-Calculus. *SIAM Journal on Computing*, 5(3):488–521, 1976.