Normalization, Approximation, and Semantics for Combinator Systems*

Steffen van Bakel
Department of Computing,
Imperial College,
180 Queen's Gate, London SW7 2BZ, U.K.
svb@doc.ic.ac.uk

Maribel Fernández
LIENS (CNRS UMR 8548),
École Normale Supérieure,
45 rue d'Ulm, 75005 Paris, France.
maribel@dmi.ens.fr

Technical Report 2000/10, Department of Computing, Imperial College

Abstract

This paper studies normalization of typeable terms and the relation between approximation semantics and filter models for Combinator Systems. It presents notions of approximants for terms, intersection type assignment, and reduction on type derivations; the last will be proved to be strongly normalizable. With this result, it is shown that, for every typeable term, there exists an approximant with the same type, and a characterization of the normalization behaviour of terms using their assignable types is given. Then the two semantics are defined and compared, and it is shown that the approximants semantics is fully abstract but the filter semantics is not.

Introduction

In this paper we will focus on the relation between two approaches for semantics in the framework of Combinator Systems (CS), being the *filter semantics*, obtained by interpreting terms by the set of intersection types that can be assigned to them, and the *approximants semantics*, where terms are interpreted by the set of their approximants, and their interrelation. Approximants are defined as rooted finite sub-trees of the (possibly infinite) normal form, based on the notion of Ω-normal forms of Huet and Lévy [16] (see also [18]).

The relation between the filter semantics and the approximation semantics has been studied extensively in the setting of the Lambda Calculus (LC) [6] (see [8, 7, 1, 3]), where it has been proved that they coincide [19, 3]. But, perhaps surprisingly, this has never been studied for more general notions of rewriting, such as Term Rewriting Systems (TRS) [12, 17].

Within the framework of orthogonal first-order TRS, a term-like model and an appropriate semantics are defined in [21], interpreting terms by the set of their approximants. For these TRS it is also possible to define a semantics where types are interpreted as multi-sorted algebras [12]. Although these types are enough to describe manipulations of objects of an algebraic data-type, they do not provide an account for polymorphism, or higher order functions, which are standard in functional programming languages. A more general and expressive type system, using intersection types, has been developed in [5] for Curryfied Term Rewriting Systems (CuTRS, first-order TRS extended with application). This type system is inspired by the Intersection Type Discipline defined in [8] (see also [7, 1]), an extension of Curry's system [10, 11] in that, essentially, terms are allowed to have more than one type (using the type constructor '$\cap$'). By introducing also the type constant 'ω' a type system for LC is obtained

*Partially supported by NATO Collaborative Research Grant CRG 970285 'Extended Rewriting and Types'.

that is closed under β-equality, and interpreting terms by their assignable types gives a filter lambda model [7, 3].

In this paper, based on the approach of [21], we will define a notion of approximation for CS and show the following *approximation result*: for all terms that can be assigned a type in the intersection system, there exists an approximant that can be assigned the same type. For LC, such an approximation result is relatively easy to obtain, because of the presence of explicit abstraction, but in order to prove these results for abstraction-free calculi, like CS, a new technique had to be developed. This technique is that of defining *reduction on derivations* as a generalization of cut-elimination, that will be proven to be strongly normalizing. This same technique can then also be applied to other formalisms, as done for example in [4] for TRS. Strong normalization of cut-elimination has been studied in the past for several systems, but in the context of intersection types this topic had not yet been tackled.

Using the approximation result, we will show the following normalization properties of typeable terms in the intersection system for CS:

- terms typeable without using ω are strongly normalizable,
- non-Curryfied terms that are typeable with σ from a basis B, such that ω does not occur in B and σ, are normalizable, and
- terms typeable with type $\sigma \neq \omega$ have a head-normal form.

This characterization of the normalization properties of terms using types in the intersection system is well-known in the context of the LC, and it also holds in TRS, provided that the rewrite rules satisfy certain conditions [5]. Perhaps less known is the fact that the notion of approximant can be useful to study the relation between typeability and normalization: in this paper we will show that the approximation result allows for a relatively easy proof of theses properties in CS (a similar result for LC was shown in [3], and an abbreviated proof for TRS appeared in [4]).

Inspired by the approximation result, we will then focus on approximation and filter semantics of CS, as a preparation for future studies of the same semantics in the context of more general rewriting systems, such as TRS. There are several advantages to keeping the computational framework relatively easy at first: confluence comes for free, and a direct relation between CS and LC facilitates definitions and insight. However, note that the normalization properties of LC do not translate directly to CS, since the mappings between LC and Combinatory Logic (a particular CS defined by Curry [9]) do not preserve normal forms or reductions (see Example *1.9*).

Although TRS are very popular in language design and their normalization properties are well-studied, there is still no thorough semantic analysis of TRS. As we have already mentioned, there exists some work in this direction, either supported by types [14] or not [21], but, for example, the relation between these models has not been studied. This paper is a first step towards filling that void, by studying two approaches to semantics for CS, the approximation semantics and the filter semantics, and comparing their expressiveness. We aim to bring these approaches to the context of TRS in future work.

Summarizing, the main contributions of this paper are:

- a strong normalization result for cut-elimination for a system with intersection types,
- a characterization of normalization properties of typeable combinator systems,
- the definition of a filter semantics for CS where terms are interpreted by their assignable types, and an approximation semantics where terms are interpreted by their approximants,
- a proof that these semantics are adequate, and
- a study of the conditions needed to obtain a full-abstraction result.

Outline

In this paper, we will, in Section *1*, define the Combinator Systems, for which we will, in Section *2*, develop a notion of type assignment that uses intersection types; the intersection type assignment system we use in this paper is a variant of the essential type assignment system for ƇTRS [5]. We will show a subject reduction result in Section *3*.

In Sections *4* to *8*, we present the formal construction needed to show that any typeable term in a typeable CS has an approximant of the same type (Theorem *8.2*). In [3], this approximation result has been obtained for LC, by a computability technique [20]. A particular problem to solve in this paper is that the approach of [3] cannot be automatically translated to a technique to use in CS, because of the absence of abstraction in CS. In order to prove the approximation result for CS, we will modify the type system slightly and introduce, in Section *5*, a notion of reduction on type-derivations in this modified system. We will show that derivation reduction is strongly normalizing (Theorem *6.5*), and this property has two direct consequences: the approximation result and a strong normalization theorem for terms that are typeable without using the universal type constant ω (Theorem *8.7*).

The combinatorial equivalent of the characterization of normalisation in LC no longer holds (see Section *8*). However, using the approximation result, we will obtain two normalization properties of typeable combinator systems: a head-normalization theorem (Theorem *8.4*) for typeable terms, and a normalization theorem for the class of typeable non-Curryfied terms, as defined in Definition *1.2* (Theorem *8.5*).

Section *9* presents the definition of a filter semantics for CS, where terms are interpreted by their assignable types, and an approximation semantics, where terms are interpreted by their approximants. The approximation semantics gives a fully abstract model for CS, whereas the filter semantics gives a semi-model only, except for special cases.

The paper finishes in Section *10*, which contains the conclusions.

1 Combinator Systems

In this section, Combinator Systems (CS) will be presented as a special kind of applicative TRS [17] where formal parameters of function symbols are not allowed to have structure, and right-hand sides of term rewriting rules are constructed of term-variables only. We have chosen to use this kind of presentation rather than the one normally used (see, for example, [17] or [6]), in view of a future extension of the results of this paper to full TRS, in the spirit of [5].

Definition 1.1 (*Combinator terms*) *i)* An *alphabet* or *signature* $\Sigma = (\mathcal{C}, \mathcal{X})$ consists of a countable infinite set $\mathcal{X}$ of variables ranged over by $x, y, z, \ldots$, a non-empty set $\mathcal{C}$ of *combinators*, ranged over by $C, D, E, \ldots$, each equipped with an arity greater than 0, and the binary function symbol *Ap* (application).

ii) The set $T(\mathcal{C}, \mathcal{X})$ of *terms*, ranged over by t, is defined by:

$$t ::= x \mid C \mid \mathit{Ap}\,(t_1, t_2)$$

As usual, since '*Ap*' is the only function symbol, we will write $(t_1\, t_2)$ instead of $\mathit{Ap}\,(t_1, t_2)$, and outermost brackets will be omitted.

In Section *8*, a normalization result is proved for terms where all subterms of the form $C\, t_1 \cdots t_n$ are such that $n \geq \mathit{arity}\,(C)$. These terms are called 'Non-Curryfied'.

Definition 1.2 (*Non-Curryfied terms*) The set $T_{NC}(\mathcal{C},\mathcal{X})$ of *non-Curryfied terms* is defined by:

$$t ::= x t_1 \cdots t_n \ (n \geq 0) \mid C t_1 \cdots t_n \ (\mathit{arity}(C) \leq n)$$

Notice that $T_{NC}(\mathcal{C},\mathcal{X})$ is a subset of $T(\mathcal{C},\mathcal{X})$.

Definition 1.3 (*Term-substitutions*) A *term-substitution* R is a map from $T(\mathcal{C},\mathcal{X})$ to $T(\mathcal{C},\mathcal{X})$, determined by its restriction to a finite set of variables, satisfying $\mathrm{R}\,(t_1\,t_2) = \mathrm{R}\,(t_1)\,\mathrm{R}\,(t_2)$. We write t^{R} instead of $\mathrm{R}\,(t)$. If R maps x_i to u_i, for $1 \leq i \leq n$, we also write $\{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$ for R, and write $t^{\overline{u}}$ for t^{R}.

Combinator Systems, and the notion of rewriting on combinator terms, are defined by the following:

Definition 1.4 (*Combinator Systems*) *i)* A *combinator rule* on $\Sigma = (\mathcal{C},\mathcal{X})$ is a pair (l,r) of terms in $T(\mathcal{C},\mathcal{X})$, such that:

- *a)* There are C and distinct $x_1,\ldots,x_n$, such that $l = C\,x_1\cdots x_n$, where $n = \mathit{arity}(C)$.
- *b)* The variables occurring in r are contained in l, and r contains no symbols from $\mathcal{C}$.

ii) A *Combinator System* (CS) is a pair $(\Sigma,\mathbf{R})$ of an alphabet Σ and a set $\mathbf{R}$ of combinator rules on $\Sigma = (\mathcal{C},\mathcal{X})$, such that there is *exactly one* rule in $\mathbf{R}$ for each combinator $C \in \mathcal{C}$. This rule (l,r) is called *the combinator rule for* C; we will use the symbol C also as name for this rule and write $l \to_C r$.

iii) A combinator rule $l \to_C r$ determines a set of *reductions* $l^{\mathrm{R}} \to_C r^{\mathrm{R}}$ for all term-substitutions R. The left hand side l^{R} is called a *redex*; it may be replaced by its '*contractum*' r^{R} inside any context C[ ]; this gives rise to *reduction steps*: $\mathrm{C}[\,l^{\mathrm{R}}\,] \to_C \mathrm{C}[\,r^{\mathrm{R}}\,]$.

iv) We write $t \to_{\mathbf{R}} t'$ if there is a rule $l \to_C r$ in $\mathbf{R}$ such that $t \to_C t'$, and call $\to_{\mathbf{R}}$ the *one-step rewrite relation* generated by $\mathbf{R}$, and $\to_{\mathbf{R}}^{+}$ (respectively $\to_{\mathbf{R}}^{*}$) the transitive (respectively reflexive and transitive) closure of $\to_{\mathbf{R}}$ (the index $\mathbf{R}$ will be omitted when it is clear from the context). If $t_0 \to^{+} t_n$, then t_n is a *reduct* of t_0.

Example 1.5 (Combinatory Logic) The standard example of a CS is Combinatory Logic (CL) – defined by Curry independently of LC [9] – that is, in our notation, formulated as follows:

$$\begin{array}{lcl} \mathbf{S}\,xyz & \to & xz(yz), \\ \mathbf{K}\,xy & \to & x, \\ \mathbf{I}\,x & \to & x. \end{array}$$

(The last rule was not part of the original definition, but is nowadays normally added.)

We will assume that no two combinators have the same interpretation in LC (see Definition *1.7*), so a CS like

$$\begin{array}{lcl} \mathbf{I}x & \to & x \\ \mathbf{J}x & \to & x \end{array}$$

is excluded, since it would give an immediate counter example against any full-abstraction result with respect to the filter semantics (see Section *9*).

This notion of reduction on combinator terms as in Definition *1.4* is also known as *weak reduction* and satisfies the Church-Rosser Property (see [6]).

Property 1.6 (Church-Rosser) *Let* $(\Sigma,\mathbf{R})$ *be a* CS. *If* $t \to^{*} u$ *and* $t \to^{*} v$*, then there exists a* w *such that* $u \to^{*} w$ *and* $v \to^{*} w$. ■

We now focus on the relation between reduction in CS and in LC.

Definition 1.7 $\langle\ \rangle_\lambda : T(\mathcal{C},\mathcal{X}) \to \Lambda$, the interpretation of combinator terms over $\mathcal{C}$ in LC, is defined by:

$$\begin{array}{rcll} \langle x\rangle_\lambda &=& x, & \text{for all } x \in \mathcal{X}, \\ \langle t_1 t_2\rangle_\lambda &=& \langle t_1\rangle_\lambda \langle t_2\rangle_\lambda, & \\ \langle C\rangle_\lambda &=& \lambda x_1 \cdots x_n.\langle r\rangle_\lambda, & \text{where } C x_1 \cdots x_n \to r \text{ is the rule for } C. \end{array}$$

Notice that, since we assume the set of term variables for CS and LC to be the same, as well as the two notions of term-application, $\langle r\rangle_\lambda = r$ for every r that is the right-hand side of a combinator rule.

Proposition 1.8 If $t \to^* t'$, then $\langle t\rangle_\lambda \twoheadrightarrow_\beta \langle t'\rangle_\lambda$.
Proof: By induction on the definition of $\to^*$. We only consider the case of $(C x_1 \cdots x_n)^{\mathrm{R}} \to_C r^{\mathrm{R}}$, where $\mathrm{R} = \{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$. Let $\mathrm{R}' = \{x_1 \mapsto \langle u_1\rangle_\lambda, \ldots, x_n \mapsto \langle u_n\rangle_\lambda\}$. Then

$$\begin{array}{rcl} \langle (C x_1 \cdots x_n)^{\mathrm{R}}\rangle_\lambda &=& \langle C u_1 \cdots u_n\rangle_\lambda \\ &=& \langle C\rangle_\lambda \langle u_1\rangle_\lambda \cdots \langle u_n\rangle_\lambda \\ &=& (\lambda x_1 \cdots x_n.r)\langle u_1\rangle_\lambda \cdots \langle u_n\rangle_\lambda \\ &\twoheadrightarrow_\beta& r^{\mathrm{R}'} \\ &=& \langle r^{\mathrm{R}}\rangle_\lambda. \end{array}$$

The proof is completed by induction on the number of steps in $\to^*$. ∎

Although this interpretation in LC of a CS, $\langle\ \rangle_\lambda$, respects reduction, in general, the length of the reduction sequence increases significantly. Only for particular CS it is also possible to define an interpretation of LC, $[\![\ ]\!]_{\mathcal{C}}$; the standard example is that of CL (for details see [11, 6, 13]). One important property of these two translations is that

$$\langle [\![M]\!]_{\mathrm{CL}}\rangle_\lambda \twoheadrightarrow_\beta M,$$

for all $M \in \Lambda$. There exists no converse of this property; moreover, the mapping $\langle\ \rangle_\lambda$ does not preserve normal forms or reductions:

Example 1.9 ([6]) *i)* $\mathbf{SK}$ is a normal form, but $\langle \mathbf{SK}\rangle_\lambda \twoheadrightarrow_\beta \lambda xy.y$,

ii) $t = \mathbf{S}(\mathbf{K}(\mathbf{SII}))(\mathbf{K}(\mathbf{SII}))$ is a normal form, but $\langle t\rangle_\lambda \twoheadrightarrow_\beta \lambda c.(\lambda x.xx)(\lambda x.xx)$, which does not have a β-normal form,

iii) $t = \mathbf{SK}((\mathbf{SII})(\mathbf{SII}))$ has no normal form, while $\langle t\rangle_\lambda \twoheadrightarrow_\beta \lambda x.x$.

For these reasons, normalization results of LC do not transfer easily to CS. Therefore, in this paper, we will study the normalization properties of CS directly in the CS framework.

We now define (head-)normal forms, (head-)normalizability, strongly normalizability, unsolvable and neutral terms.

Definition 1.10 (*(Head-)normal forms)* Let $(\Sigma, \mathbf{R})$ be a CS.

i) A term is in *normal form* with respect to **R** if it is irreducible.

ii) A term t is in *head-normal form* with respect to **R** if either
  - *a)* there are a variable x and terms $t_1, \ldots, t_n$ $(n \geq 0)$ such that $t \equiv x t_1 \cdots t_n$, or
  - *b)* there are a combinator C and terms $t_1, \ldots, t_n$ such that $t \equiv C t_1 \cdots t_n$, and $n < \mathit{arity}(C)$.

iii) A term is *(head-)normalizable* if it can be reduced to a term in (head-)normal form. A rewrite system is *strongly normalizing* (or terminating) if all the rewrite sequences are finite; it is *(head-)normalizing* if every term is (head-)normalizable.

iv) A term is called *unsolvable* if it has no head-normal form.

v) A term t is *neutral* if there are a variable x and terms $t_1, \ldots, t_n$ $(n \geq 0)$, such that $t \equiv x t_1 \cdots t_n$.

2 Intersection type assignment

It is well-known that in the study of normalization of reduction systems, the notion of types plays an important role, and that many of the now existing type assignment systems for Functional Programming Languages (FPL) are based on (extensions of) the Curry type assignment system for LC [10, 11]. The Intersection Type Discipline (ITD) as presented in [8] (see also [7, 1]) is an extension of Curry's system, in that, essentially, terms are allowed to have more than one type (using the type constructor '$\cap$'). By introducing also the type constant 'ω' a system is obtained that is closed under β-equality, and interpreting terms by their assignable types gives a filter lambda model [7, 3].

In this section, we will develop a notion of type assignment on CS that uses intersection types. It is inspired by similar definitions presented in, for example, [13] and [5]. The extension with respect to [13] is that in that paper only combinatory complete CS are considered. The change made with respect to [5] is that CS are considered instead of arbitrary TRS.

As done in [13], we will assume that, for every combinator C, there is a basic type from which all types needed for an occurrence C in a term can be obtained. Other than in that paper, however, we will not limit ourselves to basic types that are the principal type of the corresponding lambda term (see [19, 2]).

As in [5], we will use strict intersection types (see [1]), which have the same expressive power as the general intersection types defined in [7] and used in [13]. Strict types are the representatives for equivalence classes of the types considered in the system of [7]. In the set of strict types, intersection type schemes and the type constant ω play a limited role: they only occur as subtypes at the left hand side of an arrow type scheme.

Definition 2.1 (*Strict intersection types*) *i)* Let Φ be a countable infinite set of type-variables, ranged over by φ. $\mathcal{T}_s$, the set of *strict types*, ranged over by $\sigma, \tau, \ldots$, is defined by:

$$\sigma ::= \varphi \mid ((\sigma_1 \cap \cdots \cap \sigma_n) \rightarrow \sigma),\ (n \geq 0)$$

The set $\mathcal{T}$ of *strict intersection types* is defined by:

$$\mathcal{T} = \{(\sigma_1 \cap \cdots \cap \sigma_n) \mid n \geq 0\ \&\ \forall 1 \leq i \leq n\, [\sigma_i \in \mathcal{T}_s]\}$$

We will use the convention that ω is the same as an intersection of zero strict types: if $n = 0$, then $\sigma_1 \cap \cdots \cap \sigma_n \equiv \omega$, so ω does not occur in an intersection subtype. As usual in the notation of types, right-most, outermost brackets will be omitted, and, as in logic, $\cap$ binds stronger than $\rightarrow$.

ii) On $\mathcal{T}$, the relation $\leq$ is defined as the smallest preorder satisfying:

$$\begin{array}{rl} \forall 1 \leq i \leq n\, [\sigma_1 \cap \cdots \cap \sigma_n \ \leq\ \sigma_i] & (n \geq 1) \\ \forall 1 \leq i \leq n\, [\sigma \leq \sigma_i] \ \Rightarrow\ \sigma \leq \sigma_1 \cap \cdots \cap \sigma_n & (n \geq 0) \\ \rho \leq \sigma\ \&\ \tau \leq \mu \ \Rightarrow\ \sigma \rightarrow \tau \leq \rho \rightarrow \mu & \end{array}$$

iii) We define the relation $\sim$ by: $\sigma \sim \tau \iff \sigma \leq \tau \leq \sigma$.

We will work with types modulo $\sim$.

Lemma 2.2 ([3]) *For all $\sigma, \tau \in \mathcal{T}$, $\sigma \leq \tau$ if and only if there are $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_m$ such that $\sigma = \sigma_1 \cap \cdots \cap \sigma_n$, $\tau = \tau_1 \cap \cdots \cap \tau_m$, and, for every $1 \leq j \leq m$, there is a $1 \leq i \leq n$ such that $\sigma_i \leq \tau_j$.* ∎

Notice that, by definition, in $\sigma_1 \cap \cdots \cap \sigma_n$, all $\sigma_1, \ldots, \sigma_n$ are strict; sometimes we will deviate from this by writing $\sigma \cap \tau$ also for σ, τ not in $\mathcal{T}_s$.

Definition 2.3 (*Bases*) *i)* A *statement* is an expression of the form $t{:}\sigma$, where t is the *subject* and σ is the *predicate.*

ii) A *basis* B is a set of statements with (distinct) variables as subjects, and, if $x{:}\sigma \in B$, then $\sigma \neq \omega$.

iii) If $B_1, \ldots, B_n$ are bases, then $\Pi\{B_1, \ldots, B_n\}$ is the basis defined as follows:
$$x{:}\sigma_1 \cap \cdots \cap \sigma_m \in \Pi\{B_1, \ldots, B_n\}$$
if and only if $m \geq 1$ and $\{x{:}\sigma_1, \ldots, x{:}\sigma_m\}$ is the set of all statements that have x as subject that occur in $B_1 \cup \cdots \cup B_n$.

iv) The relations $\leq$ and $\sim$ are extended to bases by:
$$\begin{array}{l} B \leq B' \iff \forall x{:}\sigma' \in B'\, \exists x{:}\sigma \in B\, [\sigma \leq \sigma'] \\ B \sim B' \iff B \leq B' \leq B. \end{array}$$

We will often write $B, x{:}\sigma$ (or $B \cup \{x{:}\sigma\}$) for the basis $\Pi\{B, \{x{:}\sigma\}\}$, when x does not occur in B. Notice that, in part *(iii)*, if $n = 0$, then $\Pi\{B_1, \ldots, B_n\} = \emptyset$, and that $B \leq \emptyset$, for all B.

Our type assignment system will derive judgements of the form $B \vdash_{\mathcal{E}} t{:}\sigma$, where B is a basis and σ a type. A triple $\langle B, \sigma, E \rangle$ will be used as a representation of the type derivation, E being the set of types used for the combinators appearing in t.

We will now recall three operations on types and triples that are needed in the definition of type assignment and are standard in intersection systems. Substitution is the operation that instantiates a type (i.e. that replaces type-variables by types). The operation of expansion replaces types by the intersection of a number of copies of that type. The operation of lifting replaces basis and type by a smaller basis and a larger type, in the sense of $\leq$.

These three operations are of use in Definition *2.13*, when we want to specify how, for a specific combinator, a type required by the context can be obtained from the type provided for that combinator by the environment (Definition *2.12*). It is possible to define type assignment with fewer of less powerful operations on types, but in order to obtain enough expressive power to prove Theorem *2.18 (i)*, all three operations are needed.

Definition 2.4 (*Type-substitution*) *i)* The *type-substitution* $(\varphi \mapsto \alpha) : \mathcal{T} \rightarrow \mathcal{T}$, where $\varphi \in \Phi$ and $\alpha \in \mathcal{T}_s \cup \{\omega\}$, is defined by:
$$\begin{array}{lcll}
(\varphi \mapsto \alpha)\,(\varphi) & = & \alpha & \\
(\varphi \mapsto \alpha)\,(\varphi') & = & \varphi', & \text{if } \varphi' \neq \varphi \\
(\varphi \mapsto \alpha)\,(\sigma \rightarrow \tau) & = & \omega, & \text{if } (\varphi \mapsto \alpha)\,(\tau) = \omega \\
(\varphi \mapsto \alpha)\,(\sigma \rightarrow \tau) & = & (\varphi \mapsto \alpha)\,(\sigma) \rightarrow (\varphi \mapsto \alpha)\,(\tau), & \text{if } (\varphi \mapsto \alpha)\,(\tau) \neq \omega \\
(\varphi \mapsto \alpha)\,(\sigma_1 \cap \cdots \cap \sigma_n) & = & (\varphi \mapsto \alpha)\,(\sigma'_1) \cap \cdots \cap (\varphi \mapsto \alpha)\,(\sigma'_m), & \text{where} \\
& & \{\sigma'_1, \ldots, \sigma'_m\} = \{\sigma_i \in \{\sigma_1, \ldots, \sigma_n\} \mid (\varphi \mapsto \alpha)\,(\sigma_i) \neq \omega\} &
\end{array}$$

ii) If S_1 and S_2 are type-substitutions, then so is $S_1 \circ S_2$, where $S_1 \circ S_2(\sigma) = S_1(S_2(\sigma))$.

iii) $S(B) = \{x{:}S(\alpha) \mid x{:}\alpha \in B \;\&\; S(\alpha) \neq \omega\}$.

iv) $S(\langle B, \sigma, E \rangle) = \langle S(B), S(\sigma), \{S(\rho) \mid \rho \in E\} \rangle$.

For type-substitutions, the following properties hold:

Lemma 2.5 ([2]) *Let S be a type-substitution. If* $\sigma \leq \tau$, *then* $S(\sigma) \leq S(\tau)$ *, and if* $B \leq B'$, *then* $S(B) \leq S(B')$. ∎

Our operation of expansion is similar to the one defined in [19] for the full intersection system, we just need to make some minor changes to make sure that the type obtained is always in $\mathcal{T}$. For this, we have to check the last type-variable in arrow types (for a detailed discussion of the complexity of this operation, see [2]).

Definition 2.6 The *last type-variable* of a strict type, *last* (σ), is defined by:

$$\begin{aligned} \mathit{last}\,(\varphi) &= \varphi, \\ \mathit{last}\,(\sigma\to\tau) &= \mathit{last}\,(\tau). \end{aligned}$$

Definition 2.7 (*Expansion*) An expansion *Ex* is defined by a pair $\langle\mu,n\rangle$ where $\mu\in\mathcal{T}$ and $n\geq 2$. In order to expand a type-derivation $\langle B,\sigma,E\rangle$ we will expand each type occurring in it, for which we first need to compute the set of affected variables.

(*Affected variables*): The set $\mathcal{V}_\mu(\langle B,\sigma,E\rangle)$ of type-variables is defined by:

a) If φ occurs in μ, then $\varphi\in\mathcal{V}_\mu(\langle B,\sigma,E\rangle)$.

b) If *last* $(\tau)\in\mathcal{V}_\mu(\langle B,\sigma,E\rangle)$, with $\tau\in\mathcal{T}_s$ and τ (a subtype) in $\langle B,\sigma,E\rangle$, then for all type-variables φ that occur in τ: $\varphi\in\mathcal{V}_\mu(\langle B,\sigma,E\rangle)$.

(*Renamings*): Let $\mathcal{V}_\mu(\langle B,\sigma,E\rangle)=\{\varphi_1,\ldots,\varphi_m\}$. Choose $m\times n$ different type-variables $\varphi_1^1,\ldots,\varphi_n^1,\ldots,\varphi_1^m,\ldots,\varphi_n^m$, such that each φ_i^j does not occur in $\langle B,\sigma,E\rangle$, for $1\leq i\leq n$ and $1\leq j\leq m$. Let S_i be such that $S_i(\varphi_j)=\varphi_i^j$.

(*Expansion of a type in the derivation* $\langle B,\sigma,E\rangle$): *Ex* (τ) is inductively defined as follows:

$$\begin{array}{lll} \mathit{Ex}\,(\tau_1\cap\cdots\cap\tau_n) & = \mathit{Ex}\,(\tau_1)\cap\cdots\cap\mathit{Ex}\,(\tau_n) & \\ \mathit{Ex}\,(\tau) & = S_1(\tau)\cap\cdots\cap S_n(\tau), & \text{if } \mathit{last}\,(\tau)\in\mathcal{V}_\mu(\langle B,\sigma,E\rangle). \\ \mathit{Ex}\,(\varphi) & = \varphi, & \text{if } \varphi\notin\mathcal{V}_\mu(\langle B,\sigma,E\rangle). \\ \mathit{Ex}\,(\sigma\to\tau) & = \mathit{Ex}\,(\sigma)\to\mathit{Ex}\,(\tau), & \text{if } \mathit{last}\,(\tau)\notin\mathcal{V}_\mu(\langle B,\sigma,E\rangle). \end{array}$$

(*Expansion of the basis* B): *Ex* $(B)=\{x{:}\mathit{Ex}\,(\rho)\mid x{:}\rho\in B\}$.

(*Expansion of the type-derivation* $\langle B,\sigma,E\rangle$): *Ex* $(\langle B,\sigma,E\rangle)=\langle \mathit{Ex}\,(B),\mathit{Ex}\,(\sigma),\{\mathit{Ex}\,(\rho)\mid\rho\in E\}\rangle$.

An expansion operation *Ex* can also be applied to a type σ outside the context of a type-derivation. In that case, we define *Ex* $(\sigma)=\sigma'$ such that *Ex* $(\langle\emptyset,\sigma,\emptyset\rangle)=\langle\emptyset,\sigma',\emptyset\rangle$.

The operation of expansion is in fact an extension of that of [2] and [19], in that the set E of types used for combinators is considered when computing the effect of an expansion on a type-derivation. The proofs of the following properties are similar to those in [2].

Lemma 2.8 Let Ex be the expansion defined by $\langle\mu,n\rangle$.

i) a) For $1\leq i\leq n$, *there are* ρ_i *and* S_i *such that* $S_i(\rho)=\rho_i$ *and* $\mathit{Ex}(\rho)=\rho_1\cap\cdots\cap\rho_n$, *or*

b) $\mathit{Ex}(\rho)\in\mathcal{T}_s$.

ii) a) For $1\leq i\leq n$, *there are* B_i,σ_i, *and* S_i *such that* $S_i(\langle B,\sigma\rangle)=\langle B_i,\sigma_i\rangle$, *and* $\mathit{Ex}(\langle B,\sigma,E\rangle)=\langle\Pi\{B_1,\ldots,B_n\},\sigma_1\cap\cdots\cap\sigma_n,E'\rangle$, *or*

b) $\mathit{Ex}(\langle B,\sigma,E\rangle)=\langle B',\sigma',E'\rangle$, *with* $\sigma'\in\mathcal{T}_s$.

Lemma 2.9 Let Ex be the expansion defined by $\langle\mu,n\rangle$ *with respect to* $\langle B,\sigma,E\rangle$.

i) If ρ *appears in* B,σ, *or* E, *and* $\rho\leq\tau$, *then* $\mathit{Ex}(\rho)\leq\mathit{Ex}(\tau)$.

ii) If $B\leq B'$, *then* $\mathit{Ex}(B)\leq\mathit{Ex}(B')$. ■

Definition 2.10 (*Lifting*) A *lifting L* is an operation denoted by a pair of pairs $\langle\langle B_0,\tau_0\rangle,\langle B_1,\tau_1\rangle\rangle$ such that $\tau_0\leq\tau_1$ and $B_1\leq B_0$, and is defined by:

$$\begin{array}{llll} L\,(\sigma) = \tau_1, & \text{if } \sigma=\tau_0 & L\,(B) = B_1, & \text{if } B=B_0 \\ L\,(\sigma) = \sigma, & \text{otherwise} & L\,(B) = B, & \text{otherwise} \end{array}$$

$$L(\langle B,\sigma,E\rangle) = \langle L(B),L(\sigma),\{L(\rho)\mid\rho\in E\}\rangle\,.$$

Definition 2.11 (*Chains*) A *chain* is an object $[O_1,\ldots,O_n]$, where each O_i is an operation of type-substitution, expansion or lifting, and

$$\begin{array}{lcl}[O_1,\ldots,O_n]\,(\sigma) & = & O_n\,(\cdots(O_1\,(\sigma))\cdots),\\ [O_1,\ldots,O_n]\,(\langle B,\sigma,E\rangle) & = & O_n\,(\cdots(O_1\,(\langle B,\sigma,E\rangle))\cdots).\end{array}$$

We will use $*$ to denote the operation of concatenation of chains, and *Ch* to denote a chain.

To complete the definition of the type assignment system, we present now the type assignment rules that are used to assign types in $\mathcal{T}$ to terms and combinator rules. In order to type the combinators, we use an environment that provides a type in $\mathcal{T}_s$ for every $C\in\mathcal{C}$, and use chains of operations to obtain the type for an occurrence of the combinator from the type provided for it by the environment.

Definition 2.12 (*Environment*) Let $(\Sigma,\mathbf{R})$ be a CS, with $\Sigma=(\mathcal{C},\mathcal{X})$.

i) An *environment* for $(\Sigma,\mathbf{R})$ is a mapping $\mathcal{E}:\mathcal{C}\rightarrow\mathcal{T}_s$.

ii) For $C\in\mathcal{C}$, $\tau\in\mathcal{T}_s$, and $\mathcal{E}$ an environment, the environment $\mathcal{E}\,[C:=\tau]$ is defined by:

$$\begin{array}{lcll}\mathcal{E}\,[C:=\tau]\,(D) & = & \tau, & \text{if } D=C,\\ \mathcal{E}\,[C:=\tau]\,(D) & = & \mathcal{E}(D), & \text{otherwise.}\end{array}$$

Since an environment $\mathcal{E}$ maps all $C\in\mathcal{C}$ to types in $\mathcal{T}_s$, no combinator is mapped to ω.

We define now type assignment on terms and combinator rules.

Definition 2.13 (*Type assignment*) Let $(\Sigma,\mathbf{R})$ be a CS and $\mathcal{E}$ an environment for $(\Sigma,\mathbf{R})$.

i) *Type assignment* and *derivations* are defined by the following natural deduction system (where all types displayed are in $\mathcal{T}_s$, except for τ in rules $(\leq)$ and $(\rightarrow E)$):

$$(\mathcal{E}):\ \frac{\exists Ch\,[Ch(\mathcal{E}(C))=\tau]}{B\vdash_{\mathcal{E}} C{:}\tau}\qquad(\rightarrow E):\ \frac{B\vdash_{\mathcal{E}} t_1{:}\tau\rightarrow\sigma\quad B\vdash_{\mathcal{E}} t_2{:}\tau}{B\vdash_{\mathcal{E}} t_1t_2{:}\sigma}$$

$$(\leq):\ \frac{x{:}\tau\in B\quad \tau\leq\sigma}{B\vdash_{\mathcal{E}} x{:}\sigma}\qquad(\cap I):\ \frac{B\vdash_{\mathcal{E}} t{:}\sigma_1\quad\ldots\quad B\vdash_{\mathcal{E}} t{:}\sigma_n}{B\vdash_{\mathcal{E}} t{:}\sigma_1\cap\cdots\cap\sigma_n}\ (n\geq 0)$$

If $B\vdash_{\mathcal{E}} t{:}\sigma$ is derivable using a derivation D, we write D :: $B\vdash_{\mathcal{E}} t{:}\sigma$, and if E is the set of types used for the combinators in this derivation, we represent it by $\langle B,\sigma,E\rangle$. We write $B\vdash_{\mathcal{E}} t{:}\sigma$ to express that there exists a derivation D such that D :: $B\vdash_{\mathcal{E}} t{:}\sigma$. We write $B\vdash^{\not\omega}_{\mathcal{E}} t{:}\sigma$ if ω is not used in the derivation.

ii) Let $C\in\mathcal{C}$, *arity*$(C)=n$. The combinator rule $Cx_1\cdots x_n\rightarrow r\in\mathbf{R}$ *is typeable with respect to* $\mathcal{E}$, if there are $\sigma_1,\ldots,\sigma_n\in\mathcal{T}$ and $\sigma\in\mathcal{T}_s$, such that $\{x_1{:}\sigma_1,\ldots,x_n{:}\sigma_n\}\vdash_{\mathcal{E}} r{:}\sigma$, and $\mathcal{E}(C)=\sigma_1\rightarrow\cdots\rightarrow\sigma_n\rightarrow\sigma$.

iii) $(\Sigma,\mathbf{R})$ *is typeable with respect to* $\mathcal{E}$, if every rule in $\mathbf{R}$ is typeable with respect to $\mathcal{E}$.

Notice that if $B\vdash_{\mathcal{E}} t{:}\sigma$, then B can contain more statements than needed to obtain $t{:}\sigma$. Moreover, by part *(ii)* of this definition, also $\{x_1{:}\sigma_1,\ldots,x_n{:}\sigma_n\}\vdash_{\mathcal{E}} Cx_1\cdots x_n{:}\sigma$. However, just stating

> 'The combinator rule $l\rightarrow r$ is typeable with respect to the environment $\mathcal{E}$, if and only if there exist basis B and type σ, such that $B\vdash_{\mathcal{E}} l{:}\sigma$ and $B\vdash_{\mathcal{E}} r{:}\sigma$.'

would give a notion of type assignment that is not comparable to intersection type assignment for LC. For an example, take the combinator rule $Exy\rightarrow xy$. Let $\mathcal{E}(E)=\varphi_1\rightarrow\varphi_2\rightarrow\varphi_3$. Take $B=\{x{:}\varphi_1\cap(\varphi_2\rightarrow\varphi_3),y{:}\varphi_2\}$, then both $B\vdash_{\mathcal{E}} Exy{:}\varphi_3$ and $B\vdash_{\mathcal{E}} xy{:}\varphi_3$ are easy to derive. Notice that this combinator rule for E corresponds to the lambda term $\lambda xy.xy$, but $\varphi_1{\rightarrow}\varphi_2{\rightarrow}\varphi_3$ is not a correct type for this term.

Example 2.14 The rules of CL are typeable with respect to the environment $\mathcal{E}_{\text{CL}}$:

$$\begin{array}{lcl} \mathcal{E}_{\text{CL}}(\mathbf{S}) &=& (\varphi_1{\rightarrow}\varphi_2{\rightarrow}\varphi_3){\rightarrow}(\varphi_4{\rightarrow}\varphi_2){\rightarrow}\varphi_1{\cap}\varphi_4{\rightarrow}\varphi_3, \\ \mathcal{E}_{\text{CL}}(\mathbf{K}) &=& \varphi_5{\rightarrow}\omega{\rightarrow}\varphi_5, \\ \mathcal{E}_{\text{CL}}(\mathbf{I}) &=& \varphi_6{\rightarrow}\varphi_6. \end{array}$$

The term **SKSI** can be typed with the type $\alpha{\rightarrow}\alpha$ $(=\beta)$ with respect to $\mathcal{E}_{\text{CL}}$: take

$$\begin{array}{l} Ch_1 = [(\varphi_1 \mapsto \alpha{\rightarrow}\alpha), (\varphi_2 \mapsto \omega), (\varphi_3 \mapsto \alpha{\rightarrow}\alpha), (\varphi_4 \mapsto \omega)], \\ Ch_2 = [(\varphi_5 \mapsto \alpha{\rightarrow}\alpha)], \\ Ch_3 = [(\varphi_6 \mapsto \alpha)], \end{array}$$

then

$$\dfrac{\dfrac{\dfrac{\dfrac{Ch_1(\mathcal{E}_{\text{CL}}(\mathbf{S})) = (\beta{\rightarrow}\omega{\rightarrow}\beta){\rightarrow}\omega{\rightarrow}\beta{\rightarrow}\beta}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{S}{:}(\beta{\rightarrow}\omega{\rightarrow}\beta){\rightarrow}\omega{\rightarrow}\beta{\rightarrow}\beta} \quad \dfrac{Ch_2(\mathcal{E}_{\text{CL}}(\mathbf{K})) = \beta{\rightarrow}\omega{\rightarrow}\beta}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{K}{:}\beta{\rightarrow}\omega{\rightarrow}\beta}}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{SK}{:}\omega{\rightarrow}\beta{\rightarrow}\beta} \quad \dfrac{}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{S}{:}\omega}}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{SKS}{:}\beta} \quad \dfrac{Ch_3(\mathcal{E}_{\text{CL}}(\mathbf{I})) = \beta}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{I}{:}\beta}}{\emptyset \vdash_{\mathcal{E}_{\text{CL}}} \mathbf{SKSI}{:}\beta}$$

The definition of type assignment on CS as presented in this paper allows for the formulation of a precise relation between types assignable to terms, and those assignable to equivalent lambda terms. In fact, a result similar to part of the following property has already been proved in [13].

Definition 2.15 Let $\vdash_{\lambda\cap}$ stand for the notion of intersection type assignment on LC, as defined in [3] by the following derivation rules (where all types displayed are in $\mathcal{T}_s$, except for τ in rules $({\rightarrow}I)$, $({\rightarrow}E)$ and $(\leq)$):

$$({\rightarrow}I){:}\quad \dfrac{B, x{:}\tau \vdash_{\lambda\cap} M{:}\sigma}{B \vdash_{\lambda\cap} \lambda x.M{:}\tau{\rightarrow}\sigma} \qquad\qquad ({\rightarrow}E){:}\quad \dfrac{B \vdash_{\lambda\cap} M{:}\tau{\rightarrow}\sigma \quad B \vdash_{\lambda\cap} N{:}\tau}{B \vdash_{\lambda\cap} MN{:}\sigma}$$

$$(\leq){:}\quad \dfrac{x{:}\tau \in B \quad \tau \leq \sigma}{B \vdash_{\lambda\cap} x{:}\sigma}\ (\sigma \in \mathcal{T}_s) \qquad (\cap I){:}\quad \dfrac{B \vdash_{\lambda\cap} M{:}\sigma_1 \quad \ldots \quad B \vdash_{\lambda\cap} M{:}\sigma_n}{B \vdash_{\lambda\cap} M{:}\sigma_1{\cap}\cdots{\cap}\sigma_n}\ (n \geq 0)$$

Let $[\![\ ]\!]_{\text{CL}} : \Lambda \rightarrow T(\{\mathbf{S},\mathbf{K},\mathbf{I}\},\mathcal{X})$, be the interpretation of lambda terms in CL (for details, see [11, 6, 13]), then the following states the relation between type assignment in CS and in LC.

Property 2.16 If $B \vdash_{\lambda\cap} M{:}\sigma$, then $B \vdash_{\mathcal{E}_{\text{CL}}} [\![M]\!]_{\text{CL}}{:}\sigma$.
Proof: Similar to Theorem 3.7 of [13]. ■

A more general formulation of Property *2.16*, of course, only holds for CS that are expressive enough to encode LC. However, even for those the property is only provable if the environment used assigns those types to the combinator symbols that are the principal types [19, 2] of the corresponding lambda terms. For example, take $\vdash_{\lambda\cap} \lambda x.x{:}\alpha{\rightarrow}\alpha$ and notice that $[\![\lambda x.x]\!]_{\text{CL}} = \mathbf{I}$. If $\mathcal{E}\,(\mathbf{I}) = (\alpha{\rightarrow}\alpha){\rightarrow}\alpha{\rightarrow}\alpha$, then it is not possible to assign $\alpha{\rightarrow}\alpha$ to $\mathbf{I}$ in $\vdash_{\mathcal{E}}$ (see also Section *9*).

However, we can show the following two results for CS equipped with principal environments.

Definition 2.17 The environment $\mathcal{E}$ is called *principal for* $\mathcal{C}$, if for all $C \in \mathcal{C}$, $\mathcal{E}\,(C)$ is the principal type for $\langle C\rangle_\lambda$ in $\vdash_{\lambda\cap}$.[1]

Theorem 2.18 *Let* $((\mathcal{C},\mathcal{X}),\mathbf{R})$ *be a* CS.

i) If $\mathcal{E}$ *is principal for* $\mathcal{C}$*, then* $B \vdash_{\lambda\cap} \langle t\rangle_\lambda{:}\sigma$ *implies* $B \vdash_{\mathcal{E}} t{:}\sigma$.

ii) $B \vdash_{\mathcal{E}} t{:}\sigma$ *implies* $B \vdash_{\lambda\cap} \langle t\rangle_\lambda{:}\sigma$.

Proof: *i)* By induction on the structure of terms in $T(\mathcal{C},\mathcal{X})$. The only case that needs attention is that of $t = C \in \mathcal{C}$, so $B \vdash_{\lambda\cap} \langle C\rangle_\lambda{:}\sigma$. Since $\mathcal{E}$ is principal for $\mathcal{C}$, $\mathcal{E}(C)$ is the principal type for $\langle C\rangle_\lambda$ in $\vdash_{\lambda\cap}$ and there exists (see [2]) a chain of operations *Ch* such that $Ch\,(\mathcal{E}(C)) = \sigma$. But then $B \vdash_{\mathcal{E}} C{:}\sigma$ by rule $(\mathcal{E})$.

ii) By induction on the definition of $\langle\ \rangle_\lambda$; the only alternative that needs consideration is that of $t = C \in \mathcal{C}$ where the last rule in the derivation for $B \vdash_{\mathcal{E}} t{:}\sigma$ is $(\mathcal{E})$. Then there is a chain *Ch* such that $Ch\,(\mathcal{E}(C)) = \sigma$. Let $C\,x_1 \cdots x_n \to r$ be the rule for C. Then, by Definition *2.13(ii)*, there are $\tau_1,\ldots,\tau_n \in \mathcal{T}$ and $\tau \in \mathcal{T}_s$, such that

$$\{x_1{:}\tau_1,\ldots,x_n{:}\tau_n\} \vdash_{\mathcal{E}} r{:}\tau \text{ and } \mathcal{E}(C) = \tau_1 \to\cdots\to \tau_n \to \tau.$$

Then, by induction, $\{x_1{:}\tau_1,\ldots,x_n{:}\tau_n\} \vdash_{\lambda\cap} r{:}\tau$ (notice that $\langle r\rangle_\lambda = r$). Then, by rule $(\to I)$ of $\vdash_{\lambda\cap}$, $\vdash_{\lambda\cap} \lambda x_1 \ldots x_n.r{:}\tau_1 \to\cdots\to \tau_n \to \tau$; since $\vdash_{\lambda\cap}$ is closed for all three operations of substitution, expansion, and lifting (see [3]), we also have $\vdash_{\lambda\cap} \lambda x_1 \ldots x_n.r{:}\sigma$, so $\vdash_{\lambda\cap} \langle C\rangle_\lambda{:}\sigma$. ∎

3 Subject reduction

In this section we will show that the notion of type assignment defined here on CS satisfies the subject reduction property (Theorem *3.7*). In order to achieve this, we first show that the three operations (type-substitution, expansion, and lifting) defined in the previous section are sound on typed terms. We will also show that derivation rule $(\mathcal{E})$ is sound in the following sense: if there is an operation *O* such that $O\,(\mathcal{E}(C)) = \sigma$, then, for every type $\tau \in \mathcal{T}_s$ such that $\sigma \leq \tau$, the combinator rule for C is typeable with respect to the changed environment $\mathcal{E}\,[C := \tau]$.

Proposition 3.1 (Soundness of type-substitution) *Let S be a type-substitution.*

i) If $B \vdash_{\mathcal{E}} t{:}\sigma$*, then* $S\,(B) \vdash_{\mathcal{E}} t{:}S\,(\sigma)$.

ii) If $C\,x_1 \cdots x_n \to r$ *is a combinator rule, typeable with respect to the environment* $\mathcal{E}$*, then it is typeable with respect to* $\mathcal{E}\,[C := S\,(\mathcal{E}(C))]$.

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

ii) By Definition *2.13(ii)*, there are types $\sigma_1,\ldots,\sigma_n,\sigma$, such that $\mathcal{E}(C) = \sigma_1 \to\cdots\to \sigma_n \to \sigma$, and $\{x_1{:}\sigma_1,\ldots,x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$. By part *(i)*, we obtain $S\,(\{x_1{:}\sigma_1,\ldots,x_n{:}\sigma_n\}) \vdash_{\mathcal{E}} r{:}S\,(\sigma)$, so also $\{x_1{:}S\,(\sigma_1),\ldots,x_n{:}S\,(\sigma_n)\} \vdash_{\mathcal{E}} r{:}S\,(\sigma)$, and $S\,(\sigma_1) \to\cdots\to S\,(\sigma_n) \to S\,(\sigma) = S\,(\mathcal{E}(C))$. ∎

The following essentially shows that lifting is sound:

Lemma 3.2 *i) If* $B \vdash_{\mathcal{E}} t{:}\sigma$ *and* $\sigma \leq \tau$*, then* $B \vdash_{\mathcal{E}} t{:}\tau$.

ii) If $B \vdash_{\mathcal{E}}^{\not\omega} t{:}\sigma$ *and* $\sigma \leq_{\not\omega} \tau$*, then* $B \vdash_{\mathcal{E}}^{\not\omega} t{:}\tau$*, where* $\leq_{\not\omega}$ *is the restriction of* $\leq$ *to* ω*-free types.*

iii) If $B \vdash_{\mathcal{E}} t{:}\sigma$ *and* $B' \leq B$*, then* $B' \vdash_{\mathcal{E}} t{:}\sigma$.

[1] It is possible to define the notion of principal environment directly for CS, without side-stepping to LC, but that would significantly increase the complexity of the proofs of this paper. It would not affect any of the results; in fact, the definition above would become a provable property.

iv) If $B \vdash_{\mathcal{E}} x{:}\sigma$ if and only if there is $x{:}\tau \in B$ such that $\tau \leq \sigma$.

Proof: We will only give the proof for the first part; the second is similar and the other two are straightforward. We will first consider σ, τ both in $\mathcal{T}_s$, then σ, τ in $\mathcal{T}$.

($\sigma, \tau \in \mathcal{T}_s$): This is proven by induction on the structure of terms.

($t \equiv x$): Then there exists $x{:}\rho \in B$ such that $\rho \leq \sigma$. Since also $\rho \leq \tau$, $B \vdash_{\mathcal{E}} x{:}\tau$.

($t \equiv C$): Then there is a chain *Ch* such that $Ch\,(\mathcal{E}(C)) = \sigma$. Since $\sigma \leq \tau$, $L = \langle \langle \emptyset, \sigma \rangle, \langle \emptyset, \tau \rangle \rangle$ is a lifting, then $Ch * [L]$ is a chain, therefore also $B \vdash_{\mathcal{E}} C{:}\tau$.

($t \equiv t_1 t_2$): So $B \vdash_{\mathcal{E}} t_1{:}\rho{\rightarrow}\sigma$, and $B \vdash_{\mathcal{E}} t_2{:}\rho$, for a certain ρ. Since $\sigma \leq \tau$, also $\rho{\rightarrow}\sigma \leq \rho{\rightarrow}\tau$; notice that both $\rho{\rightarrow}\sigma$ and $\rho{\rightarrow}\tau \in \mathcal{T}_s$. Then, by induction, $B \vdash_{\mathcal{E}} t_1{:}\rho{\rightarrow}\tau$, so by ($\rightarrow$*E*), $B \vdash_{\mathcal{E}} t_1 t_2{:}\tau$.

($\sigma = \sigma_1 \cap \cdots \cap \sigma_m, \tau = \tau_1 \cap \cdots \cap \tau_n$): Then, for every $1 \leq j \leq m$, $B \vdash_{\mathcal{E}} t{:}\sigma_i$. Then by Lemma *2.2*, for every $1 \leq i \leq n$, there is a $1 \leq j_i \leq m$ such that $\sigma_{j_i} \leq \tau_i$, and notice that $\sigma_{j_i}, \tau_i \in \mathcal{T}_s$. Therefore, for every $1 \leq i \leq n$, $B \vdash_{\mathcal{E}} t{:}\tau_i$. Then by ($\cap$*I*), $B \vdash_{\mathcal{E}} t{:}\tau_1 \cap \cdots \cap \tau_n$. ■

Proposition 3.3 (Soundness of lifting) *Let L be a lifting such that $L\,(\langle B, \sigma, E \rangle) = \langle B', \sigma', E' \rangle$.*

i) If $B \vdash_{\mathcal{E}} t{:}\sigma$, then $B' \vdash_{\mathcal{E}} t{:}\sigma'$.

ii) If $C\,x_1 \cdots x_n \rightarrow r$ is a combinator rule, typeable with respect to the environment $\mathcal{E}$, it is typeable with respect to $\mathcal{E}\,[C := L\,(\mathcal{E}(C))]$.

Proof: *i)* By Lemma *3.2*.

ii) By Definition *2.13 (ii)*, there are $\sigma_1, \ldots, \sigma_n, \sigma$, such that $\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$, and $\mathcal{E}(C) = \sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma$. Since $\sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma \leq L\,(\sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma)$, because of Definition *2.1 (ii)*, there are $\tau_1, \ldots, \tau_n, \tau$, such that $L\,(\sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma) = \tau_1{\rightarrow}\cdots{\rightarrow}\tau_n{\rightarrow}\tau$, and for $1 \leq i \leq n$, $\tau_i \leq \sigma_i$, and $\sigma \leq \tau$. So $L' = \langle \langle \{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}, \sigma \rangle, \langle \{x_1{:}\tau_1, \ldots, x_n{:}\tau_n\}, \tau \rangle \rangle$ is a lifting, and by part *(i)*, we obtain $L'\,(\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}) \vdash_{\mathcal{E}} r{:}L'\,(\sigma)$, so $\{x_1{:}\tau_1, \ldots, x_n{:}\tau_n\} \vdash_{\mathcal{E}} r{:}\tau$. ■

Proposition 3.4 (Soundness of expansion) *Let Ex be an expansion such that $Ex\,(\langle B, \sigma, E \rangle) = \langle B', \sigma', E' \rangle$.*

i) If $B \vdash_{\mathcal{E}} t{:}\sigma$, then $B' \vdash_{\mathcal{E}} t{:}\sigma'$.

ii) If $C\,x_1 \cdots x_n \rightarrow r$ is a rule, typeable with respect to $\mathcal{E}$, and $Ex\,(\mathcal{E}(C)) = \tau_1 \cap \cdots \cap \tau_m \in \mathcal{T}$ ($m \geq 1$), then, for every $1 \leq j \leq m$, the rule is typeable with respect to $\mathcal{E}\,[C := \tau_j]$.

Proof: *i)* By induction on $\mathcal{T}$. We will only show the part $\sigma \in \mathcal{T}_s$. Then, by Lemma *2.8* either:

a) $\sigma' = \tau_1 \cap \cdots \cap \tau_m$, $B' = \Pi\{B_1, \ldots, B_m\}$, and for every $1 \leq j \leq m$, there is a type-substitution *S* such that $S\,(\langle B, \sigma \rangle) = \langle B_j, \tau_j \rangle$. Then, by Proposition *3.1 (i)*, for every $1 \leq j \leq m$, $B_j \vdash_{\mathcal{E}} t{:}\tau_j$. Therefore, by Proposition *3.3*, since $\Pi\{B_1, \ldots, B_m\} \leq B_j$ for every $1 \leq j \leq m$, $B' \vdash_{\mathcal{E}} t{:}\tau_j$, and by ($\cap$*I*), $B' \vdash_{\mathcal{E}} t{:}\sigma'$.

b) $\sigma' \in \mathcal{T}_s$. This part is proved by induction on the structure of terms.

($t = x$): Then, by ($\leq$), there is $x{:}\tau \in B$, such that $\tau \leq \sigma$. By Lemma *2.9 (i)*, $Ex\,(\tau) \leq \sigma'$, so $B' \vdash_{\mathcal{E}} x{:}\sigma'$.

($t = C$): Then, by ($\mathcal{E}$), there is a chain *Ch* such that $Ch\,(\mathcal{E}(C)) = \sigma$. Let Ex' be the expansion defined by $\langle \mu', n \rangle$ where μ' is the intersection of the types affected by *Ex*. Note that $E'(\sigma) = \sigma'$. Since $Ch * [Ex']$ is a chain and $Ch * [Ex']\,(\mathcal{E}(C)) = \sigma'$, we obtain $B' \vdash_{\mathcal{E}} C{:}\sigma'$.

($t = t_1 t_2$): Then, by ($\rightarrow$*E*), there is τ such that $B \vdash_{\mathcal{E}} t_1{:}\tau{\rightarrow}\sigma$ and $B \vdash_{\mathcal{E}} t_2{:}\tau$. Let Ex' be the expansion defined by $\langle \mu', n \rangle$, where μ' is the intersection of the types affected by *Ex*. By induction, Ex' is sound for the derivations $B \vdash_{\mathcal{E}} t_1{:}\tau{\rightarrow}\sigma$ and $B \vdash_{\mathcal{E}} t_2{:}\tau$, that is, $Ex'(B) \vdash_{\mathcal{E}} t_1{:}Ex'(\tau{\rightarrow}\sigma)$ and $Ex'(B) \vdash_{\mathcal{E}} t_2{:}Ex'(\tau)$. Note that $Ex'(B) = B'$ and

$Ex'(\sigma) = \sigma'$, and since $\sigma' \in \mathcal{T}_s$, $Ex'(\tau{\rightarrow}\sigma) = Ex'(\tau) \rightarrow \sigma'$. Therefore, $B' \vdash_{\mathcal{E}} t_1 t_2{:}\sigma'$.

ii) Since $\mathcal{E}(C) \in \mathcal{T}_s$, by Lemma *2.8* either:

a) $m > 1$. By Definition *2.7*, for every $1 \leq j \leq m$, there is a type-substitution S such that $S(\mathcal{E}(C)) = \tau_j$. The proof is completed by Theorem *3.1 (ii)*.

b) $m = 1$. By Definition *2.13 (ii)*, there are $\sigma_1, \ldots, \sigma_n, \sigma$, such that $\sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma = \mathcal{E}(C)$, $\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$, and $Ex\,(\sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma) = \tau$. By the result in part *(i)*, we obtain $Ex\,(\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}) \vdash_{\mathcal{E}} r{:}Ex\,(\sigma)$, so also $\{x_1{:}Ex\,(\sigma_1), \ldots, x_n{:}Ex\,(\sigma_n)\} \vdash_{\mathcal{E}} r{:}Ex\,(\sigma)$. Since $\tau \in \mathcal{T}_s$, also $\tau = Ex\,(\sigma_1){\rightarrow}\cdots{\rightarrow}Ex\,(\sigma_n){\rightarrow}Ex\,(\sigma)$. ■

Combining the above results for the different operations, we have:

Theorem 3.5 (Soundness of chains) *i) Let $B \vdash_{\mathcal{E}} t{:}\sigma$ and Ch be a chain such that $Ch\,(\langle B, \sigma, E\rangle) = \langle B', \sigma', E'\rangle$, then $B' \vdash_{\mathcal{E}} t{:}\sigma'$.*

ii) Let $l \rightarrow_C r$ be a combinator rule typeable with respect to the environment $\mathcal{E}$. If $Ch\,(\mathcal{E}(C)) = \tau \in \mathcal{T}$, then, for every $\mu \in \mathcal{T}_s$ such that $\tau \leq \mu$, C is typeable with respect to $\mathcal{E}[C := \mu]$.

Proof: By Propositions *3.1*, *3.4*, and *3.3*. ■

Using this soundness result, we will now show that the notion of type assignment as defined in this paper satisfies the subject reduction property: if $B \vdash_{\mathcal{E}} t{:}\sigma$, and t can be rewritten to t', then $B \vdash_{\mathcal{E}} t'{:}\sigma$. Of course, this result can be obtained through the mappings $[\![\ ]\!]_{\mathcal{C}}$ and $\langle\ \rangle_{\lambda}$, using the relations between the systems mentioned in the previous section, but only for combinatory complete CS and principal environments. For other CS, we must give a direct proof, for which we need the following term-substitution result.

Lemma 3.6 *i) If $B \vdash_{\mathcal{E}} t{:}\sigma$, then, for every term-substitution* R *and basis B', if for every $x{:}\tau \in B$, $B' \vdash_{\mathcal{E}} x^{\mathrm{R}}{:}\tau$, then $B' \vdash_{\mathcal{E}} t^{\mathrm{R}}{:}\sigma$.*

ii) Let $C x_1 \cdots x_n \rightarrow r$ be a combinator rule, typeable with respect to $\mathcal{E}$. For every term-substitution R, *basis B and type μ: if $B \vdash_{\mathcal{E}} C x_1 \cdots x_n{}^{\mathrm{R}}{:}\mu$, then $B \vdash_{\mathcal{E}} r^{\mathrm{R}}{:}\mu$.*

Proof: *i)* By induction on $\vdash_{\mathcal{E}}$.

($\leq$): Then $t = x$. Then there is $x{:}\tau \in B$, such that $\tau \leq \sigma$. Then, by Theorem *3.3*, $B' \vdash_{\mathcal{E}} x^{\mathrm{R}}{:}\tau$ implies $B' \vdash_{\mathcal{E}} x^{\mathrm{R}}{:}\sigma$.

($\mathcal{E}$): Then $t = C$. Immediate, since $C^{\mathrm{R}} = C$, and $C{:}\sigma$ does not depend on the basis.

($\rightarrow E$), ($\cap I$): By induction.

ii) If $C x_1 \cdots x_n \rightarrow r$ is a typeable combinator rule, then by Definition *2.13 (ii)*, there are $\sigma_1, \ldots, \sigma_n, \sigma$, such that $\mathcal{E}(C) = \sigma_1{\rightarrow}\cdots{\rightarrow}\sigma_n{\rightarrow}\sigma$ and $\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$. Also, $(C x_1 \cdots x_n)^{\mathrm{R}} = C x_1{}^{\mathrm{R}} \cdots x_n{}^{\mathrm{R}}$. From $B \vdash_{\mathcal{E}} C x_1{}^{\mathrm{R}} \cdots x_n{}^{\mathrm{R}}{:}\mu$, we know that there are $\mu_1, \ldots, \mu_n$, and a chain *Ch* such that $Ch\,(\mathcal{E}(C)) = \mu_1{\rightarrow}\cdots{\rightarrow}\mu_n{\rightarrow}\mu$, and, for $1 \leq i \leq n$, $B \vdash_{\mathcal{E}} x_i{}^{\mathrm{R}}{:}\mu_i$. Since $\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$, by Theorem *3.5 (i)*, $\{x_1{:}\mu_1, \ldots, x_n{:}\mu_n\} \vdash_{\mathcal{E}} r{:}\mu$. Then, by part *(i)*, also $B \vdash_{\mathcal{E}} r^{\mathrm{R}}{:}\sigma$. ■

Using this result, the following becomes easy.

Theorem 3.7 (Subject reduction) *If $B \vdash_{\mathcal{E}} t{:}\sigma$ and $t \rightarrow^* t'$, then $B \vdash_{\mathcal{E}} t'{:}\sigma$.*

Proof: By induction to the length of the reduction path; the case of length 1 is proved by induction on the structure of t. Of this double induction, only the case that t itself is the term-substitution instance of a left-hand side of a combinator rule is of interest; all other cases are straightforward. Then, let C and R be such that $l \rightarrow_C r$, $t = l^{\mathrm{R}}$, and $t' = r^{\mathrm{R}}$. The result follows from Lemma *3.6(ii)*. ∎

One should remark that a subject expansion theorem, i.e. the converse of the subject reduction result,

$$\textit{If } B \vdash_{\mathcal{E}} t{:}\sigma\textit{, and } t' \rightarrow t\textit{, then } B \vdash_{\mathcal{E}} t'{:}\sigma\textit{,}$$

does not hold in general. Take for example the CS

$$\begin{array}{lcl} \mathbf{K}xy & \rightarrow & x \\ \mathbf{I}x & \rightarrow & x \end{array}$$

that is typeable with respect to the environment

$$\begin{array}{lcl} \mathcal{E}(\mathbf{K}) & = & \varphi_1{\rightarrow}\omega{\rightarrow}\varphi_1, \\ \mathcal{E}(\mathbf{I}) & = & (\varphi_2{\rightarrow}\varphi_2){\rightarrow}\varphi_2{\rightarrow}\varphi_2. \end{array}$$

The term $\mathbf{IK}$ reduces to the (head-)normal form $\mathbf{K}$, but can only be typed by ω with respect to $\mathcal{E}$. Of course, $(\varphi_2{\rightarrow}\varphi_2){\rightarrow}\varphi_2{\rightarrow}\varphi_2$ is not the principal type for $\langle \mathbf{I} \rangle_\lambda$ in $\vdash_{\lambda\cap}$. In fact, we have the following result:

Theorem 3.8 (Subject expansion) *Let* $((\mathcal{C},\mathcal{X}),\mathbf{R})$ *be a* CS*, and* $\mathcal{E}$ *be principal for* $\mathcal{C}$*. If* $B \vdash_{\mathcal{E}} t{:}\sigma$ *and* $t' \rightarrow t$*, then* $B \vdash_{\mathcal{E}} t'{:}\sigma$*.*

Proof: If $B \vdash_{\mathcal{E}} t{:}\sigma$, then by Lemma *2.18(ii)*, also $B \vdash_{\lambda\cap} \langle t \rangle_\lambda{:}\sigma$. Since $t' \rightarrow t$, by Propostion *1.8* also $\langle t' \rangle_\lambda \twoheadrightarrow_\beta \langle t \rangle_\lambda$. Since $\vdash_{\lambda\cap}$ is closed for β-expansion, we have $B \vdash_{\lambda\cap} \langle t' \rangle_\lambda{:}\sigma$. Then, by Theorem *2.18(i)*, we have $B \vdash_{\mathcal{E}} t'{:}\sigma$. ∎

4 Restricted type assignment

Our aim is to define, in Section *5*, a notion of reduction on type derivations (Cut Elimination) which is strongly normalizing. For this, reduction will be, as can be expected, guided by the appearance of typeable redexes of $\rightarrow_{\mathbf{R}}$ in the conclusion of the type derivation. Each occurrence of a redex will be treated independently, since the types assigned to each occurrence of the same redex might differ.

Since derivation reduction creates a new type derivation, some care is needed to make sure that *all* necessary sub-derivations are contracted, and no reduction is attempted where it is not possible. Moreover, derivation reduction is not a '*Cut and Paste*' operation as in the LC, in the sense that, for combinator systems, the derivation that is created for the contractum is not completely constructed out of parts of the derivation for the redex: additional structure needs to be introduced, extending the size of the derivation.

In order to simplify the definition of the reduction relation, we will first define a notion of type assignment on terms in $T(\mathcal{C},\mathcal{X})$ (denoted by $\overset{\mathrm{r}}{\vdash}_{\mathcal{E}}$) that is a slight variant of the notion of type assignment in Definition *2.13*. The variation consists, essentially, of restricting bases to their relevant contents, i.e. to contain only *the types actually used* for the variables of a term. In the next section, we will prove that derivations in this system are strongly normalizable; for this we will use the well-known method of Computability Predicates [20]. Then, in Section *8*, we will show that the approximation theorem

$$\textit{If } B \vdash_{\mathcal{E}} t{:}\sigma\textit{, then there exists } a \in \mathcal{A}_{\mathcal{C}}(t) \textit{ such that } B \vdash_{\mathcal{E}} a{:}\sigma\textit{,}$$

as well as the three normalization properties stated in the introduction of this paper, are consequences of this strong normalization result for $\overset{\mathrm{r}}{\vdash}_{\mathcal{E}}$.

Definition 4.1 (*Restricted type assignment*) Let $(\Sigma, \mathbf{R})$ be a CS and $\mathcal{E}$ an environment. *Restricted type assignment* and *restricted derivations* are defined by the following natural deduction system (where all types displayed are in $\mathcal{T}_s$, except for τ in rule $(\rightarrow E)$):

$$(\mathcal{E}): \quad \frac{\exists Ch\,[Ch\,(\mathcal{E}(C)) = \sigma]}{\emptyset \Vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma} \qquad (\rightarrow E): \quad \frac{B_1 \Vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\tau{\rightarrow}\sigma \qquad B_2 \Vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\tau}{\Pi\{B_1, B_2\} \Vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma}$$

$$(Ax): \quad \frac{}{\{x{:}\sigma\} \Vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\sigma} \qquad (\cap I): \quad \frac{B_1 \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1 \quad \ldots \quad B_n \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_n}{\Pi\{B_1, \ldots, B_n\} \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1 \cap \cdots \cap \sigma_n} \ (n \geq 0)$$

We write D :: $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ if and only if there is a restricted derivation D that has $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ as conclusion, and write $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ if there exists a D such that D :: $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$.

Notice that, in rule $(\cap I)$, if $n = 0$, then $\Pi\{B_1, \ldots, B_n\} = \emptyset$ and $\sigma_1 \cap \cdots \cap \sigma_n = \omega$. Notice also that the main difference between $\vdash_{\mathcal{E}}$ and $\Vdash^{\mathrm{r}}_{\mathcal{E}}$ lies in the fact that rule $(\leq)$ has been replaced by (Ax). Also, in rule $(\rightarrow E)$ for $\vdash_{\mathcal{E}}$, the bases used in left- and right-hand subderivation have to be the same, whereas for that rule in $\Vdash^{\mathrm{r}}_{\mathcal{E}}$, this need not be the case: the respective bases are combined, using the operation $\Pi\{\ \}$. We could have used this restricted system throughout this paper, without losing any important result (see also the next lemma). But since one of the objectives was to obtain at least the expressive power of the intersection type assignment system for LC (Theorem *2.18(i)*), the choice for the full system has been to allow also types in bases that are not relevant to the type assigned to the term, i.e. for derivation rule $(\leq)$ rather than (Ax).

The relation between the two notions of type assignment $\Vdash^{\mathrm{r}}_{\mathcal{E}}$ and $\vdash_{\mathcal{E}}$ is strong, and formulated by:

Lemma 4.2 *i) If $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, then $B \vdash_{\mathcal{E}} t{:}\sigma$.*

ii) If $B \vdash_{\mathcal{E}} t{:}\sigma$, then there is a B' such that $B \leq B'$ and $B' \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$.

iii) If $B \vdash_{\mathcal{E}} t{:}\sigma$ without using ω, then there is a B' such that $B \leq B'$ and $B' \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ without using ω.

Proof: By straightforward induction on the structure of derivations. ■

Using these relations, the following lemma, that shows a subject-reduction result for restricted type assignment, becomes easy.

Theorem 4.3 *If $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau$ and $t \rightarrow^* v$, then there exist B' such that $B \leq B'$ and $B' \Vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\tau$.*

Proof: If $B \Vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau$, by Lemma *4.2(i)*, also $B \vdash_{\mathcal{E}} t{:}\tau$. Since $t \rightarrow^* v$, by Theorem *3.7*, also $B \vdash_{\mathcal{E}} v{:}\tau$. Then, by Lemma *4.2(ii)*, there exists a B' such that $B \leq B'$ and $B' \Vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\tau$. ■

Example 4.4 Let *Ch* be such that $Ch\,(\mathcal{E}\,(\mathbf{K})) = \sigma{\rightarrow}\tau{\rightarrow}\sigma$, then, using *Ch*, we have $\{x{:}\sigma\cap\tau\} \Vdash^{\mathrm{r}}_{\mathcal{E}} \mathbf{K}xx{:}\sigma$, $\mathbf{K}xx \rightarrow x$, and $\{x{:}\sigma\} \Vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\sigma$. Notice that $\{x{:}\sigma\cap\tau\} \leq \{x{:}\sigma\}$.

We will use a short-hand notation for derivations.

Definition 4.5 *i)* We write D $= \langle Ax \rangle$, if and only if the type derivation D consists of nothing but an application of (Ax), i.e. there are x and σ, such that D :: $\{x{:}\sigma\} \Vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\sigma$.

ii) We write D $= \langle \mathcal{E} \rangle$, if and only if D consists of nothing but an application of rule $(\mathcal{E})$, i.e. there are C and σ, such that D :: $\emptyset \Vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma$.

iii) We write D $= \langle \mathrm{D}_1, \mathrm{D}_2, \rightarrow E \rangle$, if and only if D is obtained from D_1 and D_2 by applying rule $(\rightarrow E)$, i.e. if there are $B_1, B_2, t_1, t_2, \sigma$, and τ such that

$$\mathrm{D}_1 :: B_1 \Vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\tau{\rightarrow}\sigma,\ \mathrm{D}_2 :: B_2 \Vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\tau, \text{ and } \mathrm{D} :: \Pi\{B_1, B_2\} \Vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma.$$

iv) We write $\mathrm{D} = \langle \mathrm{D}_1, \ldots, \mathrm{D}_n, \cap I\rangle$, if and only if D is obtained from $\mathrm{D}_1, \ldots, \mathrm{D}_n$ by applying rule ($\cap I$), i.e., for every $1 \leq i \leq n$, there are B_i and σ_i such that $\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_i$, and

$$\mathrm{D} :: \Pi\{B_1, \ldots B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1\cap\cdots\cap\sigma_n.$$

Below, in the definition of derivation reduction, we will need the following result.

Lemma 4.6 *If* $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ *and* $\sigma \leq \tau$, *then there are* D' *and* B', *such that* $B \leq B'$ *and* $\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau$.

Proof: We will prove, like for Lemma *3.2*, this lemma in two stages: first for σ, τ both in $\mathcal{T}_s$, then for σ, τ in $\mathcal{T}$.

i) $\sigma, \tau \in \mathcal{T}_s$. This is proven by induction on the structure of terms.

a) $t \equiv x$. Then $D = \langle Ax\rangle :: \{x{:}\sigma\} \vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\sigma$. Notice that $\{x{:}\sigma\} \leq \{x{:}\tau\}$, and $\mathrm{D}' = \langle Ax\rangle :: \{x{:}\tau\} \vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\tau$.

b) $t \equiv C$. Then $\mathrm{D} = \langle \mathcal{E}\rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma$, so there is a chain *Ch* such that $Ch\,(\mathcal{E}(C)) = \sigma$. Since $\sigma \leq \tau$, $L = \langle\langle\emptyset, \sigma\rangle, \langle\emptyset, \tau\rangle\rangle$ is a lifting, $Ch * [L]$ is a chain, and therefore also $\langle \mathcal{E}\rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\tau$.

c) $t \equiv t_1 t_2$, so $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\rho\rightarrow\sigma, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\rho, \rightarrow E\rangle :: \Pi\{B_1, B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma$, for a certain ρ. Since $\sigma \leq \tau$, also $\rho\rightarrow\sigma \leq \rho\rightarrow\tau$; notice that both $\rho\rightarrow\sigma$ and $\rho\rightarrow\tau \in \mathcal{T}_s$. Then, by induction, there exists B'_1 such that $B_1 \leq B'_1$ and $\mathrm{D}'_1 :: B'_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\rho\rightarrow\tau$. Then $\Pi\{B_1, B_2\} \leq \Pi\{B'_1, B_2\}$, and, by ($\rightarrow E$), there exists

$$\mathrm{D}' = \langle \mathrm{D}'_1, \mathrm{D}_2, \rightarrow E\rangle :: \Pi\{B'_1, B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\tau.$$

ii) $\sigma = \sigma_1\cap\cdots\cap\sigma_m, \tau = \tau_1\cap\cdots\cap\tau_n$. Then, by ($\cap I$), $B = \Pi\{B_1, \ldots, B_m\}$ and, for every $1 \leq j \leq m$, $B_j \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_i$. Then by Lemma *2.2*, for every $1 \leq i \leq n$, there is a $1 \leq j_i \leq m$ such that $\sigma_{j_i} \leq \tau_i$. So, by part *(i)*, for every $1 \leq i \leq n$, there is a B_{j_i} such that $B_i \leq B_{j_i}$ and $\mathrm{D}_{j_i} :: B_{j_i} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau_i$. Then $\Pi\{B_1, \ldots, B_n\} \leq \Pi\{B_{j_1}, \ldots, B_{j_n}\}$, and, by ($\cap I$), there exists

$$\mathrm{D}' = \langle \mathrm{D}_{j_i}, \ldots, \mathrm{D}_{j_i}, \cap I\rangle :: \Pi\{B_{j_1}, \ldots, B_{j_n}\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau_1\cap\cdots\cap\tau_n.$$

Notice that $\tau = \omega$ is a special case of *(ii)*; then, by construction, $B' = \emptyset$. ■

5 Derivation reduction

In this section, we will introduce a notion of reduction on derivations $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$. The effect of this reduction will be that the subderivation for a redex occurring in t (with type different from ω) will be replaced by the derivation for an instance of the right-hand side of the applied rewrite rule. We will show that this notion of reduction is strongly normalizing.

Before formally defining reduction on derivations, we will define a notion of substitution on derivations, that will consist of replacing a type derivation for a variable by another derivation.

Definition 5.1 (*Derivation substitution*) For $\mathrm{D} :: B, x{:}\sigma \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau$ and $\mathrm{D}_v :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma$, the result D' of *substituting* D_v *in* D, denoted by $\mathrm{D}' = \mathrm{D}\,[\mathrm{D}_v/x{:}\sigma] :: B'' \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\{x\mapsto v\}}{:}\tau$, is inductively defined as follows:

i) $\mathrm{D} = \langle Ax\rangle :: \{y{:}\tau\} \vdash^{\mathrm{r}}_{\mathcal{E}} y{:}\tau$. If $x = y$, then $\sigma = \tau$, and $\mathrm{D}' = \mathrm{D}_v :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\tau$; otherwise, $\mathrm{D}' = \mathrm{D}$.

ii) $\mathrm{D} = \langle \mathcal{E}\rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\tau$. Then $\mathrm{D}' = \mathrm{D}$.

iii) $\mathrm{D} = \langle \mathrm{D}_1 :: B_1, x{:}\sigma_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\rho\rightarrow\tau, \mathrm{D}_2 :: B_2, x{:}\sigma_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\rho, \rightarrow E\rangle :: \Pi\{B_1, B_2\}, x{:}\sigma \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\tau$. In particular, $\sigma = \sigma_1\cap\sigma_2 = \alpha_1\cap\cdots\cap\alpha_m$ where $\alpha_1, \ldots, \alpha_m \in \mathcal{T}_s$, such that .

$$\mathrm{D}_v = \langle \mathrm{D}^1_v :: B'_1 \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\alpha_1, \ldots, \mathrm{D}^m_v :: B'_m \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\alpha_m, \cap I\rangle.$$

Assume, without loss of generality, that $\sigma_1 = \alpha_1 \cap \ldots \cap \alpha_j$ and $\sigma_2 = \alpha_{j+1} \cap \ldots \cap \alpha_m$, and let $\mathrm{D}^1 = \langle \mathrm{D}_v^1, \ldots, \mathrm{D}_v^j, \cap I \rangle :: B^1 \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_1$, and $\mathrm{D}^2 = \langle \mathrm{D}_v^{j+1}, \ldots, \mathrm{D}_v^m, \cap I \rangle :: B^2 \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_2$. Let

$$\begin{array}{rcl} \mathrm{D}'_1 & = & \mathrm{D}_1\,[\mathrm{D}^1/x{:}\sigma_1] :: B''_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1^{\{x\mapsto v\}} : \rho\to\tau, \text{ and} \\ \mathrm{D}'_2 & = & \mathrm{D}_2\,[\mathrm{D}^2/x{:}\sigma_2] :: B''_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2^{\{x\mapsto v\}} : \rho. \end{array}$$

Then $\mathrm{D}' = \langle \mathrm{D}'_1, \mathrm{D}'_2, \to E \rangle :: \Pi\{B''_1, B''_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} (t_1\, t_2)^{\{x\mapsto v\}} : \tau$.

iv) $\mathrm{D} = \langle \mathrm{D}_1, \ldots, \mathrm{D}_n, \cap I \rangle :: B, x{:}\sigma \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau_1 \cap \cdots \cap \tau_n$. For $1 \leq i \leq n$, there are σ_i, B_i, t_i such that $\sigma = \sigma_1 \cap \cdots \cap \sigma_n, B = \Pi\{B_1, \ldots, B_n\}$, and $\mathrm{D}_i :: B_i, x{:}\sigma_i \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\tau_i$. Since $\mathrm{D}_v :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_1 \cap \cdots \cap \sigma_n$, reasoning as above in part *(iii)*, for every $1 \leq i \leq n$, there are D^i, B^i, such that $\mathrm{D}^i :: B^i \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_i$. Let $\mathrm{D}'_i = \mathrm{D}_i\,[\mathrm{D}^i/x{:}\sigma_i] :: B'_i \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\{x\mapsto v\}}{:}\tau_i$, then

$$\mathrm{D}' = \langle \mathrm{D}'_1, \ldots, \mathrm{D}'_n, \cap I \rangle :: \Pi\{B'_1, \ldots B'_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\{x\mapsto v\}} : \tau_1 \cap \cdots \cap \tau_n.$$

Before coming to the definition of derivation-reduction, we need to define the concept of 'the position of a sub-derivation in a derivation'.

Definition 5.2 Let D be a derivation, and D$'$ be a sub-derivation of D. The *position p of* D$'$ *in* D is defined by:

i) If $\mathrm{D}' = \mathrm{D}$, then $p = \varepsilon$.

ii) If the position of D$'$ in D_1 is q and $\mathrm{D} = \langle \mathrm{D}_1, \mathrm{D}_2, \to E \rangle$, then $p = 1q$.

iii) If the position of D$'$ in D_2 is q and $\mathrm{D} = \langle \mathrm{D}_1, \mathrm{D}_2, \to E \rangle$, then $p = 2q$.

iv) If the position of D$'$ in D_i, for some $1 \leq i \leq n$, is q, and $\mathrm{D} = \langle \mathrm{D}_1, \ldots, \mathrm{D}_n, \cap I \rangle$, then $p = q$.

Notice that if p is the position of a sub-derivation $\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma'$ in $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, then p is also the position of an occurrence of t' in t.

Remark 5.3 Let $\langle \mathrm{D}_1, \ldots, \mathrm{D}_n, \cap I \rangle :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1 \cap \cdots \cap \sigma_n$. Notice that, if $D_0 :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\rho$ is a sub-derivation of D_j $(1 \leq j \leq n)$ at position p, then, for $1 \leq i \neq j \leq n$, either:

- there is no sub-derivation in D_i at position p, or
- D_i has a sub-derivation $\langle \cap I \rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\omega$ at position p, or
- D_i has a sub-derivation $\mathrm{D}'_0 :: B'' \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\rho'$ (with $\rho' \in \mathcal{T}_s$) at position p.

We can now give a definition of reduction on derivations in $\vdash^{\mathrm{r}}_{\mathcal{E}}$; notice that this reduction corresponds to contracting a redex in the term that appears in the conclusion, and building a derivation for the contractum.

Definition 5.4 (*Derivation reduction)* We define reduction on $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ by induction on σ. We say that $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ reduces to $\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma$ at position p if either:

i) $\sigma \in \mathcal{T}_s$. There are three cases depending on whether t reduces at the root position or not.

a) If $t = (C x_1 \cdots x_n)^{\mathrm{R}} = C t_1 \cdots t_n$ and there is a combinator rule $C x_1 \cdots x_n \to r$, then D has the form:

$$\dfrac{\dfrac{\dfrac{\exists\mathit{Ch}\,[\mathit{Ch}\,(\mathcal{E}(C)) = \sigma_1{\to}\cdots{\to}\sigma_n{\to}\sigma]}{\emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma_1{\to}\cdots{\to}\sigma_n{\to}\sigma} \qquad \dfrac{\mathrm{D}_1}{B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\sigma_1}}{\dfrac{B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} C t_1{:}\sigma_2{\to}\cdots{\to}\sigma_n{\to}\sigma}{\vdots}} \qquad \dfrac{\mathrm{D}_n}{B_n \vdash^{\mathrm{r}}_{\mathcal{E}} t_n{:}\sigma_n}}{\Pi\{B_1,\ldots,B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} C t_1\cdots t_n{:}\sigma}$$

Then, by Definition *2.13(ii)* and Theorem *3.5*, $\{x_1{:}\sigma_1,\ldots,x_n{:}\sigma_n\} \vdash_{\mathcal{E}} r{:}\sigma$. By Lemma *4.2(ii)*, there are D'_0, and $\alpha_1,\ldots,\alpha_n$, such that, for every $1 \leq i \leq n$, $\sigma_i \leq \alpha_i$ (α_i might be ω), and

$$\mathrm{D}'_0 :: \{x_1{:}\alpha_1,\ldots,x_n{:}\alpha_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} r{:}\sigma.$$

Then, by Lemma *4.6*, for every $1 \leq i \leq n$, there are $\mathrm{D}'_i, B'_i \geq B_i$ such that $\mathrm{D}'_i :: B'_i \vdash^{\mathrm{r}}_{\mathcal{E}} t_i{:}\alpha_i$. Let $\mathrm{R} = \{x_1 \mapsto t_1,\ldots,x_n \mapsto t_n\}$, $t' = r^{\mathrm{R}}$, and

$$\mathrm{D}' = \mathrm{D}'_0\,[\mathrm{D}'_1/x_1{:}\alpha_1,\ldots,\mathrm{D}'_n/x_n{:}\alpha_n] :: \Pi\{B'_1,\ldots,B'_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t' : \sigma,$$

then $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ reduces to D' at position ε.

b) If $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\tau{\to}\sigma, \mathrm{D}_2, {\to}E\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma$, and D_1 reduces at position q to $\mathrm{D}'_1 :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'_1{:}\tau{\to}\sigma$, then D reduces to $\mathrm{D}' = \langle \mathrm{D}'_1, \mathrm{D}_2, {\to}E\rangle :: \Pi\{B',B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t'_1 t_2{:}\sigma$ at position $1q$.

c) If $\mathrm{D} = \langle \mathrm{D}_1, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\tau, {\to}E\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma$, and D_2 reduces at position q to $\mathrm{D}'_2 :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'_2{:}\tau$, then D reduces to $\mathrm{D}' = \langle \mathrm{D}_1, \mathrm{D}'_2, {\to}E\rangle :: \Pi\{B_1,B'\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t'_2{:}\sigma$ at position $2q$.

ii) $\sigma = \sigma_1{\cap}\cdots{\cap}\sigma_n$, $n > 0$. If $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1{\cap}\cdots{\cap}\sigma_n$, then there are $\mathrm{D}_1,\ldots,\mathrm{D}_n, B_1,\ldots,B_n$, such that $B = \Pi\{B_1,\ldots,B_n\}$, and, for every $1 \leq i \leq n$, $\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_i$, and $\mathrm{D} = \langle \mathrm{D}_1,\ldots,\mathrm{D}_n, {\cap}I\rangle$. If there is some $1 \leq j \leq n$ such that D_j reduces at position p to $\mathrm{D}'_j :: B'_j \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma_j$, then, by Remark *5.3*, for $1 \leq i \neq j \leq n$, either

a) there is no sub-derivation in D_i at position p, or D_i has a sub-derivation $\langle {\cap}I\rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\omega$ at position p; then take $\mathrm{D}'_i :: B'_i \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma_i$ with the same structure as D_i, and $B_i = B'_i$.

b) D_i has a sub-derivation at position p, and D_i reduces to $\mathrm{D}'_i :: B'_i \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma_i$ at position p.

Then D reduces to $\mathrm{D}' = \langle \mathrm{D}'_1,\ldots,\mathrm{D}'_n, {\cap}I\rangle :: \Pi\{B'_1,\ldots,B'_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma_1{\cap}\cdots{\cap}\sigma_n$ at position p.

We write $\mathrm{D}_1 \to_{\mathcal{D}} \mathrm{D}_2$ if there is a p such that D_1 reduces to D_2 at position p, and write $\to^*_{\mathcal{D}}$ for its reflexive and transitive closure.

Notice that D is reducible if and only if there is a subderivation $\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} C u_1\cdots u_n{:}\rho$, with $\rho \in \mathcal{T}_s$ and $n \geq \mathit{arity}\,(C)$. We write $\mathit{SN}\,(\mathrm{D})$ to indicate that D is strongly normalizable with respect to $\to_{\mathcal{D}}$.

The following properties hold:

Lemma 5.5 *i) If* $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma \to_{\mathcal{D}} \mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma$, *then* $B \leq B'$, *and* $t \to t'$.

ii) Let $\mathrm{D} = \langle \mathrm{D}_1, \mathrm{D}_2, {\to}E\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma$. *Then:* $\mathit{SN}\,(\mathrm{D})$ *implies* $\mathit{SN}\,(\mathrm{D}_1)$ *and* $\mathit{SN}\,(\mathrm{D}_2)$.

iii) If $\mathit{SN}\,(\mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} x t_1\cdots t_n{:}\sigma{\to}\tau)$ *and* $\mathit{SN}\,(\mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\sigma)$, *then* $\mathit{SN}\,(\langle \mathrm{D}_1, \mathrm{D}_2, {\to}E\rangle)$.

iv) If $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_2, {\cap}I\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1{\cap}\sigma_2$, *and* $\mathrm{D} \to_{\mathcal{D}} \mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma$ *then there are* $B'_1 \geq B_1, B'_2 \geq B_2$ *such that* $B' = \Pi\{B'_1,B'_2\}$, *and*

$$\mathrm{D}_1 \to_{\mathcal{D}} \mathrm{D}'_1 :: B'_1 \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_1 \;\; \textit{or} \;\; \mathrm{D}_2 \to_{\mathcal{D}} \mathrm{D}'_2 :: B'_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t'{:}\sigma_2.$$

v) If $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_2, {\cap}I\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1{\cap}\sigma_2$, *then:*

$$\mathit{SN}(\mathrm{D}) \text{ if and only if } \mathit{SN}(\mathrm{D}_1) \text{ and } \mathit{SN}(\mathrm{D}_2).$$

Proof: Straightforward. ■

6 Strong normalization

In this section, we will prove that derivations in the restricted type assignment system are strongly normalizable with respect to the notion of reduction defined in the previous section; for this we will use the well-known method of Computability Predicates [20].

Definition 6.1 (*Computability predicate*) *i)* Let B be a basis, $t \in T(\mathcal{C}, \mathcal{X})$, and σ a type. We define $\mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma)$ recursively on σ by:

a) $\mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\varphi) \iff \mathit{SN}(\mathrm{D})$.

b) $\mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma{\rightarrow}\tau) \iff$
$(\mathit{Comp}\,(\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\sigma) \Rightarrow \mathit{Comp}\,(\langle \mathrm{D}, \mathrm{D}', {\rightarrow}E\rangle :: \Pi\{B, B'\} \vdash^{\mathrm{r}}_{\mathcal{E}} tu{:}\tau))$.

c) $\mathit{Comp}\,(\langle \mathrm{D}_1, \ldots, \mathrm{D}_n, {\cap}I\rangle :: \Pi\{B_1, \ldots, B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1{\cap}\cdots{\cap}\sigma_n) \iff$
$\forall 1 \leq i \leq n\,[\mathit{Comp}\,(\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_i)]$.

ii) We say that a term-substitution R is *computable in a basis* B if, for every $x{:}\sigma \in B$, there are basis B_x and derivation D_x such that $\mathit{Comp}\,(\mathrm{D}_x :: B_x \vdash^{\mathrm{r}}_{\mathcal{E}} x^{\mathrm{R}}{:}\sigma)$.

Notice that $\mathit{Comp}\,(\langle {\cap}I\rangle :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\omega)$ holds for all t by *(i.c)* when $n = 0$.

We will prove that *Comp* satisfies the standard properties of computability predicates.

Lemma 6.2 *i)* $\mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma) \Rightarrow \mathit{SN}(\mathrm{D})$.

ii) $\mathit{SN}(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} xt_1\cdots t_m{:}\sigma) \Rightarrow \mathit{Comp}\,(\mathrm{D})$.

Proof: By simultaneous induction on the structure of types. The case $\sigma = \varphi$ is immediate, $\sigma = \sigma_1{\cap}\cdots{\cap}\sigma_n$ follows from Definition *6.1 (i.c)* and Lemma *5.5 (v)*, and for $\sigma = \alpha{\rightarrow}\beta$:

i) Let x be a variable not appearing in B and t.

$$\begin{array}{rcl}
\{x{:}\alpha\} \vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\alpha \;\&\; \mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\alpha{\rightarrow}\beta) & \Rightarrow & (\mathrm{IH}\,(ii)) \\
\mathit{Comp}\,(\mathrm{D}' :: \{x{:}\alpha\} \vdash^{\mathrm{r}}_{\mathcal{E}} x{:}\alpha) \;\&\; \mathit{Comp}\,(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\alpha{\rightarrow}\beta) & \Rightarrow & (6.1\,(i.b)) \\
\mathit{Comp}\,(\mathrm{D}'' = \langle \mathrm{D}, \mathrm{D}', {\rightarrow}E\rangle :: B, x{:}\alpha \vdash^{\mathrm{r}}_{\mathcal{E}} tx{:}\beta) & \Rightarrow & (\mathrm{IH}\,(i)) \\
\mathit{SN}(\mathrm{D}'') & \Rightarrow & (5.5\,(ii)) \\
\mathit{SN}(\mathrm{D}). & &
\end{array}$$

ii)

$$\begin{array}{rcl}
\mathit{SN}(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} xt_1\cdots t_m{:}\alpha{\rightarrow}\beta) & \Rightarrow & (\mathrm{IH}\,(i)) \\
(\mathit{Comp}\,(\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} u{:}\alpha) \Rightarrow \mathit{SN}(\mathrm{D}) \;\&\; \mathit{SN}(\mathrm{D}')) & \Rightarrow & (5.5\,(iii)) \\
(\mathit{Comp}\,(\mathrm{D}') \Rightarrow \mathit{SN}(\langle \mathrm{D}, \mathrm{D}', {\rightarrow}E\rangle :: \Pi\{B, B'\} \vdash^{\mathrm{r}}_{\mathcal{E}} xt_1\cdots t_m u{:}\beta)) & \Rightarrow & (\mathrm{IH}\,(ii)) \\
(\mathit{Comp}\,(\mathrm{D}') \Rightarrow \mathit{Comp}\,(\langle \mathrm{D}, \mathrm{D}', {\rightarrow}E\rangle)) & \Rightarrow & (6.1\,(i.b)) \\
\mathit{Comp}\,(\mathrm{D}). & &
\end{array}$$

■

We will now come to the term-substitution theorem, the final construction in the proof of our strong normalization result, for which we need the following ordering:

Definition 6.3 *i)* $\rhd$ stands for the well-founded encompassment ordering: $u \rhd v$ if $u \neq v$ modulo renaming of variables, and $v^{\mathrm{R}} = u|_p$ for some position p in u and term-substitution R.

ii) We define the ordering $\gg$ on pairs – consisting of a natural number and a term – as the object $(>_{\mathbb{N}}, \rhd)_{lex}$, where *lex* denotes *lexicographic extension.*

iii) Given a term t and a term-substitution R, the *interpretation* $\mathcal{I}(t^{\mathrm{R}})$ of t^{R} is defined as the pair $\langle n, t\rangle$ where n is the number of combinators appearing in t.

Note that the encompassment ordering contains the strict superterm relation (denoted by $\rhd$).

We can now prove the term-substitution theorem.

Theorem 6.4 *If* $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ *and* R *is computable in* B, *then there exists a* $\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\mathrm{R}}{:}\sigma$ *such that Comp* (D').

Proof: We will consider the interpretation of t^{R}, and prove the theorem by Nötherian induction on $\gg$ (which is well-founded). If t is a variable, then $B = \{x{:}\sigma\}$, and since R is assumed to be computable in B, there exists a D' such that *Comp* $(\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} x^{\mathrm{R}}{:}\sigma)$. Also, the case $\sigma = \omega$ is trivially computable. So, without loss of generality, we can assume that t is not a variable (so neither is t^{R}). Also, if $\sigma = \sigma_1\cap\cdots\cap\sigma_n$, then the last rule applied is $(\cap I)$, and we can reason on each σ_i separately, so we can focus on the case where $\sigma \in \mathcal{T}_s$.

We distinguish the following cases for t^{R}:

(t^{R} *is neutral*): Then there are $x \in \mathcal{X}, t_1, \ldots, t_n$ $(n > 0)$ such that $t^{\mathrm{R}} = x\,t_1 \cdots t_n$; also t is neutral, so there exist $z \in \mathcal{X}$ and $u_1, \ldots, u_m$ $(m > 0)$ such that $t = z\,u_1 \cdots u_m$, and $z^{\mathrm{R}} = x\,t_1 \cdots t_k$ $(k \geq 0, k + m = n)$. Since $B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, there exist $\sigma_1, \ldots, \sigma_m, B_1, \ldots, B_m, \mathrm{D}_1, \ldots, \mathrm{D}_m$ such that

$$\mathrm{D}_0 :: \{z{:}\sigma_1\to\cdots\to\sigma_m\to\sigma\} \vdash^{\mathrm{r}}_{\mathcal{E}} z{:}\sigma_1\to\cdots\to\sigma_m\to\sigma, \text{ and } \mathrm{D}_j :: B_j \vdash^{\mathrm{r}}_{\mathcal{E}} u_j{:}\sigma_j,$$

for every $1 \leq j \leq m$, and $B = \Pi\{B_1, \ldots, B_m\}$. Since $\mathcal{I}(t^{\mathrm{R}}) \gg \mathcal{I}(u_j{}^{\mathrm{R}})$, by induction, there exist D'_j such that *Comp* $(\mathrm{D}'_j :: B'_j \vdash^{\mathrm{r}}_{\mathcal{E}} u_j{}^{\mathrm{R}}{:}\sigma_j)$, for every $1 \leq j \leq m$. Also, since R is computable in B, there exists D'_0 such that *Comp* $(\mathrm{D}'_0 :: B'_0 \vdash^{\mathrm{r}}_{\mathcal{E}} z^{\mathrm{R}}{:}\sigma_1\to\cdots\to\sigma_m\to\sigma)$. Then, by Definition *6.1 (i.b)*,

$$\mathit{Comp}\,(\langle\cdots\langle \mathrm{D}'_0, \mathrm{D}'_1, \to\! E\rangle, \cdots, \mathrm{D}'_n, \to\! E\rangle :: \Pi\{B'_0, B'_1, \ldots, B'_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\mathrm{R}}{:}\sigma).$$

(t^{R} *is not neutral*): Then there are $C \in \mathcal{C}, t_1, \ldots, t_n$ $(n \geq 0)$ such that $t^{\mathrm{R}} = C\,t_1 \cdots t_n$. Now, three cases are possible:

a) $t = z s_1 \ldots s_m$ $(m \leq n)$, or $t = C s_1 \cdots s_n$, and at least one of the s_i is not a variable. Since $\mathcal{I}(t^{\mathrm{R}}) \gg \mathcal{I}(s_i{}^{\mathrm{R}})$, by induction $s_i{}^{\mathrm{R}}$ is computable, for every $1 \leq i \leq m$, or $1 \leq i \leq n$, respectively. Let y be a fresh variable, and $\mathrm{R}' = \mathrm{R} \cup \{y \mapsto s_i{}^{\mathrm{R}}\}$. Then $t^{\mathrm{R}} = (t[y]_i)^{\mathrm{R}'}$, and $\mathcal{I}(t^{\mathrm{R}}) \gg \mathcal{I}((t[y]_i)^{\mathrm{R}'})$. Then t^{R} is computable by induction.

b) $t = z z_1 \cdots z_m$ $(m \leq n)$. Then $z^{\mathrm{R}} = C t_1 \cdots t_k$ $(k + m = n)$. In this case we can proceed as for the case that t^{R} is neutral.

c) $t = C z_1 \cdots z_n$.

$(n \neq 0)$: Then $\mathcal{I}(t^{\mathrm{R}}) \gg \mathcal{I}(C^{\mathrm{R}})$, and $\mathrm{D}_0 :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma_1\to\cdots\to\sigma_n\to\sigma$, for certain $\sigma_1, \ldots, \sigma_n$, and, by induction, *Comp* $(\mathrm{D}_0 :: \emptyset \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma_1\to\cdots\to\sigma_n\to\sigma)$. Since R is computable in B, for every $1 \leq i \leq n$ there is D_i such that *Comp* $(\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} z_i{}^{\mathrm{R}}{:}\sigma_i)$, so by Definition *6.1 (i.b)*, also *Comp* $(\langle\cdots\langle \mathrm{D}_0, \mathrm{D}_1, \to\! E\rangle\cdots, \mathrm{D}_n, \to\! E\rangle :: \Pi\{B_1, \ldots, B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t^{\mathrm{R}}{:}\sigma)$.

$(n = 0)$: Then $B = \emptyset$ and $\mathrm{D} = \langle\mathcal{E}\rangle$. Let $\sigma = \sigma_1\to\cdots\to\sigma_n\to\varphi$; in order to prove that there exists a D' such that *Comp* $(\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} C{:}\sigma)$ it is sufficient to prove

$$\begin{array}{l} \forall 1 \leq i \leq n\ \exists \mathrm{D}_i\ [\mathit{Comp}\,(\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} u_i{:}\sigma_i)] \Rightarrow \\ \qquad \mathit{Comp}\,(\langle\cdots\langle \mathrm{D}, \mathrm{D}_1, \to\! E\rangle, \cdots, \mathrm{D}_n, \to\! E\rangle :: \Pi\{B_1, \ldots, B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} C u_1 \cdots u_n{:}\varphi). \end{array}$$

Take $\mathrm{D}_0 = \langle\cdots\langle \mathrm{D}, \mathrm{D}_1, \to\! E\rangle\cdots, \mathrm{D}_n, \to\! E\rangle$, then by Definition *6.1 (i.a)* it suffices to prove

$$\forall 1 \leq i \leq n\ \exists \mathrm{D}_i [\mathit{Comp}\,(\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} u_i{:}\sigma_i)] \Rightarrow \mathit{SN}\,(\mathrm{D}_0).$$

We will proceed by induction on the sum of the maximal lengths of the reduction paths on the derivations $\mathrm{D}_i :: B \vdash^{\mathrm{r}}_{\mathcal{E}} u_i{:}\sigma_i$ to their normal forms (notice that these derivations are

strongly normalizable by Lemma *6.2 (i)*, since they are computable).
Consider all possible rewrite steps out of D_0.

A) $\mathrm{D}_0 \rightarrow_{\mathcal{D}} \mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\varphi$ at the outermost level. Then there are $C, u_1, \ldots, u_n$ and variables $x_1, \ldots, x_n$ such that $t = C u_1 \cdots u_n$, the rule applied is $C x_1 \cdots x_i \rightarrow r$, so $i = \mathit{arity}(C), x_{i+1}, \ldots, x_n$ are fresh term-variables, and R_1 is the term-substitution $\{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$. Since *Comp* $(\mathrm{D} :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} u_i{:}\sigma_i)$ for all $1 \leq i \leq n$, R_1 is computable in $\{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}$. Then $\mathcal{I}((C x_1 \cdots x_n)^{\mathrm{R}}) \gg \mathcal{I}((r\, x_{i+1} \cdots x_n)^{\mathrm{R}_1})$, so by induction *Comp* $(\mathrm{D}' :: B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\varphi)$, and *SN* (D') by Definition *6.1 (i.a)*.

B) $\mathrm{D}_0 \rightarrow_{\mathcal{D}} \mathrm{D}' :: \Pi\{B', B'_1, \ldots, B'_m\} \vdash^{\mathrm{r}}_{\mathcal{E}} C u'_1 \cdots u'_n{:}\varphi$, and the reduction took place inside one of the u_i, then by induction *SN* (D').

So, for all D' such that $\mathrm{D} \rightarrow_{\mathcal{D}} \mathrm{D}'$, we have proved *SN* (D'), so, in particular, *SN* (D). ■

The main result of this section then is the strong normalization theorem for derivation reduction in $\vdash^{\mathrm{r}}_{\mathcal{E}}$.

Theorem 6.5 (Strong normalisation of derivation reduction) *If* $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, *then SN* (D).

Proof: If $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, then, taking R such that $x^{\mathrm{R}} = x$, by Theorem *6.4*, *Comp* $(\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma)$. Then, by Lemma *6.2 (i)*, *SN* (D). ■

7 Approximants

Now we will develop, essentially following [22] (see also [6]), a notion of approximant for combinator terms. This will be done by introducing a special symbol $\perp$ into the definition of terms. The general idea is that a term a directly approximates a term t if they are identical but for those places where a has an occurrence of $\perp$.

Definition 7.1 (*Combinator terms with* $\perp$) *i)* The set $T(\mathcal{C},\mathcal{X},\perp)$ is defined by:

$$t ::= \perp \mid x \mid C \mid \mathit{Ap}\,(t_1, t_2)$$

ii) The notion of rewriting of Definition *1.4* extends naturally to terms in $T(\mathcal{C},\mathcal{X},\perp)$, and we will use the same symbol '$\rightarrow_{\mathbf{R}}$' to denote the rewriting relation induced by $(\Sigma, \mathbf{R})$ on $T(\mathcal{C},\mathcal{X},\perp)$.

The relation $\sqsubseteq$ on terms, as given in the following definition, takes $\perp$ to be the smallest term.

Definition 7.2 *i)* We define the relation $\sqsubseteq$ on $T(\mathcal{C},\mathcal{X},\perp)$ inductively by:

$$\begin{array}{rcl} \perp & \sqsubseteq & t, \\ t & \sqsubseteq & t, \\ t_1 \sqsubseteq u_1 \ \&\ t_2 \sqsubseteq u_2 & \Longleftrightarrow & t_1 t_2 \sqsubseteq u_1 u_2. \end{array}$$

ii) t and u are called *compatible* if there exists a v such that $t \sqsubseteq v$ and $u \sqsubseteq v$.

Definition 7.3 (*Approximate normal forms*) *i)* $\mathcal{A}_{\mathcal{C}}$, the set of *approximate normal forms* of $T(\mathcal{C},\mathcal{X},\perp)$, ranged over by a, is inductively defined by:

$$a ::= \perp \mid x a_1 \cdots a_n \ (n \geq 0) \mid C a_1 \cdots a_n \ (n < \mathit{arity}(C)).$$

ii) $\mathcal{DA}\,(t)$, *the direct approximant of* t with respect to $(\Sigma, \mathbf{R})$ is defined by:

$$
\begin{aligned}
\mathcal{DA}(x) &= x \\
\mathcal{DA}(C) &= C \\
\mathcal{DA}(t_1 t_2) &= \bot, \text{ if } \mathcal{DA}(t_1) = \bot \text{ or} \\
&\qquad\qquad \mathcal{DA}(t_1) = C a_1 \cdots a_n, \text{ and } \mathit{arity}(C) = n{+}1 \\
&= \mathcal{DA}(t_1)\,\mathcal{DA}(t_2), \text{ otherwise}
\end{aligned}
$$

Notice that every normal form in $T(\mathcal{C},\mathcal{X})$ is also an approximate normal form.

For $\sqsubseteq$, the following properties hold:

Lemma 7.4 *i)* $t \sqsubseteq u \sqsubseteq v \;\Rightarrow\; t \sqsubseteq v$.
ii) t *is a head-normal form* $\iff \exists a \in \mathcal{A}_{\mathcal{C}}\,[a \sqsubseteq t \;\&\; a \neq \bot]$.
iii) If $a \in \mathcal{A}_{\mathcal{C}}$ *and* $a \sqsubseteq t$, *then* $a \sqsubseteq \mathcal{DA}(t)$.
Proof: By induction on the definition of $\sqsubseteq$. ■

The relation between reduction and $\sqsubseteq$ is expressed by:

Lemma 7.5 *i)* $a \in \mathcal{A}_{\mathcal{C}} \;\&\; v \to^* w \;\&\; a \sqsubseteq v \;\Rightarrow\; a \sqsubseteq w$.
ii) $t_0 \sqsubseteq t \;\&\; t_0 \to t_1 \;\Rightarrow\; \exists t'[t \to t' \;\&\; t_1 \sqsubseteq t']$.
Proof: By induction on the structure of terms. ■

We will now introduce a notion of 'join' on terms containing $\bot$, that is of use in the proof of Lemma *8.1*.

Definition 7.6 On $T(\mathcal{C},\mathcal{X},\bot)$, the partial mapping $\sqcup : T(\mathcal{C},\mathcal{X},\bot) \times T(\mathcal{C},\mathcal{X},\bot) \to T(\mathcal{C},\mathcal{X},\bot)$ is defined by:

$$
\begin{aligned}
\bot \sqcup t = t \sqcup \bot &= t \\
x \sqcup x &= x \\
C \sqcup C &= C \\
(t_1 t_2) \sqcup (u_1 u_2) &= (t_1 \sqcup u_1)(t_2 \sqcup u_2)
\end{aligned}
$$

The last alternative defines the join on applications in a more general way than that of [15], which would state that $(t_1 t_2) \sqcup (u_1 u_2) \sqsubseteq (t_1 \sqcup u_1)(t_2 \sqcup u_2)$, since it is not always sure if a join of two arbitrary terms exists. However, we will use our more general definition only on terms that are compatible, so the conflict is only apparent. So, when we write a term as $v \sqcup u$, we assume v and u to be compatible.

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

Lemma 7.7 *If* $t_1 \sqsubseteq t$ *and* $t_2 \sqsubseteq t$, *then* $t_1 \sqcup t_2$ *is defined, and:* $t_1 \sqsubseteq t_1 \sqcup t_2$, $t_2 \sqsubseteq t_1 \sqcup t_2$, *and* $t_1 \sqcup t_2 \sqsubseteq t$.
Proof: By induction on the structure of terms. ■

Approximants of terms are defined by:

Definition 7.8 (*Approximants*) $\mathcal{A}_{\mathcal{C}}(t) = \{a \in \mathcal{A}_{\mathcal{C}} \mid \exists u\,[t \to^* u \;\&\; a \sqsubseteq u]\}$ is the *set of approximants of* t.

In Section 9, using this definition, we will define a semantics for CS, and we will need the following properties relating approximants and reduction.

Lemma 7.9 *i)* $t \rightarrow^* t' \Rightarrow \mathcal{A}_{\mathcal{C}}(t) = \mathcal{A}_{\mathcal{C}}(t')$.
ii) $a, a' \in \mathcal{A}_{\mathcal{C}}(t) \Rightarrow a \sqcup a' \in \mathcal{A}_{\mathcal{C}}(t)$.

Proof: *i)* $\subseteq$)

$$\begin{array}{rcl} t \rightarrow^* t' \ \&\ a \in \mathcal{A}_{\mathcal{C}}(t) & \Rightarrow & \\ t \rightarrow^* t' \ \&\ \exists v\, [t \rightarrow^* v \ \&\ a \sqsubseteq v] & \Rightarrow & (1.6) \\ \exists v, w\, [t \rightarrow^* v \ \&\ v \rightarrow^* w \ \&\ t' \rightarrow^* w \ \&\ a \sqsubseteq v] & \Rightarrow & (7.5(i)) \\ \exists w\, [t' \rightarrow^* w \ \&\ a \sqsubseteq w] & \Rightarrow & a \in \mathcal{A}_{\mathcal{C}}(t'). \end{array}$$

$\supseteq$) $a \in \mathcal{A}_{\mathcal{C}}(t') \Rightarrow \exists v\, [t' \rightarrow^* v \ \&\ a \sqsubseteq v] \Rightarrow \exists v\, [t \rightarrow^* v \ \&\ a \sqsubseteq v] \Rightarrow a \in \mathcal{A}_{\mathcal{C}}(t)$.

ii)

$$\begin{array}{rcl} a \in \mathcal{A}_{\mathcal{C}}(t) \ \&\ a' \in \mathcal{A}_{\mathcal{C}}(t) & \Rightarrow & (7.8) \\ \exists u, u'\, [t \rightarrow^* u \ \&\ a \sqsubseteq u \ \&\ t \rightarrow^* u' \ \&\ a' \sqsubseteq u'] & \Rightarrow & (1.6 \ \&\ 7.5(i)) \\ \exists u, u', v\, [t \rightarrow^* u \rightarrow^* v \ \&\ t \rightarrow^* u' \rightarrow^* v \ \&\ a \sqsubseteq v \ \&\ a' \sqsubseteq v] & \Rightarrow & (7.7) \\ \exists v\, [t \rightarrow^* v \ \&\ a \sqcup a' \sqsubseteq v] & \Rightarrow & a \sqcup a' \in \mathcal{A}_{\mathcal{C}}(t). \blacksquare \end{array}$$

Lemma 7.10 *If* $\mathcal{A}_{\mathcal{C}}(t) = \{\perp\}$*, then* t *is unsolvable.*
Proof: If $\mathcal{A}_{\mathcal{C}}(t) = \{\perp\}$, then, for all v such that $t \rightarrow^* v$, and $a \in \mathcal{A}_{\mathcal{C}}$, if $a \sqsubseteq v$, then $a = \perp$. So, in particular, there is no v such that $t \rightarrow^* v$ and v is of the shape $x\, a_1 \cdots a_n$, with $(n \geq 0)$ or $C\, a_1 \cdots a_n$ with $(n < \textit{arity}\,(C))$, since otherwise $x \perp \cdots \perp \sqsubseteq v$ or $C \perp \cdots \perp \sqsubseteq v$. Therefore, t does not reduce to a term in head normal form (it is unsolvable). ■

The following result is crucial for the proof of Lemma *9.4*:

Lemma 7.11 *If, for* $t_1, t_2 \in T(\mathcal{C}, \mathcal{X})$ *and* $a \in \mathcal{A}_{\mathcal{C}}$*, there exists* u *such that* $t_1\, t_2 \rightarrow^* u$ *and* $a \sqsubseteq u$*, then there exist* $a_1 \in \mathcal{A}_{\mathcal{C}}(t_1)$*,* $a_2 \in \mathcal{A}_{\mathcal{C}}(t_2)$ *and* u' *such that* $a_1\, a_2 \rightarrow^* u'$ *and* $a \sqsubseteq u'$*.*
Proof: The case $a = \perp$ is trivial. For $a \neq \perp$: if $t_1\, t_2 \rightarrow^* u$, then either:

i) $u = u_1\, u_2$, and $t_j \rightarrow^* u_j$, for $j = 1, 2$. Since $a \sqsubseteq u_1\, u_2$, there are a_1, a_2 such that $a = a_1\, a_2$, and $a_j \sqsubseteq u_j$, for $j = 1, 2$. Notice that $a_1\, a_2 \in \mathcal{A}_{\mathcal{C}}$, and take $u' = a$.

ii) There exist $C, p_1, \ldots, p_n$ such that $C\, x_1 \cdots x_n \rightarrow r$,

$$t_1\, t_2 \rightarrow^* C\, p_1 \cdots p_n \rightarrow r^{\overline{p}} \rightarrow^* u,$$

and none of the reductions in the first part of this sequence take place at the root position. Since some of the reductions that take place after contracting the redex $C\, p_1 \cdots p_n$ are in fact residuals of redexes already occurring in $p_1, \ldots, p_n$, we can take the reduction sequence that first contracts all relevant redexes (and their residuals) occurring in $p_1, \ldots, p_n$. Then, since the rewrite system is orthogonal (i.e. rules are left linear and without superpositions), there exists $p'_1, \ldots, p'_n$ and v such that

$$t_1\, t_2 \rightarrow^* C\, p_1 \cdots p_n \rightarrow^* C\, p'_1 \cdots p'_n \rightarrow r^{\overline{p}'} \rightarrow^* v \text{ and } u \rightarrow^* v$$

such that in the reduction sequence $r^{\overline{p}'} \rightarrow^* v$ only redexes are contracted that are created *after* the redex $C\, p'_1 \cdots p'_n$ was contracted. Take $a_i = \mathcal{DA}\,(p'_i)$, for $1 \leq i \leq n$, then the redexes that are erased have no relevance to the sequence $r^{\overline{p}'} \rightarrow^* v$; moreover, there is only one redex in $C\, a_1 \cdots a_n$, being that term itself, and both $C\, a_1 \cdots a_{n-1}$ and a_n are in $\mathcal{A}_{\mathcal{C}}$. Notice that $t_1 \rightarrow^* C\, p'_1 \cdots p'_{n-1}$, and $C\, a_1 \cdots a_{n-1} \sqsubseteq C\, p'_1 \cdots p'_{n-1}$, and that $t_2 \rightarrow^* p'_n$ and $a_n \sqsubseteq p'_n$.
We now focus on the reduction sequence

$$C\, p'_1 \cdots p'_n \rightarrow r^{\overline{p}'} \rightarrow^* v$$

Notice that, by the construction sketched above, only redexes that are newly created are contracted, and that any redex created in this sequence corresponds to a redex being created for a sequence starting with $C\, a_1 \cdots a_n$, therefore

$$C\, a_1 \cdots a_n \rightarrow r^{\overline{a}} \rightarrow^* u',$$

and each term created in this reduction is smaller than (in the sense of $\sqsubseteq$) the corresponding term in the reduction sequence above (hence $u' \sqsubseteq v$), and each redex in u' corresponds to a redex in v. Take $a' = \mathcal{DA}(v)$, then $a' \sqsubseteq v$, and all redexes are masked by $\perp$. Since $u' \sqsubseteq v$ by masking all the 'old' redexes, we also have that $a' = \mathcal{DA}(u')$. Since $a \sqsubseteq u$, also $a \sqsubseteq v$ and therefore $a \sqsubseteq a'$. We then deduce $a \sqsubseteq u'$. ∎

To come to a notion of type assignment on $T(\mathcal{C},\mathcal{X},\perp)$, the definition of type assignment as given in Definitions *2.13* and *4.1* need *not* be changed, it suffices that the terms are allowed to be in $T(\mathcal{C},\mathcal{X},\perp)$. In particular, $\mathcal{E}$ does not produce a type for $\perp$; since $\perp \notin \mathcal{C}$, and because of Definition *2.13*, this implies that $\perp$ can only appear in (sub)terms that are typed with ω.

The following property is needed in the proof of Theorem *8.5*:

Lemma 7.12 *If $B \vdash_{\mathcal{E}} t{:}\sigma$, where B,σ are ω-free, and t is combinator-free, then t is $\perp$-free.*

Proof: By induction on t. We consider all possible cases:

($t = \perp t_1 \cdots t_n,\, n \geq 0$): Impossible, since $\sigma \neq \omega$.

($t = x\, t_1 \cdots t_n,\, n \geq 0$): Without loss of generality, we can assume $\sigma \in \mathcal{T}_s$. Then $B \vdash_{\mathcal{E}} x{:}\sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \sigma$, and $B \vdash_{\mathcal{E}} t_i{:}\sigma_i$, for $1 \leq i \leq n$. Therefore, there are $\sigma'_1,\ldots,\sigma'_{n+1}$ such that $x{:}\sigma'_1 \rightarrow \ldots \rightarrow \sigma'_n \rightarrow \sigma'_{n+1} \in B$, all $\sigma'_1,\ldots,\sigma'_{n+1}$ are ω-free, $\sigma_i \leq \sigma'_i$ for $1 \leq i \leq n$, and $\sigma'_{n+1} \leq \sigma$. Then, by Lemma *3.2(i)*, $B \vdash_{\mathcal{E}} t_i{:}\sigma'_i$, for $1 \leq i \leq n$. Then, by induction, t_i does not contain $\perp$, for $1 \leq i \leq n$. ∎

In Lemma *8.1*, we will need the following result.

Lemma 7.13 *i) If* D :: $B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, *and* $t \sqsubseteq v$, *then also* D$'$ :: $B \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma$, *where D' has the same structure as* D.

ii) If D :: $B \vdash_{\mathcal{E}} t{:}\sigma$, *and* $t \sqsubseteq v$, *then also* D$'$ :: $B \vdash_{\mathcal{E}} v{:}\sigma$.

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

($\rightarrow$*E*): D $= \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\rho\rightarrow\tau, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\rho, \rightarrow E\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\tau$. Then there are $v_1 \sqsupseteq t_1, v_2 \sqsupseteq t_2$ such that $v = v_1 v_2$, and, $\mathrm{D}'_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} v_1{:}\rho\rightarrow\tau$ and $\mathrm{D}'_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} v_2{:}\rho$ by induction. Therefore there exists $D' = \langle \mathrm{D}'_1, \mathrm{D}'_2, \rightarrow E\rangle :: \Pi\{B_1,B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} v_1 v_2{:}\tau$, which has the same structure as D.

($\cap$*I*): D $= \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1,\ldots,\mathrm{D}_n :: B_n \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_n, \cap I\rangle :: \Pi\{B_1,\ldots,B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1\cap\cdots\cap\sigma_n$, with $n \geq 0$. Then, by induction, for $1 \leq i \leq n$, $\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_i$, so also

$$\langle \mathrm{D}_1,\ldots,\mathrm{D}_n,\cap I\rangle :: \Pi\{B_1,\ldots,B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma_1\cap\cdots\cap\sigma_n.$$

Notice that the only interesting case is hidden in the last part: $n = 0$. Then, in particular, t can be $\perp$, and v can be any term. The cases (*Ax*) and $(\mathcal{E})$ are immediate.

ii) If D :: $B \vdash_{\mathcal{E}} t{:}\sigma$, then, by Lemma *4.2(ii)*, there is a $B' \geq B$ such that D$'$ :: $B' \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$. Since $t \sqsubseteq v$, by the first part also D$'$ :: $B' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma$. Then also D$'$ :: $B \vdash_{\mathcal{E}} v{:}\sigma$. ∎

8 Approximation and normalization

The approximation result that will be proved in this section has been reached also in [3] for the essential system for LC, $\vdash_{\lambda\cap}$. That result, however, cannot be transferred to typed CS, and neither can the there used technique. The crucial point in the problem is that the property

'there is an $A \in \mathcal{A}(Mz)$ such that $B, z{:}\alpha \vdash_{\lambda\cap} A{:}\beta$'
implies
'there is an $A \in \mathcal{A}(M)$ such that $B \vdash_{\lambda\cap} A{:}\alpha\rightarrow\beta$'.

when z does not occur in M, is relatively easy to prove, since the following holds:

$$\textit{If } A \in \mathcal{A}(Mz) \textit{ and } z \notin FV(M)\textit{, then either:}$$
$$A \equiv A'z \textit{ with } z \notin FV(A') \textit{ and } A' \in \mathcal{A}(M)\textit{, or } \lambda z.A \in \mathcal{A}(M).$$

The first of these properties is hard to prove in arbitrary CS, because there is no known way to express abstraction adequately in CS that are not combinatory complete. Moreover, even in combinatory complete systems like CL, using the existence of a bijection through the mappings $\langle\ \rangle_\lambda$ and $[\![\ ]\!]_{\mathrm{CL}}$, it is not possible to prove this first property using the second. Take, for example, the term $\mathbf{SK}y$, $B = \{z{:}\alpha\}$, and

$$\mathit{Ch} \ = [(\varphi_1 \mapsto \alpha), (\varphi_2 \mapsto \omega), (\varphi_3 \mapsto \alpha), (\varphi_4 \mapsto \alpha), (\varphi_5 \mapsto \alpha)]$$

then we can derive the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathit{Ch}\,(\mathcal{E}_{\mathrm{CL}}(\mathbf{S})) = (\alpha\to\omega\to\alpha)\to\omega\to\alpha\to\alpha}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathbf{S}{:}(\alpha\to\omega\to\alpha)\to\omega\to\alpha\to\alpha} \quad \dfrac{\mathit{Ch}\,(\mathcal{E}_{\mathrm{CL}}(\mathbf{K})) = \alpha\to\omega\to\alpha}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathbf{K}{:}\alpha\to\omega\to\alpha}}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathbf{SK}{:}\omega\to\alpha\to\alpha} \quad \dfrac{}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} y{:}\omega}}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathbf{SK}y{:}\alpha\to\alpha} \quad \dfrac{z{:}\alpha \in B}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} z{:}\alpha}}{B \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathbf{SK}yz{:}\alpha}$$

Notice that $\mathcal{A}_{\mathrm{CL}}(\mathbf{SK}yz) = \{\bot, z\}$ and also $\{z{:}\alpha\} \vdash_{\mathcal{E}_{\mathrm{CL}}} z{:}\alpha$. Following the above property, since none of the approximants of $\mathbf{SK}yz$ is an application term, we would then like to obtain something like $[\![\lambda z.\langle z\rangle_\lambda]\!]_{\mathrm{CL}} \in \mathcal{A}_{\mathrm{CL}}(\mathbf{SK}y)$ and $\emptyset \vdash_{\mathcal{E}_{\mathrm{CL}}} [\![\lambda z.\langle z\rangle_\lambda]\!]_{\mathrm{CL}}{:}\alpha\to\alpha$. However,

$$[\![\lambda z.\langle z\rangle_\lambda]\!]_{\mathrm{CL}} = \mathbf{I} \text{ and } \mathcal{A}_{\mathrm{CL}}(\mathbf{SK}y) = \{\bot, \mathbf{S}\bot\bot, \mathbf{SK}\bot, \mathbf{S}\bot y, \mathbf{SK}y\}.$$

This problem is overcome in this paper using the strong normalization result proved in the previous section for derivation reduction in $\vdash^{\mathrm{r}}_{\mathcal{E}}$.

We will need the following intermediate result.

Lemma 8.1 *If* $\mathrm{D} :: B \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$ *is in normal form with respect to* $\to_{\mathcal{D}}$*, then there exists an* $a \in \mathcal{A}_{\mathcal{C}}$ *such that* $a \sqsubseteq t$ *and* $\mathrm{D}' :: B \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma$.

Proof: By induction on the structure of derivations.

($\to E$): Let $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t_1{:}\tau\to\sigma, \mathrm{D}_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} t_2{:}\tau, \to E\rangle :: \Pi\{B_1, B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} t_1 t_2{:}\sigma$. Then, by induction, there are $a_1 \sqsubseteq t_1, a_2 \sqsubseteq t_2$ in $\mathcal{A}_{\mathcal{C}}$ such that $\mathrm{D}'_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} a_1{:}\tau\to\sigma$, and $\mathrm{D}'_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} a_2{:}\tau$, and $\langle \mathrm{D}'_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} a_1{:}\tau\to\sigma, \mathrm{D}'_2 :: B_2 \vdash^{\mathrm{r}}_{\mathcal{E}} a_2{:}\tau, \to E\rangle :: \Pi\{B_1, B_2\} \vdash^{\mathrm{r}}_{\mathcal{E}} a_1 a_2{:}\sigma$. By Definition *7.2* we know that $a_1 a_2 \sqsubseteq t_1 t_2$.
Now $a_1 a_2 \notin \mathcal{A}_{\mathcal{C}}$ if there is a $C \in \mathcal{C}$ such that $a_1 = C a_1^1 \cdots a_1^{n-1}$ and $\mathit{arity}(C) = n$. But then there are $t_1^1, \ldots, t_1^{n-1}$ such that $t_1 = C t_1^1 \cdots t_1^{n-1}$, and $t = C t_1^1 \cdots t_1^{n-1} t_2$. In particular, by the remark after Definition *5.4*, D is reducible, which is impossible. So $a_1 a_2 \in \mathcal{A}_{\mathcal{C}}$.

($\cap I$): Let $\mathrm{D} = \langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1, \ldots, \mathrm{D}_n :: B_n \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_n, \cap I\rangle :: \Pi\{B_1, \ldots, B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma_1\cap\cdots\cap\sigma_n$. By induction, for $1 \le i \le n$, there is an $a_i \sqsubseteq t$ in $\mathcal{A}_{\mathcal{C}}$ such that $\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} a_i{:}\sigma_i$. Take now $a = a_1 \sqcup \cdots \sqcup a_n$. Since, for $1 \le i \le n$, $a_i \sqsubseteq a$, by Lemma *7.13* also $\mathrm{D}_i :: B_i \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma_i$, so we get $\langle \mathrm{D}_1 :: B_1 \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma_1, \ldots, \mathrm{D}_n :: B_n \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma_n, \cap I\rangle :: \Pi\{B_1, \ldots, B_n\} \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma_1\cap\cdots\cap\sigma_n$. Since $a_i \sqsubseteq t$ for all $1 \le i \le n$, by Lemma *7.7* also $a \sqsubseteq t$.

The cases $(\mathcal{E})$ and (*Ax*) are immediate.

Notice that the only real case lies hidden in part ($\cap I$): if $n = 0$, then $a = \bot$.

Theorem 8.2 (Approximation) *If* $B \vdash_{\mathcal{E}} t{:}\sigma$*, then there exists an* $a \in \mathcal{A}_{\mathcal{C}}(t)$ *such that* $B \vdash_{\mathcal{E}} a{:}\sigma$.

Proof: For every D such that D :: $B \vdash_{\mathcal{E}} t{:}\sigma$, there are D′ and B' such that D′ :: $B' \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$, and $B \leq B'$, by Lemma *4.2(ii)*. Then, by Theorem *6.5*, *SN* (D′). Let D″ :: $B'' \vdash^{\mathrm{r}}_{\mathcal{E}} v{:}\sigma$ be a normal form of D′ with respect to $\rightarrow_{\mathcal{D}}$, then by Lemma *8.1*, there is an $a \in \mathcal{A}_{\mathcal{C}}$ such that $a \sqsubseteq v$ and D‴ :: $B'' \vdash^{\mathrm{r}}_{\mathcal{E}} a{:}\sigma$. Then, by Lemma *5.5* (twice), $B' \leq B''$, and $t \rightarrow^* v$, so $a \in \mathcal{A}_{\mathcal{C}}(t)$. Also, by Lemma *4.2(i)* and Lemma *3.2(iii)*, $B \vdash_{\mathcal{E}} a{:}\sigma$. ■

For principal environments we can show that the converse of this result also holds.

Theorem 8.3 *Let* $((\mathcal{C},\mathcal{X}),\mathbf{R})$ *be a* CS*, and* $\mathcal{E}$ *be principal for* $\mathcal{C}$*. If there is an* $a \in \mathcal{A}_{\mathcal{C}}(t)$ *such that* $B \vdash_{\mathcal{E}} a{:}\sigma$*, then* $B \vdash_{\mathcal{E}} t{:}\sigma$*.*
Proof: If $a \in \mathcal{A}_{\mathcal{C}}(t)$ such that $B \vdash_{\mathcal{E}} a{:}\sigma$, then there exists a v such that $t \rightarrow^* v$ and $a \sqsubseteq v$. But then, by Lemma *7.13*, also $B \vdash_{\mathcal{E}} v{:}\sigma$. Since $\mathcal{E}$ is principal for $\mathcal{C}$, by Theorem *3.8*, also $B \vdash_{\mathcal{E}} t{:}\sigma$. ■

Theorem 8.4 (Head-normalisation) *Let* $t \in T(\mathcal{C},\mathcal{X})$*. If* $B \vdash_{\mathcal{E}} t{:}\sigma$*, and* $\sigma \neq \omega$*, then* t *has a head-normal form.*
Proof: If $B \vdash_{\mathcal{E}} t{:}\sigma$, then by Theorem *8.2*, there is an $a \in \mathcal{A}_{\mathcal{C}}(t)$ such that $B \vdash_{\mathcal{E}} a{:}\sigma$. Since $\sigma \neq \omega$, $a \neq \perp$, and since $a \in \mathcal{A}_{\mathcal{C}}$, there are x or C, and terms $a_1,\ldots,a_n$ such that $a = x\,a_1 \cdots a_n$, or $a = C\,a_1 \cdots a_n$ with *arity*$(C) < n$. Also, since $a \in \mathcal{A}_{\mathcal{C}}(t)$, there is a v such that $t \rightarrow v$ and $a \sqsubseteq v$. Since $a \sqsubseteq v$, there are $t_1,\ldots,t_n$ such that either $v = x\,t_1 \cdots t_n$, or $v = C\,t_1 \cdots t_n$, with *arity*$(C) < n$. But then v is in head-normal form, so t has a head-normal form. ■

The combinatorial equivalent of another well-known result for intersection type assignment in the LC, i.e. the property

If $B \vdash_{\mathcal{E}} t{:}\sigma$*, and* B,σ *are* ω*-free, then* t *has a normal form*

no longer holds. Take for example the CS

$$\begin{array}{lcl} \mathbf{Z}xy & \rightarrow & y, \\ \mathbf{D}x & \rightarrow & xx, \end{array}$$

then $\mathbf{Z}(\mathbf{DD})$ is typeable with a type not containing ω, but the term $\mathbf{Z}(\mathbf{DD})$ has no normal form. However, we can prove this result for the class of typeable non-Curryfied terms.

Theorem 8.5 (Normalisation) *Let* $t \in T_{NC}(\mathcal{C},\mathcal{X})$*. If* $B \vdash_{\mathcal{E}} t{:}\sigma$*, and* B,σ *are* ω*-free, then* t *has a normal form.*
Proof: By Theorem *8.2*, there is an $a \in \mathcal{A}_{\mathcal{C}}(t)$ such that $B \vdash_{\mathcal{E}} a{:}\sigma$. Notice that if $t \in T_{NC}(\mathcal{C},\mathcal{X})$, and t' is a reduct of t then also $t' \in T_{NC}(\mathcal{C},\mathcal{X})$. Therefore, a cannot contain any $C \in \mathcal{C}$. Then $a = x\,a_1 \cdots a_n$, where each a_i contains only variables and eventually $\perp$. But, by Lemma *7.12*, a does not contain $\perp$. Now, since $a \in \mathcal{A}_{\mathcal{C}}(t)$, there exists $v \in T(\mathcal{C},\mathcal{X})$ such that $t \rightarrow^* v$ and $a \sqsubseteq v$. Since a does not contain $\perp$, $v = a$, and since a is in normal form, t has a normal form. ■

We will now show that, using Theorem *6.5*, all terms typeable in the subsystem of $\vdash_{\mathcal{E}}$ that does not use ω ($\vdash^{\not\omega}_{\mathcal{E}}$), are strongly normalizable.

Lemma 8.6 *i) If* D *is a derivation in* $\vdash^{\not\omega}_{\mathcal{E}}$ *, and* D $\rightarrow_{\mathcal{D}}$ D′*, then also* D′ *is a derivation in* $\vdash^{\not\omega}_{\mathcal{E}}$ *.*
ii) D :: $B \vdash^{\not\omega}_{\mathcal{E}} t{:}\sigma \rightarrow_{\mathcal{D}}$ D′ :: $B' \vdash^{\not\omega}_{\mathcal{E}} t'{:}\sigma$*, if and only if* $t \rightarrow t'$*.*
Proof: By Definition *5.4* and Lemma *(ii)*. ■

Thus, in the type system $\vdash^{\not\omega}_{\mathcal{E}}$, $\rightarrow_{\mathcal{D}}$ mimics $\rightarrow$ and vice-versa. This observation immediately leads to the following result.

Theorem 8.7 Let $t \in T(\mathcal{C},\mathcal{X})$. If $B \vdash^{\omega}_{\mathcal{E}} t{:}\sigma$, then t is strongly normalizable.

Proof: Let D be such that D :: $B \vdash^{\omega}_{\mathcal{E}} t{:}\sigma$. Since also D :: $B \vdash_{\mathcal{E}} t{:}\sigma$, by Lemma *4.2(iii)*, there are D$'$, B' such that $B \leq B'$, and D$'$ :: $B' \vdash^{\mathrm{r}}_{\mathcal{E}} t{:}\sigma$. By Theorem *6.5*, D$'$ is strongly normalizable with respect to $\rightarrow_{\mathcal{D}}$. By Lemma *8.6(ii)*, all derivation redexes in D correspond to redexes in t and vice-versa, a property that is preserved under reduction. So also t is strongly normalizable. ∎

It is worthwhile to notice that, unlike for LC with $\vdash_{\lambda\cap}$, the reverse implication of the three theorems does not hold in general. For this, it is sufficient to note that a subject expansion theorem does not hold (see also the last remark of Section *3*).

Another aspect worth noting is that, unlike in LC, no longer every term in normal form is typeable without ω in basis and type. Take for example $t = \mathbf{S}(\mathbf{K}(\mathbf{SII}))(\mathbf{K}(\mathbf{SII}))$, and note that, by Property *2.18* every type assignable to t (regardless of the environment used) is a type assignable to $\lambda c.(\lambda x.xx)(\lambda x.xx)$ in $\vdash_{\lambda\cap}$. Since this last term has no head-normal form, only ω can be assigned to it.

9 Semantics

In this section, we will define two semantics for CS. The first is a filter model, where terms will be interpreted by the set of their assignable types; the second an approximation model, where terms will be interpreted by the set of their approximants.

Definition 9.1 (*Filters*) *i)* A subset d of $\mathcal{T}$ is a *filter* if and only if:

a) If $\sigma_1,\ldots,\sigma_n \in d$ $(n \geq 0)$, then $\sigma_1 \cap \cdots \cap \sigma_n \in d$.

b) If $\sigma \in d$ and $\sigma \leq \tau$, then $\tau \in d$.

ii) If V is a subset of $\mathcal{T}$, then $\uparrow V$ is the smallest filter that contains V, and $\uparrow\sigma = \uparrow\{\sigma\}$.

iii) $\mathcal{F} = \{d \subseteq \mathcal{T} \mid d \text{ is a filter}\}$.

Notice that a filter is never empty, since by part *(i.a)*, for all d, $\omega \in d$. $\langle \mathcal{F}, \subseteq \rangle$ is a cpo and henceforward it will be considered with the corresponding Scott topology.

Notice moreover that, by rule $(\cap I)$ and Theorem *3.3*, $\{\sigma \mid B \vdash_{\mathcal{E}} t{:}\sigma\} \in \mathcal{F}$.

Definition 9.2 *i)* Application on $\wp\mathcal{A}_{\mathcal{C}}$, $\cdot : \wp\mathcal{A}_{\mathcal{C}} \times \wp\mathcal{A}_{\mathcal{C}} \rightarrow \wp\mathcal{A}_{\mathcal{C}}$, is defined as follows:

$$A_1 \cdot A_2 \;=\; \{a \in \mathcal{A}_{\mathcal{C}} \mid \exists a_1 \in A_1, a_2 \in A_2, u\, [a_1\, a_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\}.$$

ii) Application on $\mathcal{F}$, $\cdot : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{F}$, is defined as follows:

$$d \cdot e \;=\; \uparrow\{\tau \mid \exists\, \sigma \in e\, [\sigma \rightarrow \tau \in d]\}.$$

We will define two interpretations of terms:

Definition 9.3 *i)* The interpretation of terms in the domain of approximants over $\mathcal{C}$ is defined as:
$[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = \mathcal{A}_{\mathcal{C}}(t) = \{a \in \mathcal{A}_{\mathcal{C}} \mid \exists u\, [t \rightarrow^* u \;\&\; a \sqsubseteq u]\}$.

ii) Let ξ be a valuation of term variables in $\mathcal{F}$; we write $B \models \xi$ if and only if, for all $x{:}\sigma \in B$, $\sigma \in \xi(x)$. $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$, the interpretation of terms in $\mathcal{F}$ via ξ and $\mathcal{E}$ is defined by:
$[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = \{\sigma \mid \exists B\, [B \models \xi \;\&\; B \vdash_{\mathcal{E}} t{:}\sigma]\}$.

Both applications are well-defined, in the sense that they respect application on terms.

Lemma 9.4 i) $[\![t_1]\!]^{\mathcal{A}}_{\mathcal{C}} \cdot [\![t_2]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![t_1\, t_2]\!]^{\mathcal{A}}_{\mathcal{C}}$.

ii) $[\![t_1]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} \cdot [\![t_2]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![t_1\,t_2]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.

Proof: *i)* $\subseteq$)

$$\begin{array}{rcl}
[\![t_1]\!]^{\mathcal{A}}_{\mathcal{C}} \cdot [\![t_2]\!]^{\mathcal{A}}_{\mathcal{C}} & = & (\text{Definition } \mathit{9.2(i)}) \\
\{a \in \mathcal{A}_{\mathcal{C}} \mid \exists a_1 \in [\![t_1]\!]^{\mathcal{A}}_{\mathcal{C}}, a_2 \in [\![t_2]\!]^{\mathcal{A}}_{\mathcal{C}}, u [a_1\,a_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\} & = & \\
\{a \in \mathcal{A}_{\mathcal{C}} \mid \exists a_1, a_2 \in \mathcal{A}_{\mathcal{C}}, u\, [\exists u_1\, [t_1 \rightarrow^* u_1 \;\&\; a_1 \sqsubseteq u_1] \;\& & & \\
\exists u_2\, [t_2 \rightarrow^* u_2 \;\&\; a_2 \sqsubseteq u_2] \;\&\; a_1\,a_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\} & \subseteq & (\text{Lemma } \mathit{7.5(ii)}) \\
\{a \in \mathcal{A}_{\mathcal{C}} \mid \exists u\, [t_1\,t_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\} & = & [\![t_1\,t_2]\!]^{\mathcal{A}}_{\mathcal{C}}
\end{array}$$

$\supseteq$)

$$\begin{array}{rcl}
[\![t_1\,t_2]\!]^{\mathcal{A}}_{\mathcal{C}} = \{a \in \mathcal{A}_{\mathcal{C}} \mid \exists u\, [t_1\,t_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\} & \subseteq & (\text{Lemma } \mathit{7.11}) \\
\{a \in \mathcal{A}_{\mathcal{C}} \mid \exists a_1 \in [\![t_1]\!]^{\mathcal{A}}_{\mathcal{C}}, a_2 \in [\![t_2]\!]^{\mathcal{A}}_{\mathcal{C}}, u[a_1\,a_2 \rightarrow^* u \;\&\; a \sqsubseteq u]\} & = & [\![t_1]\!]^{\mathcal{A}}_{\mathcal{C}} \cdot [\![t_2]\!]^{\mathcal{A}}_{\mathcal{C}}
\end{array}$$

ii)

$$\begin{array}{rcl}
[\![t_1]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} \cdot [\![t_2]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} & = & \uparrow\{\sigma \mid \exists\, \tau \in [\![t_2]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}\, [\tau \rightarrow \sigma \in [\![t_1]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}]\} \\
& = & \uparrow\{\sigma \mid \exists \tau\, [\exists B\, [B \vdash_{\mathcal{E}} t_1{:}\tau \rightarrow \sigma] \;\&\; \exists B\, [B \vdash_{\mathcal{E}} t_2{:}\tau]]\} \\
& = & \uparrow\{\sigma \mid \exists \tau, B\, [B \vdash_{\mathcal{E}} t_1{:}\tau \rightarrow \sigma \;\&\; B \vdash_{\mathcal{E}} t_2{:}\tau]\} \\
& = & \uparrow\{\sigma \mid \exists B\, [B \vdash_{\mathcal{E}} t_1\,t_2{:}\sigma]\} \\
& = & \{\sigma \mid \exists B\, [B \vdash_{\mathcal{E}} t_1\,t_2{:}\sigma]\} \\
& = & [\![t_1\,t_2]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}
\end{array}$$

■

As seen above in Lemma *7.9(i)*, if $t \rightarrow^* t'$, then $\mathcal{A}_{\mathcal{C}}(t) = \mathcal{A}_{\mathcal{C}}(t')$, which implies that, at least, if $t \rightarrow^* t'$, then $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![t']\!]^{\mathcal{A}}_{\mathcal{C}}$. The converse does not hold, since unsolvable terms that are not in $\rightarrow^*$, still have the same image under $[\![\;]\!]^{\mathcal{A}}_{\mathcal{C}}$, namely $\perp$.

The following relation expresses that terms are equivalent if they share a common reduct.

Definition 9.5 We define the equivalence relation $=_{\mathbf{R}} \;\subseteq T(\mathcal{C},\mathcal{X}) \times T(\mathcal{C},\mathcal{X})$ by:

$$\begin{array}{rcl}
t \rightarrow^*_{\mathbf{R}} v & \Rightarrow & t =_{\mathbf{R}} v \\
t =_{\mathbf{R}} v & \Rightarrow & v =_{\mathbf{R}} t \\
t =_{\mathbf{R}} v \;\&\; v =_{\mathbf{R}} w & \Rightarrow & t =_{\mathbf{R}} w
\end{array}$$

Lemma 9.6 *If $t =_{\mathbf{R}} v$, then there exists u such that $t \rightarrow^*_{\mathbf{R}} u$ and $v \rightarrow^*_{\mathbf{R}} u$.*

Proof: By induction on the definition of $=_{\mathbf{R}}$. If $t =_{\mathbf{R}} v \;\&\; v =_{\mathbf{R}} w \Rightarrow t =_{\mathbf{R}} w$, then, by induction there are u_1 and u_2 such that $t \rightarrow^*_{\mathbf{R}} u_1$ and $v \rightarrow^*_{\mathbf{R}} u_1$, and $v \rightarrow^*_{\mathbf{R}} u_2$ and $w \rightarrow^*_{\mathbf{R}} u_2$. Since $v \rightarrow^*_{\mathbf{R}} u_1$ and $v \rightarrow^*_{\mathbf{R}} u_2$, by Property *1.6*, there exist a u_3 such that $u_1 \rightarrow^*_{\mathbf{R}} u_3$ and $u_2 \rightarrow^*_{\mathbf{R}} u_3$. But then, in particular, $t \rightarrow^*_{\mathbf{R}} u_3$ and $w \rightarrow^*_{\mathbf{R}} u_3$.

The other cases are straightforward. ■

The approximant semantics is adequate, in that it equates terms that have a common reduct.

Theorem 9.7 (Adequacy of the Approximation Model) *If $t =_{\mathbf{R}} v$, then $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![v]\!]^{\mathcal{A}}_{\mathcal{C}}$.*

Proof: Consequence of Lemmas *9.6* and *7.9(i)*. ■

The converse of this result, '*If $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![v]\!]^{\mathcal{A}}_{\mathcal{C}}$, then $t =_{\mathbf{R}} v$*' does not hold.

Example 9.8 Take

$$\begin{array}{rcl}
\mathbf{D}x & \rightarrow & x\,x \\
\mathbf{W}x & \rightarrow & x\,x\,x
\end{array}$$

Notice that $\mathbf{SK}(\mathbf{DD})$ and $\mathbf{SK}(\mathbf{WW})$ both have only one redex, and that this property is preserved under reduction. Then

$$\mathbf{SK}(\mathbf{DD}) \rightarrow \mathbf{SK}(\mathbf{DD}) \rightarrow \mathbf{SK}(\mathbf{DD}) \rightarrow \cdots$$

and

$$\mathbf{SK}(\mathbf{WW}) \rightarrow \mathbf{SK}(\mathbf{WWW}) \rightarrow \mathbf{SK}(\mathbf{WWWW}) \rightarrow \cdots,$$

so

$$[\![\mathbf{SK}(\mathbf{DD})]\!]^{\mathcal{A}}_{\mathcal{C}} = \{\bot, S\bot\bot, SK\bot\} = [\![\mathbf{SK}(\mathbf{WW})]\!]^{\mathcal{A}}_{\mathcal{C}},$$

but there is no u such that both $\mathbf{SK}(\mathbf{DD}) \to^* u$ and $\mathbf{SK}(\mathbf{WW}) \to^* u$.

We could identify all unsolvable terms, as to obtain $\mathbf{SK}(\mathbf{DD}) \approx_{\mathbf{R}} \mathbf{SK}(\mathbf{WW})$, as is used also for LC.

Definition 9.9 We define the equivalence relation $\approx_{\mathbf{R}} \subseteq T(\mathcal{C},\mathcal{X}) \times T(\mathcal{C},\mathcal{X})$ by:

$$\begin{array}{rcl}
t \to^*_{\mathbf{R}} v & \Rightarrow & t \approx_{\mathbf{R}} v \\
t, v \text{ are unsolvable} & \Rightarrow & t \approx_{\mathbf{R}} v \\
t \approx_{\mathbf{R}} v & \Rightarrow & v \approx_{\mathbf{R}} t \\
t \approx_{\mathbf{R}} v \;\&\; v \approx_{\mathbf{R}} w & \Rightarrow & t \approx_{\mathbf{R}} w \\
t \approx_{\mathbf{R}} v & \Rightarrow & wt \approx_{\mathbf{R}} wv \;\&\; tw \approx_{\mathbf{R}} vw
\end{array}$$

Notice that $\mathbf{SK}(\mathbf{DD}) \approx_{\mathbf{R}} \mathbf{SK}(\mathbf{WW})$.

Theorem 9.10 *If $t \approx_{\mathbf{R}} v$, then $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![v]\!]^{\mathcal{A}}_{\mathcal{C}}$.*
Proof: By induction on the definition of $\approx_{\mathbf{R}}$. The case $t \to^*_{\mathbf{R}} v$ follows from Lemma *7.9(i)*. If t, v are unsolvable, then $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = \{\bot\} = [\![v]\!]^{\mathcal{A}}_{\mathcal{C}}$. The last case is a consequence of Lemma *9.4*. The other two cases are trivial. ■

Although, by $\approx_{\mathbf{R}}$, terms are equated that are unsolvable, still we do not get a full-abstraction result, since it can be that solvable terms have the same infinite set of approximants, whilst sharing no terms during reduction.

Example 9.11 Take

$$\begin{array}{rcl}
\mathbf{T}xy & \to & y(xxy) \\
\mathbf{Y}xy & \to & y(xy(xy)) \\
\mathbf{X}xy & \to & x(yy)
\end{array}$$

Then we have the following reduction sequences:

$$\begin{array}{rcl@{\qquad}rcl}
\mathbf{YX}z & \to & z(\mathbf{X}z(\mathbf{X}z)) & \mathbf{TT}z & \to & z(\mathbf{TT}z) \\
& \to & z(z(\mathbf{X}z(\mathbf{X}z))) & & \to & z(z(\mathbf{TT}z)) \\
& \to & z(z(z(\mathbf{X}z(\mathbf{X}z)))) & & \to & z(z(z(\mathbf{TT}z))) \\
& \cdots & & & \cdots & \\
& \to & z(z(z(z(z(z\cdots))))) & & \to & z(z(z(z(z(z\cdots)))))
\end{array}$$

In particular,

$$[\![\mathbf{YX}z]\!]^{\mathcal{A}}_{\mathcal{C}} \;=\; \{\bot, z\bot, z(z\bot), z(z(z\bot)), \ldots\} \;=\; [\![\mathbf{TT}z]\!]^{\mathcal{A}}_{\mathcal{C}},$$

but *not* $\mathbf{YX}z \approx_{\mathbf{R}} \mathbf{TT}z$.

We can obtain a full-abstraction result for the approximation semantics using the following relation:

Definition 9.12 The relation $\approx^{hnf}_{\mathbf{R}}$ is defined co-inductively as follows: $t \approx^{hnf}_{\mathbf{R}} u$ if and only if either

i) t and u are both unsolvable, or

ii) if $Ct_1 \cdots t_n$ is the *hnf* of t, then the *hnf* of u is $Cu_1 \cdots u_n$, and, for $1 \leq i \leq n$, $t_i \approx^{hnf}_{\mathbf{R}} u_i$, or

iii) if $xt_1 \cdots t_n$ is the *hnf* of t, then the *hnf* of u is $xu_1 \cdots u_n$, and, for $1 \leq i \leq n$, $t_i \approx^{hnf}_{\mathbf{R}} u_i$.

Theorem 9.13 (Full Abstraction of the Approximation Model) *$t \approx^{hnf}_{\mathbf{R}} u$ if and only if $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u]\!]^{\mathcal{A}}_{\mathcal{C}}$.*
Proof: *(only if)* By coinduction. It is sufficient to show that if $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u]\!]^{\mathcal{A}}_{\mathcal{C}}$ then either

i) t, u are unsolvable, or

ii) if $Ct_1 \cdots t_n$ is the *hnf* of t then $Cu_1 \cdots u_n$ is the *hnf* of u and $[\![t_i]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u_i]\!]^{\mathcal{A}}_{\mathcal{C}}$ for $1 \leq i \leq n$, or

iii) if $xt_1 \cdots t_n$ is the *hnf* of t then $xu_1 \cdots u_n$ is the *hnf* of u and $[\![t_i]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u_i]\!]^{\mathcal{A}}_{\mathcal{C}}$ for $1 \leq i \leq n$.

This is a straightforward consequence of the fact that u and t have the same set of approximants.

(if) We take $a \in [\![t]\!]^{\mathcal{A}}_{\mathcal{C}}$ and show $a \in [\![u]\!]^{\mathcal{A}}_{\mathcal{C}}$ by induction on the depth of a.

($a = \bot$): Trivial.

($a = Ca_1 \ldots a_n$): Then $\mathit{hnf}(t) = Ct_1 \cdots t_n$, therefore $\mathit{hnf}(u) = Cu_1 \cdots u_n$ and $t_i \approx^{\mathit{hnf}}_{\mathbf{R}} u_i$ for $1 \leq i \leq n$. Since $a_i \in [\![t_i]\!]^{\mathcal{A}}_{\mathcal{C}}$ and its depth is smaller than that of a, by induction we conclude that $a_i \in [\![u_i]\!]^{\mathcal{A}}_{\mathcal{C}}$. Therefore $a \in [\![u]\!]^{\mathcal{A}}_{\mathcal{C}}$.

($a = xa_1 \ldots a_n$): Similar. ∎

The filter semantics gives a semi-model with respect to $\rightarrow_{\mathbf{R}}$.

Theorem 9.14 *If $t \rightarrow^*_{\mathbf{R}} v$, then $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} \subseteq [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.*

Proof: Take $\sigma \in [\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$. Then $\exists B\, [B \models \xi \ \&\ B \vdash_{\mathcal{E}} t{:}\sigma]$, and, since $t \rightarrow^*_{\mathbf{R}} v$, by Theorem *3.7*, also $\exists B\, [B \models \xi \ \&\ B \vdash_{\mathcal{E}} v{:}\sigma]$, so $\sigma \in [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$. ∎

In view of the fact that type assignment in $\vdash_{\mathcal{E}}$ is not closed for subject-expansion (see the remark at the end of Section *3*), it is, in general, not possible to show a stronger result like '*If $t =_{\mathbf{R}} v$, then $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$*'. However, when using a principal environment, the result holds.

Theorem 9.15 (Adequacy of the Filter Model) *Let $((\mathcal{C},\mathcal{X}),\mathbf{R})$ be a* CS*, and $\mathcal{E}$ be principal for $\mathcal{C}$, then $t =_{\mathbf{R}} v$ implies $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.*

Proof: By Theorem *3.7* and *3.8*. ∎

We even have the following result easily.

Theorem 9.16 *Let $((\mathcal{C},\mathcal{X}),\mathbf{R})$ be a* CS*, and $\mathcal{E}$ be principal for $\mathcal{C}$, then $t \approx_{\mathbf{R}} v$ implies $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.*

Proof: By induction on the definition of $\approx_{\mathbf{R}}$. The case $t \rightarrow^*_{\mathbf{R}} v$ is covered by Theorem *3.7* and *3.8*. If t, v are unsolvable, then by Theorem *8.4*, $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = \{\omega\} = [\![v]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$. The last case is a consequence of Lemma *9.4*. The other two cases follow by straightforward induction. ∎

The converse of these results do not hold.

Example 9.17 Take $\mathbf{T}, \mathbf{Y}, \mathbf{X}$ as in Example *9.11*, and let

$$\begin{aligned} \mathcal{E}(\mathbf{T}) &= ((\varphi_1{\rightarrow}\varphi_2{\rightarrow}\varphi_3)\cap\varphi_1){\rightarrow}((\varphi_3{\rightarrow}\varphi_4)\cap\varphi_2){\rightarrow}\varphi_4, \\ \mathcal{E}(\mathbf{Y}) &= ((\varphi_1{\rightarrow}\varphi_2)\cap\varphi_3\cap\varphi_4){\rightarrow}((\varphi_3{\rightarrow}\varphi_5{\rightarrow}\varphi_1)\cap(\varphi_4{\rightarrow}\varphi_5){\rightarrow}\varphi_2, \\ \mathcal{E}(\mathbf{X}) &= (\varphi_1{\rightarrow}\varphi_2){\rightarrow}((\varphi_3{\rightarrow}\varphi_1)\cap\varphi_3){\rightarrow}\varphi_2, \end{aligned}$$

then

$$\begin{aligned} [\![\mathbf{YX}]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} &= \{\omega, (\omega{\rightarrow}\varphi_1){\rightarrow}\varphi_1, ((\omega{\rightarrow}\varphi_1)\cap(\varphi_1{\rightarrow}\varphi_2)){\rightarrow}\varphi_2, \\ &\qquad ((\omega{\rightarrow}\varphi_1)\cap(\varphi_1{\rightarrow}\varphi_2)\cap(\varphi_2{\rightarrow}\varphi_3)){\rightarrow}\varphi_3, \ldots\} = [\![\mathbf{TT}]\!]^{\mathcal{F}}, \end{aligned}$$

(notice that these types correspond directly to the approximants of Example *9.11*) but neither $\mathbf{YX} =_{\mathbf{R}} \mathbf{TT}$, nor $\mathbf{YX} \approx_{\mathbf{R}} \mathbf{TT}$.

For the filter semantics, we have, as can be expected:

Theorem 9.18 *Let $((\mathcal{C},\mathcal{X}),\mathbf{R})$ be a* CS*, and $\mathcal{E}$ be principal for $\mathcal{C}$, then $t \approx^{\mathit{hnf}}_{\mathbf{R}} u$ implies $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![u]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.*

Proof: If $t \approx_{\mathbf{R}}^{hnf} u$, then, by Theorem *9.13*, $[\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u]\!]^{\mathcal{A}}_{\mathcal{C}}$. Let $\sigma \in [\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$ (the other case is similar), then there exists a B such that $B \models \xi$ & $B \vdash_{\mathcal{E}} t{:}\sigma$. Then, by Theorem *8.2*, there exists an $a \in \mathcal{A}_{\mathcal{C}}(t)$ such that $B \vdash_{\mathcal{E}} a{:}\sigma$. Since $\mathcal{A}_{\mathcal{C}}(t) = [\![t]\!]^{\mathcal{A}}_{\mathcal{C}} = [\![u]\!]^{\mathcal{A}}_{\mathcal{C}} = \mathcal{A}_{\mathcal{C}}(u)$, $a \in \mathcal{A}_{\mathcal{C}}(u)$, and by Theorem *8.3*, $B \vdash_{\mathcal{E}} u{:}\sigma$, so $\sigma \in [\![u]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$. ∎

Perhaps surprisingly (in LC the approximation and the filter semantics coincide), we do not have a full-abstraction result with respect to filter semantics.

Example 9.19 Take

$$\begin{array}{lcl} \mathbf{E}xy & \to & xy \\ \mathbf{I}x & \to & x \end{array} \quad \text{and} \quad \begin{array}{lcl} \mathcal{E}(\mathbf{E}) & = & ((\varphi_1{\to}\varphi_2){\to}\varphi_1{\to}\varphi_2 \\ \mathcal{E}(\mathbf{I}) & = & \varphi_1{\to}\varphi_1 \end{array}$$

Then

$$[\![\mathbf{E}\,\mathbf{I}]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} = [\![\mathbf{I}]\!]^{\mathcal{F}}_{\xi,\mathcal{E}},$$

but neither $\mathbf{E}\,\mathbf{I} =_{\mathbf{R}} \mathbf{I}$, nor $\mathbf{E}\,\mathbf{I} \approx_{\mathbf{R}} \mathbf{I}$, nor $\mathbf{E}\,\mathbf{I} \approx_{\mathbf{R}}^{hnf} \mathbf{I}$.

The relation between the two semantics is formulated by:

Theorem 9.20 $[\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} \subseteq \bigcup_{a \in [\![t]\!]^{\mathcal{A}}_{\mathcal{C}}} [\![a]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.

Proof: If $\sigma \in [\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$, then there is a B such that $B \models \xi$ and $B \vdash_{\mathcal{E}} t{:}\sigma$. Then, by Theorem *8.2*, there is an $a \in \mathcal{A}_{\mathcal{C}}(t)$ such that $B \vdash_{\mathcal{E}} a{:}\sigma$. ∎

Note that the inclusion is strict, since the Subject Expansion property does not hold in general. Also, as can be expected:

Theorem 9.21 Let $((\mathcal{C},\mathcal{X}),\mathbf{R})$ be a CS*, $\mathcal{E}$ principal for $\mathcal{C}$, then $\bigcup_{a \in [\![t]\!]^{\mathcal{A}}_{\mathcal{C}}} [\![a]\!]^{\mathcal{F}}_{\xi,\mathcal{E}} \subseteq [\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$.*

Proof: If $\sigma \in \bigcup_{a \in [\![t]\!]^{\mathcal{A}}_{\mathcal{C}}} [\![a]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$, then there exists $a \in [\![t]\!]^{\mathcal{A}}_{\mathcal{C}}$, and B such that $B \vdash_{\mathcal{E}} a{:}\sigma$. Then, by Theorem *8.3*, also $B \vdash_{\mathcal{E}} t{:}\sigma$, so $\sigma \in [\![t]\!]^{\mathcal{F}}_{\xi,\mathcal{E}}$. ∎

10 Conclusions

The approximation result has important consequences both from a computational point of view, since it allows us to characterise the normalization properties of typeable terms, and from a semantic point of view, since it allows us to study the relations between filter models and approximantion models. This is true both for the LC and for CS, but the characterizations of normalization and the relations between the models are different in each case. The most striking difference is probably the fact that the models do not coincide in general in the case of CS (the filter model is only a semi-model in general) whereas they do coincide for the LC. Of course, the lack of Subject Expansion in CS explains the fact that we only have a semi-model. However, the fact that for CS the approximation model is fully abstract, but the filter model is not, is related to the fact that we have a “weak” form of reduction in CS, compared with the reduction in LC.

The proof of the approximation result uses a notion of Cut Elimination (Derivation Reduction) which is new in the context of intersection types. It can be adapted to other rewriting systems (in particular, the LC and TRS), where it also helps to obtain easier proofs of the characterisation of normalisation properties of typeable terms (for TRS the proof was sketched in [4]). In the future we hope to be able to extend the semantic study presented in this paper to the more general TRS studied in [4].

Acknowledgments

We would like to thank Mariangiola Dezani and Felice Cardone for many inspiring discussions on the subject of this paper.

References

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

[2] S. van Bakel. Principal type schemes for the Strict Type Assignment System. *Logic and Computation*, 3(6):643–670, 1993.

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

[4] S. van Bakel and M. Fernández. Approximation and Normalization Results for Typeable Term Rewriting Systems. In Gilles Dowek, Jan Heering, Karl Meinke, and Bernhard Möller, editors, *Proceedings of HOA '95. Second International Workshop on Higher Order Algebra, Logic and Term Rewriting*, Paderborn, Germany. *Selected Papers*, volume 1074 of *Lecture Notes in Computer Science*, pages 17–36. Springer-Verlag, 1996.

[5] S. van Bakel and M. Fernández. Normalization Results for Typeable Rewrite Systems. *Information and Computation*, 133(2):73–116, 1997.

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

[7] 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.

[8] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Functional characters of solvable terms. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 27:45–58, 1981.

[9] H.B. Curry. Grundlagen der Kombinatorischen Logik. *American Journal of Mathematics*, 52:509–536, 789–834, 1930.

[10] H.B. Curry. Functionality in Combinatory Logic. In *Proc. Nat. Acad. Sci. U.S.A.*, volume 20, pages 584–590, 1934.

[11] H.B. Curry and R. Feys. *Combinatory Logic*, volume 1. North-Holland, Amsterdam, 1958.

[12] N. Dershowitz and J.P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6, pages 245–320. North-Holland, 1990.

[13] M. Dezani-Ciancaglini and J.R. Hindley. Intersection types for combinatory logic. *Theoretical Computer Science*, 100:303–324, 1992.

[14] K. Futatsugi, J. Goguen, J.P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Proceedings 12*$^{\text{th}}$ *ACM Symposium on Principles of Programming Languages*, pages 52–66, 1985.

[15] 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.

[16] G. Huet and J.J. Lévy. Computations in Orthogonal Rewriting Systems. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic. Essays in Honour of Alan Robinson*. MIT Press, 1991.

[17] J.W. Klop. Term Rewriting Systems. In S. Abramsky, Dov.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 1–116. Clarendon Press, 1992.

[18] J.W. Klop and A. Middeldorp. Sequentiality in Orthogonal Term Rewriting Systems. *Journal of Symbolic Computation*, 12:161–195, 1991.

[19] S. Ronchi Della Rocca and B. Venneri. Principal type schemes for an extended type theory. *Theoretical Computer Science*, 28:151–169, 1984.

[20] W.W. Tait. Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic*, 32(2):198–223, 1967.

[21] S.R. Thatte. Full Abstraction and Limiting Completeness in Equational Languages. *Theoretical Computer Science*, 65:85–119, 1989.

[22] C.P. Wadsworth. The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus. *SIAM J. Comput.*, 5:488–521, 1976.