A Direct Proof of Strong Normalization for an Extended Herbelin's Calculus

Kentaro Kikuchi*

Department of Mathematics and Informatics, Chiba University,
1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
kentaro@math.s.chiba-u.ac.jp
June 28, 2006

Abstract. Herbelin presented (at CSL'94) an explicit substitution calculus with a sequent calculus as a type system, in which reduction steps correspond to cut-elimination steps. The calculus, extended with some rules for substitution propagation, simulates β-reduction of ordinary λ-calculus. In this paper we present a proof of strong normalization for the typable terms of the calculus. The proof is a direct one in the sense that it does not depend on the result of strong normalization for the simply typed λ-calculus, unlike an earlier proof by Dyckhoff and Urban.

1 Introduction

In [12], Herbelin introduced a sequent calculus in which a unique cut-free proof is associated to each normal term of the simply typed λ-calculus. This is in contrast to the usual assignment (see, e.g. [18, p. 73]) which associates several cut-free proofs to the same normal terms. Herbelin developed a term calculus whose reduction steps correspond to cut-elimination steps in the sequent calculus. Some of the cut rules introduce explicit substitution operators [1], and cut-propagation steps of cut-elimination correspond to the propagation of explicit substitutions. Herbelin proved strong normalization for the typed terms of his calculus.

However, he also observed that the reduction rules in the calculus are not enough to simulate full β-reduction (e.g., it fails to simulate the leftmost reduction). Espírito Santo [11] and Dyckhoff and Urban [10] identified a set of terms in the calculus that correspond to terms in the untyped λ-calculus, and introduced additional reduction rules needed to allow substitutions to propagate properly. Thus it turned out that ordinary λ-calculus can be completely embedded in Herbelin's calculus with the additional rules, and in the typed case β-reduction can be analyzed through cut-elimination.

On the other hand, it is reported [8] that Herbelin's sequent calculus is particularly suited to proof search used in the theory of logic programming. This means that the extended Herbelin's calculus is a promising candidate for a proof-theoretic basis for integrating functional and logic programming languages.

* This work was supported by JSPS Research Fellowships for Young Scientists and by Grant-in-Aid for JSPS Fellows.

Since the extended Herbelin's calculus behaves as an explicit substitution calculus for the λ-calculus, it is expected that various techniques from the field of explicit substitutions work as well for this calculus. Dyckhoff and Urban [10] indeed proved strong normalization for the typed terms of the calculus, using the method of [4] and the result of strong normalization for the simply typed λ-calculus. Note that as shown in [15], strong normalization for typed terms of an explicit substitution calculus is not a trivial property. In fact, a careful choice of reduction rules is required for proving strong normalization of cut-elimination that simulates β-reduction.

In this paper we prove strong normalization for the typable terms of the extended Herbelin's calculus directly without using the result for the simply typed λ-calculus. Our proof is an adaptation of the reducibility method [17] to explicit substitution calculus and to Herbelin-style calculus. For the explicit substitution calculus $\lambda\mathtt{x}$ [5], such adaptations have been considered in [6, 7]. Compared with their proofs, ours makes use of a more general closure condition on reducibility sets; they are closed under $\mathtt{x}$-conversion whenever the term is decent (Lemma 8). This is closely related to a lemma for an inductive proof of preservation of strong normalization (PSN) as developed in [2, 5]. Our method is easily applicable to the case of $\lambda\mathtt{x}$, simplifying the proofs in [6, 7].

The paper is organized as follows. In Section 2 we introduce the calculus and type system. In Section 3 we consider the subset of terms that correspond to ordinary λ-terms. In Section 4 we study a subcalculus that plays an important role in our proofs. In Section 5 we explain how to simulate β-reduction in the calculus. In Section 6 we prove the main lemma, from which we derive PSN, and in Section 7 we give a reducibility proof of strong normalization using the main lemma. Finally in Section 8 we conclude and give suggestions for further work.

2 $\overline{\lambda}\mathtt{x}$-calculus

Table 1 presents the syntax and typing rules of $\overline{\lambda}\mathtt{x}$-calculus, which is the same as the calculus $(AO + ES + B)$ in [10] and varies a little from the calculi in [11], although we mainly follow notations in the latter. The syntax of $\overline{\lambda}\mathtt{x}$ has two kinds of expressions: terms and lists of terms, ranged over by u, v, t and by l, l', respectively. The set of terms is denoted by $\mathcal{T}_{\overline{\lambda}\mathtt{x}}$ and the set of lists of terms by $\mathcal{L}_{\overline{\lambda}\mathtt{x}}$. Elements of $\mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$ are called $\overline{\lambda}\mathtt{x}$-*terms* and ranged over by a, b, c. In $a\langle x := v\rangle$, $\langle x := v\rangle$ is called an *explicit substitution* or simply substitution and v is called the *body* of the substitution. The notions of free and bound variables are defined as usual, with an additional clause that the variable x in $a\langle x := v\rangle$ binds the free occurrences of x in a. We assume the following variable convention: names of bound variables are different from the names of free variables, and, moreover, different occurrences of the abstraction operator have different binding variables. The set of free variables of a $\overline{\lambda}\mathtt{x}$-term a is denoted by $FV(a)$. The symbol $\equiv$ denotes syntactical equality modulo α-conversion.

A *typing context*, ranged over by Γ, is a finite set of pairs $\{x_1 : A_1, \ldots, x_n : A_n\}$ where the variables are pairwise distinct. $\Gamma, x : A$ denotes the union $\Gamma \cup \{x :$

Table 1. $\overline{\lambda}$x-calculus

$$u, v, t ::= xl \mid \lambda x.t \mid tl \mid t\langle x := v\rangle$$
$$l, l' ::= [] \mid t :: l \mid ll' \mid l\langle x := v\rangle$$

$$\frac{}{\Gamma; A \vdash [] : A}\ Ax \qquad \frac{\Gamma, x : A; A \vdash l : B}{\Gamma, x : A; - \vdash xl : B}\ Der$$

$$\frac{\Gamma; - \vdash t : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash t :: l : C}\ L\supset \qquad \frac{\Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash \lambda x.t : A \supset B}\ R\supset$$

$$\frac{\Gamma; B \vdash l : A \quad \Gamma; A \vdash l' : C}{\Gamma; B \vdash ll' : C}\ Cut_1 \qquad \frac{\Gamma; - \vdash v : A \quad \Gamma, x : A; B \vdash l : C}{\Gamma; B \vdash l\langle x := v\rangle : C}\ Cut_2$$

$$\frac{\Gamma; - \vdash t : A \quad \Gamma; A \vdash l : B}{\Gamma; - \vdash tl : B}\ Cut_3 \qquad \frac{\Gamma; - \vdash v : A \quad \Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash t\langle x := v\rangle : B}\ Cut_4$$

$(Beta)$	$(\lambda x.t)(u :: l) \to t\langle x := u\rangle l$	
(1a)	$[]l \to l$	
(1b)	$(u :: l)l' \to u :: (ll')$	
(2a)	$[]\langle x := v\rangle \to []$	
(2b)	$(u :: l)\langle x := v\rangle \to u\langle x := v\rangle :: l\langle x := v\rangle$	
(3a)	$(xl)l' \to x(ll')$	
(3b)	$(\lambda y.t)[] \to \lambda y.t$	
(4a)	$(yl)\langle x := v\rangle \to yl\langle x := v\rangle$	$(y \not\equiv x)$
(4b)	$(xl)\langle x := v\rangle \to vl\langle x := v\rangle$	
(4c)	$(\lambda y.t)\langle x := v\rangle \to \lambda y.t\langle x := v\rangle$	
(5a)	$(ll')l'' \to l(l'l'')$	
(5b)	$(ll')\langle x := v\rangle \to l\langle x := v\rangle l'\langle x := v\rangle$	
(5c)	$(tl)l' \to t(ll')$	
(5d)	$(tl)\langle x := v\rangle \to t\langle x := v\rangle l\langle x := v\rangle$	

$A\}$ where x does not appear in Γ. There are two kinds of derivable sequents: $\Gamma; - \vdash t : B$ and $\Gamma; A \vdash l : B$, both of which have a distinguished position in the LHS called *stoup*. The crucial restriction of this sequent calculus is that the rule $L \supset$ introduces $A \supset B$ in the stoup and B has to be in the stoup of the right subderivation's endsequent. In the cut-free case, the last rule of the right subderivation of an instance of $L \supset$ is again $L \supset$ and so on until Ax is reached. This yields an assignment of a unique cut-free proof to each normal term of the simply typed λ-calculus (cf. [18, Section 6.3]).

The notion of $\overline{\lambda}\mathtt{x}$-reduction is defined by the contextual closures of all reduction rules in Table 1. We use $\rightarrow_{\overline{\lambda}\mathtt{x}}$ for one-step reduction, $\xrightarrow{+}_{\overline{\lambda}\mathtt{x}}$ for its transitive closure, and $\xrightarrow{*}_{\overline{\lambda}\mathtt{x}}$ for its reflexive transitive closure. The set of $\overline{\lambda}\mathtt{x}$-terms that are strongly normalizing with respect to $\overline{\lambda}\mathtt{x}$-reduction is denoted by $\mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. These kinds of notations are also used for the notions of other reductions introduced in this paper.

The subcalculus of $\overline{\lambda}\mathtt{x}$ without the *Beta*-rule is denoted by $\mathtt{x}$. This subcalculus plays an important role in this paper and is studied in Section 4.

Herbelin's original $\overline{\lambda}$-calculus [12] is essentially the calculus without the last four reduction rules in Table 1. These rules are necessary for the simulation of full β-reduction of the λ-calculus.

The reduction rules of $\overline{\lambda}\mathtt{x}$-calculus also define cut-elimination procedures for typing derivations of $\overline{\lambda}\mathtt{x}$-terms, which ensures that the subject reduction property holds in this type system. In Appendix A, we display the cut-elimination steps corresponding to $\overline{\lambda}\mathtt{x}$-reduction for typable terms.

3 Pure Terms

Table 2 presents the syntax of *pure terms*, which are the subset of $\overline{\lambda}\mathtt{x}$-terms that correspond to terms of ordinary λ-calculus. The grammar of pure terms is close to the inductive characterization of the set of λ-terms found, e.g. in [14]. For the definition of β-reduction on pure terms, we need meta-substitution $[\_/\_]$, which requires further meta-operations $\{\_\}\_$ and $\_@\_$ since a $\overline{\lambda}\mathtt{x}$-term obtained by substituting a pure term for a variable is not in general a pure term. Note the similarity between the definition of these meta-operations and the reduction rules of $\overline{\lambda}\mathtt{x}$-calculus. $\overline{\lambda}\mathtt{x}$-calculus can be considered in some sense a calculus making these meta-operations explicit, while usual explicit substitution calculi make the usual substitution explicit.

In Appendix B, we give basic properties of these meta-operations on pure terms.

Translations between pure terms and ordinary λ-terms are given in [11] and [10] through a grammar of λ-terms that is different from the usual one. Here we define translations between λ-terms with the usual grammar and pure terms as shown in Table 3.

Proposition 1. *$\Theta \circ \Psi = id$ and $\Psi \circ \Theta = id$.*

Proof. See Appendix C. □

Table 2. Pure terms

$$u, v, t ::= xl \mid \lambda x.t \mid (\lambda x.t)(u :: l)$$
$$l, l' ::= [] \mid t :: l$$

$$(\beta) \qquad (\lambda x.t)(u :: l) \;\rightarrow\; \{t[u/x]\}l$$

where

$$\begin{aligned}
[]@l &=_{def} l \\
(u :: l)@l' &=_{def} u :: (l@l') \\
[][v/x] &=_{def} [] \\
(u :: l)[v/x] &=_{def} u[v/x] :: l[v/x] \\
\{xl\}l' &=_{def} x(l@l') \\
\{\lambda y.t\}[] &=_{def} \lambda y.t \\
\{\lambda y.t\}(u :: l) &=_{def} (\lambda y.t)(u :: l) \\
\{(\lambda y.t)(u :: l)\}l' &=_{def} (\lambda y.t)(u :: (l@l')) \\
(yl)[v/x] &=_{def} yl[v/x] \qquad (y \not\equiv x) \\
(xl)[v/x] &=_{def} \{v\}l[v/x] \\
(\lambda y.t)[v/x] &=_{def} \lambda y.t[v/x] \\
((\lambda y.t)(u :: l))[v/x] &=_{def} (\lambda y.t[v/x])(u[v/x] :: l[v/x])
\end{aligned}$$

Theorem 1.

1. *For any* λ*-terms* M, M'*, if* $M \rightarrow_\beta M'$ *then* $\Psi(M) \rightarrow_\beta \Psi(M')$*.*
2. *For any pure terms* $t, t' \in \mathcal{T}_{\overline{\lambda}\mathtt{x}}$*, if* $t \rightarrow_\beta t'$ *then* $\Theta(t) \rightarrow_\beta \Theta(t')$*.*

Proof. See Appendix C. □

It is also possible to show that the translations preserve the types of terms, defining translations on typing derivations. Later we see that β-reduction on pure terms can be simulated by $\overline{\lambda}\mathtt{x}$-reduction, thus showing how to simulate normalization in natural deduction by cut-elimination in Herbelin's sequent calculus.

4 Properties of the Subcalculus x

In this section we study properties of the subcalculus $\mathtt{x}$ which is obtained from $\overline{\lambda}\mathtt{x}$-calculus by deleting the *Beta*-rule. In the typed case it corresponds to cut-elimination steps except the key-case, i.e., the one where the cut-formula $A \supset B$

Table 3. Translations Ψ and Θ

$$\begin{array}{rl} \Psi(x) &=_{def} x[] \\ \Psi(MN) &=_{def} \{\Psi(M)\}(\Psi(N) :: []) \\ \Psi(\lambda x.M) &=_{def} \lambda x.\Psi(M) \end{array} \qquad \begin{array}{rl} \Theta(xl) &=_{def} \Theta'(x, l) \\ \Theta(\lambda x.t) &=_{def} \lambda x.\Theta(t) \\ \Theta((\lambda x.t)(u :: l)) &=_{def} \Theta'(\lambda x.\Theta(t), u :: l) \\ \Theta'(M, []) &=_{def} M \\ \Theta'(M, u :: l) &=_{def} \Theta'(M\Theta(u), l) \end{array}$$

is introduced in the last rules of both left and right subderivations. We show that the subcalculus is strongly normalizing and confluent and that its normal forms are pure terms.

Proposition 2. *The subcalculus* $\mathtt{x}$ *is strongly normalizing.*

Proof. The proof is by interpretation, following Appendix A of [9]. We define a function $h : \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}} \longrightarrow \mathbb{N}$ as follows:

$$\begin{array}{ll} h(xl) & =_{def} h(l) + 1 \\ h(\lambda x.t) & =_{def} h(t) + 1 \\ h(tl) & =_{def} 2 \times h(t) + h(l) + 1 \\ h(t\langle z := v\rangle) & =_{def} h(t) \times (3 \times h(v) + 1) \\ & \\ h([]) & =_{def} 1 \\ h(u :: l) & =_{def} h(u) + h(l) + 1 \\ h(ll') & =_{def} 2 \times h(l) + h(l') + 1 \\ h(l\langle z := v\rangle) & =_{def} h(l) \times (3 \times h(v) + 1) \end{array}$$

and observe that if $a \rightarrow_{\mathtt{x}} b$ then $h(a) > h(b)$. □

Proposition 3. *The subcalculus* $\mathtt{x}$ *is confluent.*

Proof. By Newman's Lemma, it suffices to check the local confluence; for details see Appendix D. □

As a result, we can define the unique $\mathtt{x}$-normal form of each $\overline{\lambda}\mathtt{x}$-term.

Definition 1. *Let* $a \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. *The unique* $\mathtt{x}$*-normal form of* a *is denoted by* $\mathtt{x}(a)$.

Proposition 4. *Let* $a \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. a *is a pure term iff* a *is in* $\mathtt{x}$*-normal form.*

Proof. The only if part is by induction on the structure of pure terms. We prove the if part by induction on the structure of a. Suppose that a is in $\mathtt{x}$-normal form. Then by the induction hypothesis, all subterms of a are pure. Here if a

is not pure then a is one of the forms $tl(\not\equiv (\lambda x.t_0)(u :: l_0))$, $t\langle x := v\rangle$, ll' and $l\langle x := v\rangle$ where t, l, v, l' are pure. In any case we see that a is an $\mathtt{x}$-redex, which is a contradiction. □

The next proposition shows that the subcalculus $\mathtt{x}$ correctly simulates the meta-operations on pure terms.

Proposition 5. *Let t, v, l, l' be pure terms with $t, v \in \mathcal{T}_{\overline{\lambda}\mathtt{x}}$ and $l, l' \in \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. Then*

1. $ll' \overset{*}{\rightarrow}_{\mathtt{x}} l@l'$,
2. $l\langle y := v\rangle \overset{*}{\rightarrow}_{\mathtt{x}} l[v/y]$,
3. $tl \overset{*}{\rightarrow}_{\mathtt{x}} \{t\}l$,
4. $t\langle y := v\rangle \overset{*}{\rightarrow}_{\mathtt{x}} t[v/y]$.

Proof. The first part is by induction on the structure of l. The third part is by the following case analysis.

3. (a) $t \equiv xl_0$. Then $(xl_0)l \rightarrow_{\mathtt{x}} x(l_0 l) \overset{1*}{\rightarrow}_{\mathtt{x}} x(l_0@l) \equiv \{xl_0\}l$.
   (b) $t \equiv \lambda y.t_0$ and $l \equiv []$. Then $(\lambda y.t_0)[] \rightarrow_{\mathtt{x}} \lambda y.t_0 \equiv \{\lambda y.t_0\}[]$.
   (c) $t \equiv \lambda y.t_0$ and $l \equiv u :: l_0$. Then $(\lambda y.t_0)(u :: l_0) \equiv \{\lambda y.t_0\}(u :: l_0)$.
   (d) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $((\lambda y.t_0)(u :: l_0))l \rightarrow_{\mathtt{x}} (\lambda y.t_0)((u :: l_0)l) \rightarrow_{\mathtt{x}} (\lambda y.t_0)(u :: (l_0 l)) \overset{1*}{\rightarrow}_{\mathtt{x}} (\lambda y.t_0)(u :: (l_0@l)) \equiv \{(\lambda y.t_0)(u :: l_0)\}l$.

The remaining two parts are proved by simultaneous induction on the structure of l or t.

2. (a) $l \equiv []$. Then $[]\langle y := v\rangle \rightarrow_{\mathtt{x}} [] \equiv [][v/y]$.
   (b) $l \equiv u :: l_0$. Then $(u :: l_0)\langle y := v\rangle \rightarrow_{\mathtt{x}} u\langle y := v\rangle :: l_0\langle y := v\rangle \overset{IH*}{\rightarrow}_{\mathtt{x}} u[v/y] :: l_0[v/y] \equiv (u :: l_0)[v/y]$.
4. (a) $t \equiv xl_0$ $(x \not\equiv y)$. Then $(xl_0)\langle y := v\rangle \rightarrow_{\mathtt{x}} xl_0\langle y := v\rangle \overset{IH*}{\rightarrow}_{\mathtt{x}} xl_0[v/y] \equiv (xl_0)[v/y]$.
   (b) $t \equiv yl_0$. Then $(yl_0)\langle y := v\rangle \rightarrow_{\mathtt{x}} vl_0\langle y := v\rangle \overset{IH*}{\rightarrow}_{\mathtt{x}} vl_0[v/y] \overset{3*}{\rightarrow}_{\mathtt{x}} \{v\}l_0[v/y] \equiv (yl_0)[v/y]$.
   (c) $t \equiv \lambda z.t_0$. Then $(\lambda z.t_0)\langle y := v\rangle \rightarrow_{\mathtt{x}} \lambda z.t_0\langle y := v\rangle \overset{IH*}{\rightarrow}_{\mathtt{x}} \lambda z.t_0[v/y] \equiv (\lambda z.t_0)[v/y]$.
   (d) $t \equiv (\lambda z.t_0)(u :: l_0)$. Then $((\lambda z.t_0)(u :: l_0))\langle y := v\rangle \rightarrow_{\mathtt{x}} (\lambda z.t_0)\langle y := v\rangle(u :: l_0)\langle y := v\rangle \overset{*}{\rightarrow}_{\mathtt{x}} (\lambda z.t_0\langle y := v\rangle)(u\langle y := v\rangle :: l_0\langle y := v\rangle) \overset{IH*}{\rightarrow}_{\mathtt{x}} (\lambda z.t_0[v/y])(u[v/y] :: l_0[v/y]) \equiv ((\lambda z.t_0)(u :: l_0))[v/y]$. □

From the above proposition we have the following lemma which allows us to reduce inference on $\mathtt{x}$-normal form to inference for meta-operations on pure terms.

Lemma 1. *Let $t, v \in \mathcal{T}_{\overline{\lambda}\mathtt{x}}$ and $l, l' \in \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. Then*

1. $\mathtt{x}(ll') \equiv \mathtt{x}(l)@\mathtt{x}(l')$,
2. $\mathtt{x}(l\langle y := v\rangle) \equiv \mathtt{x}(l)[\mathtt{x}(v)/y]$,

3. $\mathtt{x}(tl) \equiv \{\mathtt{x}(t)\}\mathtt{x}(l)$,
4. $\mathtt{x}(t\langle y := v\rangle) \equiv \mathtt{x}(t)[\mathtt{x}(v)/y]$.

Proof. We only consider the fourth part. Since $\mathtt{x}(t)$ and $\mathtt{x}(v)$ are pure terms, we have $\mathtt{x}(t)\langle y := \mathtt{x}(v)\rangle \xrightarrow{*}_{\mathtt{x}} \mathtt{x}(t)[\mathtt{x}(v)/y]$ by Proposition 5 (4). Hence, $\mathtt{x}(t\langle y := v\rangle) \equiv \mathtt{x}(\mathtt{x}(t)\langle y := \mathtt{x}(v)\rangle) \equiv \mathtt{x}(\mathtt{x}(t)[\mathtt{x}(v)/y]) \equiv \mathtt{x}(t)[\mathtt{x}(v)/y]$. □

5 Simulation of β-reduction

Now we are in a position to show that $\overline{\lambda}\mathtt{x}$-reduction simulates β-reduction.

Theorem 2. *For any pure terms $a, b \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$, if $a \rightarrow_\beta b$ then $a \xrightarrow{+}_{\overline{\lambda}\mathtt{x}} b$.*

Proof. By induction on the structure of a. We treat the case $a \equiv (\lambda x.t)(u :: l)$, $b \equiv \{t[u/x]\}l$. Then use $\rightarrow_{Beta}$ to create $t\langle x := u\rangle l$, and use Proposition 5 (4) and (3) to reach $\{t[u/x]\}l$. □

Since the translations in Section 3 preserve the types of terms, the proof of the above theorem indicates how to simulate normalization in natural deduction by cut-elimination in Herbelin's sequent calculus. Specifically, a redex in natural deduction is translated into the key-case corresponding to a *Beta*-redex $(\lambda x.t)(u :: l)$. Then transformation is performed as in Appendix A to create the proof corresponding to $t\langle x := u\rangle l$, followed by cut-reduction steps to reach the proof corresponding to $\{t[u/x]\}l$. The latter cut-reduction steps are in fact strongly normalizing and confluent, since they correspond to reduction steps of the subcalculus $\mathtt{x}$.

The strictness in Theorem 2 has a nice consequence.

Corollary 1. *Let $a \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. If $a \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ then $\mathtt{x}(a) \in \mathcal{SN}_\beta$.*

Proof. Suppose $\mathtt{x}(a) \notin \mathcal{SN}_\beta$. Using Theorem 2 we get an infinite $\overline{\lambda}\mathtt{x}$-reduction sequence starting with $\mathtt{x}(a)$. Since $a \xrightarrow{*}_{\overline{\lambda}\mathtt{x}} \mathtt{x}(a)$ we have $a \notin \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. □

6 Main Lemma

In this section we give an inductive proof of PSN for $\overline{\lambda}\mathtt{x}$-calculus with respect to β-reduction on pure terms. Although PSN itself was already proved in [10] in a different way, our main lemma is also useful for the reducibility proof of strong normalization in the next section. We follow the method of [2, 3, 5] for $\lambda\mathtt{x}$-calculus, in which the key notions are void reduction and decent terms.

Definition 2. *A substitution $\langle x := v\rangle$ is said to be* void *in $a\langle x := v\rangle$ if $x \notin FV(\mathtt{x}(a))$.* Void reduction *is $\overline{\lambda}\mathtt{x}$-reduction inside the body of a void substitution (more precisely, it is the contextual closure of the reduction: $a\langle z := v\rangle \rightarrow_{\overline{\lambda}\mathtt{x}} a\langle z := v'\rangle$ where $v \rightarrow_{\overline{\lambda}\mathtt{x}} v'$ and $z \notin FV(\mathtt{x}(a))$).*

As in the case of $\lambda\mathtt{x}$, we have the following lemmas.

Lemma 2 (Projection). *Let $a, b \in \mathcal{T}_{\overline{\lambda \mathtt{x}}} \cup \mathcal{L}_{\overline{\lambda \mathtt{x}}}$. If $a \to_{\overline{\lambda \mathtt{x}}} b$, then $\mathtt{x}(a) \xrightarrow{*}_\beta \mathtt{x}(b)$.*

Proof. If $a \to_{\mathtt{x}} b$ then $\mathtt{x}(a) \equiv \mathtt{x}(b)$. In what follows we show that if $a \to_{Beta} b$ then $\mathtt{x}(a) \xrightarrow{*}_\beta \mathtt{x}(b)$. For this we prove the following by simultaneous induction on the structure of t or l:

1. if $t \to_{Beta} t'$ then $\mathtt{x}(t) \xrightarrow{*}_\beta \mathtt{x}(t')$,
2. if $l \to_{Beta} l'$ then $\mathtt{x}(l) \xrightarrow{*}_\beta \mathtt{x}(l')$.

1. (a) $t \equiv y l_0$ with $l_0 \to_{Beta} l_0'$. By the induction hypothesis, $\mathtt{x}(l_0) \xrightarrow{*}_\beta \mathtt{x}(l_0')$. Hence $\mathtt{x}(y l_0) \equiv y \mathtt{x}(l_0) \xrightarrow{*}_\beta y \mathtt{x}(l_0') \equiv \mathtt{x}(y l_0')$.
   (b) $t \equiv \lambda y.t_0$ with $t_0 \to_{Beta} t_0'$. Similar use of the induction hypothesis.
   (c) $t \equiv t_0 l_0$. There are three cases.
      i. $t_0 \to_{Beta} t_0'$. By the induction hypothesis, $\mathtt{x}(t_0) \xrightarrow{*}_\beta \mathtt{x}(t_0')$. Hence

$$
\begin{aligned}
\mathtt{x}(t_0 l_0) &\equiv \{\mathtt{x}(t_0)\}\mathtt{x}(l_0) && \text{(by Lemma 1 (3))}\\
&\xrightarrow{*}_\beta \{\mathtt{x}(t_0')\}\mathtt{x}(l_0) && \text{(by Lemma 16 (1))}\\
&\equiv \mathtt{x}(t_0' l_0) && \text{(by Lemma 1 (3))}
\end{aligned}
$$

      ii. $l_0 \to_{Beta} l_0'$. Similar, using Lemma 16 (2).
      iii. $t \equiv (\lambda z.t_1)(u :: l_1) \to_{Beta} t_1\langle z := u\rangle l_1 \equiv t'$. Then

$$
\begin{aligned}
\mathtt{x}((\lambda z.t_1)(u :: l_1)) &\equiv \{\mathtt{x}(\lambda z.t_1)\}\mathtt{x}(u :: l_1) && \text{(by Lemma 1 (3))}\\
&\equiv \{\lambda z.\mathtt{x}(t_1)\}(\mathtt{x}(u) :: \mathtt{x}(l_1)) \\
&\equiv (\lambda z.\mathtt{x}(t_1))(\mathtt{x}(u) :: \mathtt{x}(l_1)) \\
&\to_\beta \{\mathtt{x}(t_1)[\mathtt{x}(u)/z]\}\mathtt{x}(l_1) \\
&\equiv \{\mathtt{x}(t_1\langle z := u\rangle)\}\mathtt{x}(l_1) && \text{(by Lemma 1 (4))}\\
&\equiv \mathtt{x}(t_1\langle z := u\rangle l_1) && \text{(by Lemma 1 (3))}
\end{aligned}
$$

   (d) $t \equiv t_0\langle z := v\rangle$. There are two cases.
      i. $t_0 \to_{Beta} t_0'$. By the induction hypothesis, $\mathtt{x}(t_0) \xrightarrow{*}_\beta \mathtt{x}(t_0')$. Hence

$$
\begin{aligned}
\mathtt{x}(t_0\langle z := v\rangle) &\equiv \mathtt{x}(t_0)[\mathtt{x}(v)/z] && \text{(by Lemma 1 (4))}\\
&\xrightarrow{*}_\beta \mathtt{x}(t_0')[\mathtt{x}(v)/z] && \text{(by Lemma 21 (1))}\\
&\equiv \mathtt{x}(t_0'\langle z := v\rangle) && \text{(by Lemma 1 (4))}
\end{aligned}
$$

      ii. $v \to_{Beta} v'$. By the induction hypothesis, $\mathtt{x}(v) \xrightarrow{*}_\beta \mathtt{x}(v')$. Hence

$$
\begin{aligned}
\mathtt{x}(t_0\langle z := v\rangle) &\equiv \mathtt{x}(t_0)[\mathtt{x}(v)/z] && \text{(by Lemma 1 (4))}\\
&\xrightarrow{*}_\beta \mathtt{x}(t_0)[\mathtt{x}(v')/z] && \text{(by Lemma 22 (1))}\\
&\equiv \mathtt{x}(t_0\langle z := v'\rangle) && \text{(by Lemma 1 (4))}
\end{aligned}
$$

2. (a) $l \equiv []$. Trivial.
   (b) $l \equiv u :: l_0$. Routine use of the induction hypothesis.

(c) $l \equiv l_0 l_1$. Similar to the first two parts of case 1(c) above.
(d) $l \equiv l_0\langle z := v\rangle$. Similar to the case 1(d) above. □

Lemma 3. *Let $a, b \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. If $a \rightarrow_{Beta} b$ and the reduction is not void, then $\mathtt{x}(a) \xrightarrow{+}_\beta \mathtt{x}(b)$.*

Proof. By a similar induction to the proof of Lemma 2. Since the reduction is not void, we have $z \in FV(\mathtt{x}(a_0))$ in the case $a \equiv a_0\langle z := v\rangle$ and $v \rightarrow_{Beta} v'$, and use Lemma 23 rather than Lemma 22. □

Lemma 4. *If $a_0, a_1, \ldots \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$ such that $\mathtt{x}(a_0) \in \mathcal{SN}_\beta$ and $a_0 \rightarrow_{\overline{\lambda}\mathtt{x}} a_1 \rightarrow_{\overline{\lambda}\mathtt{x}} \ldots$ is an infinite $\overline{\lambda}\mathtt{x}$-reduction sequence, there is a $k \in \mathbb{N}$ such that for all $i \geqslant k$, $a_i \rightarrow_{\overline{\lambda}\mathtt{x}} a_{i+1}$ is void.*

Proof. Since $\rightarrow_\mathtt{x}$ is strongly normalizing, we may assume that the infinite $\overline{\lambda}\mathtt{x}$-reduction sequence has the form $a_0 \xrightarrow{*}_\mathtt{x} a_1 \rightarrow_{Beta} a_2 \xrightarrow{*}_\mathtt{x} a_3 \ldots$. Now, by Lemma 2, we have $\mathtt{x}(a_0) \xrightarrow{*}_\beta \mathtt{x}(a_2) \xrightarrow{*}_\beta \mathtt{x}(a_4) \xrightarrow{*}_\beta \mathtt{x}(a_6) \ldots$, where by Lemma 3 we have $\mathtt{x}(a_{2n}) \xrightarrow{+}_\beta \mathtt{x}(a_{2n+2})$ if $a_{2n+1} \rightarrow_{Beta} a_{2n+2}$ is not void.

Now, since $\mathtt{x}(a_0) \in \mathcal{SN}_\beta$, there is a $j \in \mathbb{N}$ such that for all $i \geqslant j$, $a_{2i+1} \rightarrow_{Beta} a_{2i+2}$ is void. In what follows we prove that from some point onwards not only the $\rightarrow_{Beta}$ reductions are void but also the $\rightarrow_\mathtt{x}$ reductions. This is done by defining an interpretation h on $\mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$:

$$
\begin{array}{ll}
h(xl) & =_{def} h(l) + 1 \\
h(\lambda x.t) & =_{def} h(t) + 1 \\
h(tl) & =_{def} 2 \times h(t) + h(l) + 1 \\
h(t\langle z := v\rangle) & =_{def} \begin{cases} h(t) \times (3 \times h(v) + 1) & \text{if } z \in FV(\mathtt{x}(t)) \\ h(t) \times 4 & \text{if } z \notin FV(\mathtt{x}(t)) \end{cases}
\end{array}
$$

$$
\begin{array}{ll}
h([]) & =_{def} 1 \\
h(u :: l) & =_{def} h(u) + h(l) + 1 \\
h(ll') & =_{def} 2 \times h(l) + h(l') + 1 \\
h(l\langle z := v\rangle) & =_{def} \begin{cases} h(l) \times (3 \times h(v) + 1) & \text{if } z \in FV(\mathtt{x}(l)) \\ h(l) \times 4 & \text{if } z \notin FV(\mathtt{x}(l)) \end{cases}
\end{array}
$$

One may then verify that:

- if $a \rightarrow_{\overline{\lambda}\mathtt{x}} b$ is void, then $h(a) = h(b)$, and
- if $a \rightarrow_\mathtt{x} b$ is not void, then $h(a) > h(b)$.

Thus there must be a $k > j$ such that for all $i \geqslant k$ we have that not only $a_{2i+1} \rightarrow_{Beta} a_{2i+2}$ is void but also $a_{2i} \xrightarrow{*}_\mathtt{x} a_{2i+1}$. □

Definition 3 (Decent terms). *Let $a \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$. a is said to be* decent *if for every substitution $\langle z := v\rangle$ occurring in a, $v \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.*

The next proposition follows easily from the fact that void reduction takes place inside the body of a (void) substitution.

Proposition 6. *For decent terms, void reduction is strongly normalizing.*

Our aim is now to prove the converse of Corollary 1 when we restrict the $\overline{\lambda}\mathtt{x}$-terms in question to decent terms. Before we proceed to the proof we need one more lemma.

Definition 4. *Let a be a pure term with $a \in \mathcal{SN}_\beta$. $\mathrm{maxred}_\beta(a)$ is defined as the maximal length of all β-reduction sequences starting from a.*

Lemma 5. *Let $a, b \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$ with $\mathtt{x}(a) \in \mathcal{SN}_\beta$. If b is a subterm of a and is not inside the body of a substitution in a, then $\mathrm{maxred}_\beta(\mathtt{x}(b)) \leqslant \mathrm{maxred}_\beta(\mathtt{x}(a))$.*

Proof. By induction on the structure of a. We treat the case $a \equiv t\langle z := v\rangle$. If b is a strict subterm of a and is not inside the body of a substitution in a, then b is a subterm of t. Hence

$$\begin{array}{rcll} \mathrm{maxred}_\beta(\mathtt{x}(b)) & \leqslant & \mathrm{maxred}_\beta(\mathtt{x}(t)) & \text{(by the induction hypothesis)} \\ & \leqslant & \mathrm{maxred}_\beta(\mathtt{x}(t)[\mathtt{x}(v)/z]) & \text{(by Lemma 21 (1))} \\ & = & \mathrm{maxred}_\beta(\mathtt{x}(t\langle z := v\rangle)) & \text{(by Lemma 1 (4))} \end{array}$$

$\square$

Lemma 6 (Main lemma). *If a is decent and $\mathtt{x}(a) \in \mathcal{SN}_\beta$, then $a \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.*

Proof. By induction on $\mathrm{maxred}_\beta(\mathtt{x}(a))$. Suppose that a is decent and that if b is decent and $\mathrm{maxred}_\beta(\mathtt{x}(b)) < \mathrm{maxred}_\beta(\mathtt{x}(a))$ then $b \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. We first show that if $a \to_{\overline{\lambda}\mathtt{x}} a'$ then a' is decent. If the reduction takes place inside the body of a substitution in a then clearly a' is decent. In what follows we show that for all subterms b of a, if the reduction $b \to_{\overline{\lambda}\mathtt{x}} b'$ takes place outside the body of a substitution in a then b' is decent. This is proved by induction on the structure of b. We treat some cases.

- $b \equiv tl$, $l \to_{\overline{\lambda}\mathtt{x}} l'$ and $b' \equiv tl'$. Then l' is decent by the induction hypothesis. Therefore b' is decent.
- $b \equiv (tl)\langle z := v\rangle$ and $b' \equiv t\langle z := v\rangle l\langle z := v\rangle$. Then all bodies of substitutions in b' are also bodies of substitutions in b. Hence b' is decent.
- $b \equiv (\lambda z.t)(u :: l)$ and $b' \equiv t\langle z := u\rangle l$. In this case we crucially need the first induction hypothesis. All bodies of substitutions in t or l are also bodies of substitutions in b; we need to show that the new body of a substitution, u, is in $\mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ too. Now since u is a subterm of b, u is decent, and $\mathrm{maxred}_\beta(\mathtt{x}(u)) < \mathrm{maxred}_\beta((\lambda z.\mathtt{x}(t))(\mathtt{x}(u) :: \mathtt{x}(l))) = \mathrm{maxred}_\beta(\mathtt{x}(b)) \leqslant \mathrm{maxred}_\beta(\mathtt{x}(a))$ by Lemma 5. Therefore, by the first induction hypothesis, $u \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.

Hence if $a \to_{\overline{\lambda}\mathtt{x}} a'$ then a' is decent. Moreover, if $a \to_{\overline{\lambda}\mathtt{x}} a'$ then by Lemma 2, $\mathtt{x}(a) \xrightarrow{*}_{\beta} \mathtt{x}(a')$ and so $\text{maxred}_{\beta}(\mathtt{x}(a')) \leqslant \text{maxred}_{\beta}(\mathtt{x}(a))$, where if $\text{maxred}_{\beta}(\mathtt{x}(a')) < \text{maxred}_{\beta}(\mathtt{x}(a))$ then $a' \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ by the first induction hypothesis, otherwise $\text{maxred}_{\beta}(\mathtt{x}(a')) = \text{maxred}_{\beta}(\mathtt{x}(a))$ and we can apply the above argument to a' to show that if $a' \to_{\overline{\lambda}\mathtt{x}} a''$ then a'' is decent. Thus we see that for any a' such that $a \xrightarrow{*}_{\overline{\lambda}\mathtt{x}} a'$, a' is decent.

Now suppose that a has an infinite $\overline{\lambda}\mathtt{x}$-reduction path. By Lemma 4 there is a term a' on this path such that from a' on all reductions are void. But we just proved that a' is decent, so we have a contradiction with Proposition 6. Therefore, $a \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. □

Corollary 2 (PSN). *For any pure term $a \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \cup \mathcal{L}_{\overline{\lambda}\mathtt{x}}$, $a \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ iff $a \in \mathcal{SN}_{\beta}$.*

Proof. The only if part is by Corollary 1. Since pure terms are decent, we have the if part by Lemma 6. □

7 Strong Normalization

In this section we prove that all typable $\overline{\lambda}\mathtt{x}$-terms are strongly normalizing. For this we use the reducibility method adapted to explicit substitution calculus and to Herbelin-style calculus. Here we consider reducibility sets only over $\mathcal{T}_{\overline{\lambda}\mathtt{x}}$ and not over $\mathcal{L}_{\overline{\lambda}\mathtt{x}}$, which is sufficient to prove our main result.

Definition 5. *For each type A, the set $\mathcal{R}^{A}$ is defined inductively as follows:*

$$\mathcal{R}^{\varphi} =_{def} \mathcal{SN}_{\overline{\lambda}\mathtt{x}} \cap \mathcal{T}_{\overline{\lambda}\mathtt{x}}$$
$$\mathcal{R}^{B \supset C} =_{def} \{t \in \mathcal{T}_{\overline{\lambda}\mathtt{x}} \mid \forall v \in \mathcal{R}^{B}[t(v :: []) \in \mathcal{R}^{C}]\}$$

where φ is a type variable.

In the following, we abbreviate a $\overline{\lambda}\mathtt{x}$-term $t_1 :: (t_2 :: \cdots (t_n :: []) \ldots)$ to $t_1 :: t_2 :: \cdots :: t_n :: []$, and $(\ldots((tl_1)l_2)\ldots)l_n$ to $tl_1 l_2 \ldots l_n$.

Lemma 7. *For every type A and every variable x,*

1. *$\mathcal{R}^{A} \subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.*
2. *If $x(u_1 :: \cdots :: u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$, then $(x[])(u_1 :: []) \ldots (u_n :: []) \in \mathcal{R}^{A}$.*

Proof. By simultaneous induction on the structure of A.

i) A is a type variable φ.
   1. By the definition of $\mathcal{R}^{\varphi}$.
   2. Let $x(u_1 :: \cdots :: u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. Then $(x[])(u_1 :: []) \ldots (u_n :: [])$ is decent, and $\mathtt{x}(x(u_1 :: \cdots :: u_n :: [])) \in \mathcal{SN}_{\beta}$ by Corollary 1. Since $\mathtt{x}((x[])(u_1 :: []) \ldots (u_n :: [])) \equiv \mathtt{x}(x(u_1 :: \cdots :: u_n :: []))$, we have $(x[])(u_1 :: []) \ldots (u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ by Lemma 6. Hence $(x[])(u_1 :: []) \ldots (u_n :: []) \in \mathcal{R}^{\varphi}$.

ii) A is of the form $B \supset C$.

1. Let $t \in \mathcal{R}^{B \supset C}$. By the induction hypothesis for the second item, $x[] \in \mathcal{R}^B$ and so $t(x[] :: []) \in \mathcal{R}^C$. By the induction hypothesis for the first item, $t(x[] :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. Hence $t \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.
2. Let $x(u_1 :: \cdots :: u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ and let $v \in \mathcal{R}^B$. By the induction hypothesis for the first item, $v \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ and so $x(u_1 :: \cdots :: u_n :: v :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. By the induction hypothesis for the second item, $(x[])(u_1 :: []) \ldots (u_n :: [])(v :: []) \in \mathcal{R}^C$. Hence $(x[])(u_1 :: []) \ldots (u_n :: []) \in \mathcal{R}^{B \supset C}$ if $x(u_1 :: \cdots :: u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. □

Henceforth we use the result of Lemma 7 (1) without reference.

The next two lemmas are essential to our reducibility proof in which the reducibility sets need to be closed under certain expansion with respect to $\overline{\lambda}\mathtt{x}$-reduction.

Lemma 8. *Let* $\mathtt{x}(s) \equiv \mathtt{x}(t)$ *and* $t \in \mathcal{R}^A$. *If* s *is decent (in particular, if every substitution body in* s *is a subterm of* t*), then* $s \in \mathcal{R}^A$.

Proof. By induction on the structure of A.

i) A is a type variable φ. Let $\mathtt{x}(s) \equiv \mathtt{x}(t)$ and $t \in \mathcal{R}^\varphi (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$. Then $\mathtt{x}(s) \equiv \mathtt{x}(t) \in \mathcal{SN}_\beta$ by Corollary 1. If s is decent, then $s \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$ by Lemma 6, and hence $s \in \mathcal{R}^\varphi$.

ii) A is of the form $B \supset C$. We show that for any $v \in \mathcal{R}^B$, $s(v :: []) \in \mathcal{R}^C$. Suppose $v \in \mathcal{R}^B (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$. Since $t \in \mathcal{R}^{B \supset C}$ by assumption, we have $t(v :: []) \in \mathcal{R}^C$. By Lemma 1 (3), $\mathtt{x}(s(v :: [])) \equiv \{\mathtt{x}(s)\}\mathtt{x}(v :: []) \equiv \{\mathtt{x}(t)\}\mathtt{x}(v :: []) \equiv \mathtt{x}(t(v :: []))$. Since s and v are decent, so is $s(v :: [])$. Hence by the induction hypothesis, we have $s(v :: []) \in \mathcal{R}^C$. □

Lemma 9. *If* $(t\langle x := v\rangle)(u_1 :: []) \ldots (u_n :: []) \in \mathcal{R}^A$, *then* $(\lambda x.t)(v :: [])(u_1 :: []) \ldots (u_n :: []) \in \mathcal{R}^A$.

Proof. By induction on the structure of A.

i) A is a type variable φ. By Lemma 8, it suffices to show that if $(t\langle x := v\rangle)(u_1 :: \cdots :: u_n :: []) \in \mathcal{R}^\varphi$ then $(\lambda x.t)(v :: u_1 :: \cdots :: u_n :: []) \in \mathcal{R}^\varphi$. Suppose $(t\langle x := v\rangle)(u_1 :: \cdots :: u_n :: []) \in \mathcal{R}^\varphi (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$. Then $t, v, u_1, \ldots, u_n \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. Hence any infinite reduction sequence starting from $(\lambda x.t)(v :: u_1 :: \cdots :: u_n :: [])$ must have the form,

$$\begin{aligned}(\lambda x.t)(v :: u_1 :: \cdots :: u_n :: []) &\xrightarrow{*}_{\overline{\lambda}\mathtt{x}} (\lambda x.t')(v' :: u'_1 :: \cdots :: u'_n :: []) \\ &\rightarrow_{Beta} (t'\langle x := v'\rangle)(u'_1 :: \cdots :: u'_n :: []) \\ &\rightarrow_{\overline{\lambda}\mathtt{x}} \cdots\end{aligned}$$

where $t \xrightarrow{*}_{\overline{\lambda}\mathtt{x}} t'$, $v \xrightarrow{*}_{\overline{\lambda}\mathtt{x}} v'$ and $u_i \xrightarrow{*}_{\overline{\lambda}\mathtt{x}} u'_i$ for $1 \leqslant i \leqslant n$. But then there is an infinite reduction sequence,

$$\begin{aligned}(t\langle x := v\rangle)(u_1 :: \cdots :: u_n :: []) &\xrightarrow{*}_{\overline{\lambda}\mathtt{x}} (t'\langle x := v'\rangle)(u'_1 :: \cdots :: u'_n :: []) \\ &\rightarrow_{\overline{\lambda}\mathtt{x}} \cdots\end{aligned}$$

contradicting the hypothesis. Thus $(\lambda x.t)(v :: u_1 :: \cdots :: u_n :: []) \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}} \cap \mathcal{T}_{\overline{\lambda}\mathtt{x}} = \mathcal{R}^\varphi$.

ii) A is of the form $B \supset C$. Easily seen by the induction hypothesis for C. □

Now we prove the reducibility lemma using the results we obtained so far.

Lemma 10.

1. *Let $x_1 : A_1, \ldots, x_n : A_n; - \vdash t : B$ and let $u_i \in \mathcal{R}^{A_i}$ for each $1 \leqslant i \leqslant n$ where $x_j \notin FV(u_i)$ for all $1 \leqslant j \leqslant n$. Then $t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$.*
2. *Let $x_1 : A_1, \ldots, x_n : A_n; B \vdash l : C$ and let $u_i \in \mathcal{R}^{A_i}$ for each $1 \leqslant i \leqslant n$ where $x_j \notin FV(u_i)$ for all $1 \leqslant j \leqslant n$. Then for any $t \in \mathcal{R}^B$, $t(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$.*

Proof. Both items are proved simultaneously by induction on the structure of derivations. Let Γ denote $x_1 : A_1, \ldots, x_n : A_n$.

i)
$$\frac{}{\Gamma; A \vdash [] : A}\ Ax$$

We show that for any $t \in \mathcal{R}^A$, $t([]\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^A$. Suppose $t \in \mathcal{R}^A (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$. By assumption, we have $u_i \in \mathcal{R}^{A_i} (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$ for each $1 \leqslant i \leqslant n$. Hence $t([]\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)$ is decent. Now we have $\mathtt{x}(t([]\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)) \equiv \mathtt{x}(t[]) \equiv \{\mathtt{x}(t)\}[] \equiv \mathtt{x}(t)$ using Lemma 1 (3) and Lemma 12 (2). Hence $t([]\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^A$ by Lemma 8.

ii)
$$\frac{\Gamma, x : A; A \vdash l : B}{\Gamma, x : A; - \vdash xl : B}\ Der$$

We show that

$$(xl)\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$$

where $u \in \mathcal{R}^A$ and $x_j \notin FV(u)$ for all $1 \leqslant j \leqslant n$. By the induction hypothesis, $u(l\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^B$. Now,

$$\begin{aligned}
&(xl)\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle \\
\xrightarrow{*}_{\mathtt{x}}&(xl\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle)\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle \\
\rightarrow_{\mathtt{x}}&(ul\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle)\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle \\
\xrightarrow{*}_{\mathtt{x}}&u\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle l\langle x_1 := u_1\rangle \ldots \langle x := u\rangle \ldots \langle x_n := u_n\rangle
\end{aligned}$$

and hence

$$\begin{aligned}
&\mathtt{x}((xl)\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle) \\
\equiv\ &\mathtt{x}(u\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle l\langle x_1 := u_1\rangle \ldots \langle x := u\rangle \ldots \langle x_n := u_n\rangle) \\
\equiv\ &\{\mathtt{x}(u)[\mathtt{x}(u_{i+1})/x_{i+1}] \ldots [\mathtt{x}(u_n)/x_n]\}\mathtt{x}(l)[\mathtt{x}(u_1)/x_1] \ldots [\mathtt{x}(u)/x] \ldots [\mathtt{x}(u_n)/x_n] \\
&\hfill \text{(by Lemma 1)} \\
\equiv\ &\{\mathtt{x}(u)\}\mathtt{x}(l)[\mathtt{x}(u_1)/x_1] \ldots [\mathtt{x}(u)/x] \ldots [\mathtt{x}(u_n)/x_n] \qquad \text{(by Lemma 19)} \\
\equiv\ &\mathtt{x}(u(l\langle x_1 := u_1\rangle \ldots \langle x := u\rangle \ldots \langle x_n := u_n\rangle)). \qquad \text{(by Lemma 1)}
\end{aligned}$$

Therefore, by Lemma 8, we have

$$(xl)\langle x_1 := u_1\rangle \ldots \langle x_i := u_i\rangle\langle x := u\rangle\langle x_{i+1} := u_{i+1}\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B.$$

iii)

$$\frac{\Gamma; - \vdash v : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash v :: l : C}\ L\supset$$

We show that for any $t \in \mathcal{R}^{A\supset B}$, $t((v :: l)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$. Suppose $t \in \mathcal{R}^{A\supset B}$. By the induction hypothesis for the left premise, $v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^A$, and so $t(v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle :: []) \in \mathcal{R}^B$. Then by the induction hypothesis for the right premise,

$$(t(v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle :: []))(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C.$$

Since $\mathtt{x}(t((v :: l)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)) \equiv \mathtt{x}((t(v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle :: []))(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle))$, we have $t((v :: l)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$ by applying Lemma 8.

iv)

$$\frac{\Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash \lambda x.t : A \supset B}\ R\supset$$

We show that $(\lambda x.t)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^{A\supset B}$. Suppose $v \in \mathcal{R}^A$. Then by the induction hypothesis, $t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\langle x := v\rangle \in \mathcal{R}^B$. By Lemma 9, $(\lambda x.t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)(v :: []) \in \mathcal{R}^B$. Hence $\lambda x.t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^{A\supset B}$, and by applying Lemma 8 we have $(\lambda x.t)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^{A\supset B}$.

v)

$$\frac{\Gamma; B \vdash l : A \quad \Gamma; A \vdash l' : C}{\Gamma; B \vdash ll' : C}\ Cut_1$$

We show that for any $t \in \mathcal{R}^B$, $t((ll')\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$. Suppose $t \in \mathcal{R}^B$. Then by the induction hypothesis for the left premise, $t(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^A$, and by the induction hypothesis for the right premise, $(t(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle))(l'\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$. By applying Lemma 8, we have $t((ll')\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$.

vi)

$$\frac{\Gamma; - \vdash v : A \quad \Gamma, x : A; B \vdash l : C}{\Gamma; B \vdash l\langle x := v\rangle : C}\ Cut_2$$

We show that for any $t \in \mathcal{R}^B$, $t(l\langle x := v\rangle\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$. Suppose $t \in \mathcal{R}^B$. By the induction hypothesis for the left premise, $v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^A$. Then by the induction hypothesis for the right premise, $t(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\langle x := v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\rangle) \in \mathcal{R}^C$. Using Lemma 1 and the substitution lemma (Lemma 20) we have $\mathtt{x}(t(l\langle x := v\rangle\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)) \equiv \mathtt{x}(t(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\langle x := v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\rangle))$, and hence $t(l\langle x := v\rangle\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^C$ by Lemma 8.

vii)

$$\frac{\Gamma; - \vdash t : A \quad \Gamma; A \vdash l : B}{\Gamma; - \vdash tl : B}\ Cut_3$$

We show that $(tl)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$. By the induction hypothesis for the left premise, $t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^A$, and by the induction hypothesis for the right premise, $(t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle)(l\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle) \in \mathcal{R}^B$. By applying Lemma 8, we have $(tl)\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$.

viii)

$$\frac{\Gamma; - \vdash v : A \quad \Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash t\langle x := v\rangle : B}\ Cut_4$$

We show that $t\langle x := v\rangle\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$. By the induction hypothesis for the left premise, $v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^A$. Then by the induction hypothesis for the right premise, $t\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\langle x := v\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle\rangle \in \mathcal{R}^B$. Using the substitution lemma (Lemma 20) and Lemma 8, we have $t\langle x := v\rangle\langle x_1 := u_1\rangle \ldots \langle x_n := u_n\rangle \in \mathcal{R}^B$. □

Theorem 3.

1. *Let $\Gamma; - \vdash t : B$ for some Γ and B. Then $t \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.*
2. *Let $\Gamma; B \vdash l : C$ for some Γ, B and C. Then $l \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.*

Proof. 1. Suppose Γ is $\{x_1 : A_1, \ldots, x_n : A_n\}$. By Lemma 10 (1), $t\langle x_1 := y_1[]\rangle \ldots \langle x_n := y_n[]\rangle \in \mathcal{R}^B (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$ where each y_i is fresh and $y_i[] \in \mathcal{R}^{A_i}$ by Lemma 7 (2). Hence $t \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$.
2. Similarly, $(z[])(l\langle x_1 := y_1[]\rangle \ldots \langle x_n := y_n[]\rangle) \in \mathcal{R}^C (\subseteq \mathcal{SN}_{\overline{\lambda}\mathtt{x}})$ where z is also fresh. Hence $l \in \mathcal{SN}_{\overline{\lambda}\mathtt{x}}$. □

The above theorem shows that the cut-elimination procedure defined by the reduction rules of $\overline{\lambda}\mathtt{x}$-calculus is strongly normalizing. Also, by Corollary 1, we have strong normalization for typable pure terms with respect to β-reduction, and thus strong normalization for typable terms in the usual λ-calculus as well.

8 Conclusion

In this paper we presented a direct proof of strong normalization for the typable terms of the extended Herbelin's calculus. The main lemma was useful for both the inductive proof of PSN with respect to β-reduction on pure terms and the reducibility proof of strong normalization for typable terms. Our reducibility method seems to be helpful in investigating other reduction properties and semantical aspects of the calculus.

In the literature [13, 16] there are reducibility methods for other calculi with explicit substitutions. The relationship between them and ours will be investigated in future work.

Since the extended Herbelin's calculus clarifies how to simulate β-reduction by cut-elimination, it can be viewed as a basis for understanding computational meanings of various cut-elimination procedures. It would also be interesting to use the calculus for studies of integrating proof search and proof normalization.

Acknowledgements. I am grateful to Takafumi Sakurai for useful discussions and encouragement. I also thank Roy Dyckhoff, Stéphane Lengrand and the anonymous referees for valuable comments.

References

1. M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 1:375–416, 1991.
2. R. Bloo. Preservation of strong normalisation for explicit substitution. Computing Science Report 95-08, Eindhoven University of Technology, 1995.
3. R. Bloo. *Preservation of Termination for Explicit Substitution*. PhD thesis, Eindhoven University of Technology, 1997.
4. R. Bloo and H. Geuvers. Explicit substitution: on the edge of strong normalization. *Theoretical Computer Science*, 211:375–395, 1999.
5. R. Bloo and K. H. Rose. Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection. In *Proceedings of CSN'95 (Computing Science in the Netherlands)*, pages 62–72, 1995.
6. E. Bonelli. Perpetuality in a named lambda calculus with explicit substitutions. *Mathematical Structures in Computer Science*, 11:47–90, 2001.
7. D. Dougherty and P. Lescanne. Reductions, intersection types, and explicit substitutions. *Mathematical Structures in Computer Science*, 13:55–85, 2003.
8. R. Dyckhoff and L. Pinto. Proof search in constructive logic. In S. B. Cooper and J. K. Truss, editors, *Proceedings of the Logic Colloquium 1997*, London Mathematical Society Lecture Note Series 258, pages 53–65. Cambridge University Press, 1997.
9. R. Dyckhoff and C. Urban. Strong normalisation of Herbelin's explicit substitution calculus with substitution propagation. In *Proceedings of WESTAPP'01*, pages 26–45, 2001.
10. R. Dyckhoff and C. Urban. Strong normalization of Herbelin's explicit substitution calculus with substitution propagation. *Journal of Logic and Computation*, 13:689–706, 2003.
11. J. Espírito Santo. Revisiting the correspondence between cut elimination and normalisation. In *Proceedings of ICALP'00*, Lecture Notes in Computer Science 1853, pages 600–611. Springer-Verlag, 2000.
12. H. Herbelin. A λ-calculus structure isomorphic to Gentzen-style sequent calculus structure. In *Proceedings of CSL'94*, Lecture Notes in Computer Science 933, pages 61–75. Springer-Verlag, 1995.
13. H. Herbelin. Explicit substitutions and reducibility. *Journal of Logic and Computation*, 11:429–449, 2001.
14. F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed λ-calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic*, 42:59–87, 2003.
15. P.-A. Melliès. Typed λ-calculi with explicit substitutions may not terminate. In *Proceedings of TLCA'95*, Lecture Notes in Computer Science 902, pages 328–334. Springer-Verlag, 1995.
16. E. Ritter. Characterising explicit substitutions which preserve termination. In *Proceedings of TLCA'99*, Lecture Notes in Computer Science 1581, pages 325–339. Springer-Verlag, 1999.

17. W. W. Tait. Intensional interpretations of functionals of finite type I. *The Journal of Symbolic Logic*, 32:198–212, 1967.
18. A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge Tracts in Theoretical Computer Science 43. Cambridge University Press, 2nd edition, 2000.

A Cut-Elimination Steps for Typing Derivations

In this section we display the cut-elimination steps corresponding to $\overline{\lambda}\mathbf{x}$-reduction for typable terms. First we consider the following lemma, which is needed in the case (4c) below.

Lemma 11. *Let $\Gamma; - \vdash t : A$ where x does not appear in Γ. Then $\Gamma, x : B; - \vdash t : A$.*

Proof. By induction on the structure of derivations. □

Cut-Elimination Steps for Typing Derivations

$(Beta)$ $\quad (\lambda x.t)(u :: l) \to t\langle x := u\rangle l$

$$\dfrac{\dfrac{\Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash \lambda x.t : A \supset B}\,R\supset \qquad \dfrac{\Gamma; - \vdash u : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash u :: l : C}\,L\supset}{\Gamma; - \vdash (\lambda x.t)(u :: l) : C}\,Cut_3$$

$$\to \quad \dfrac{\dfrac{\Gamma; - \vdash u : A \quad \Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash t\langle x := u\rangle : B}\,Cut_4 \qquad \Gamma; B \vdash l : C}{\Gamma; - \vdash t\langle x := u\rangle l : C}\,Cut_3$$

(1a) $\quad [\,]l \to l$

$$\dfrac{\dfrac{}{\Gamma; A \vdash [\,] : A}\,Ax \qquad \Gamma; A \vdash l : C}{\Gamma; A \vdash [\,]l : C}\,Cut_1 \quad \to \quad \Gamma; A \vdash l : C$$

(1b) $\quad (u :: l)l' \to u :: (ll')$

$$\dfrac{\dfrac{\Gamma; - \vdash u : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash u :: l : C}\,L\supset \qquad \Gamma; C \vdash l' : D}{\Gamma; A \supset B \vdash (u :: l)l' : D}\,Cut_1$$

$$\to \quad \dfrac{\Gamma; - \vdash u : A \qquad \dfrac{\Gamma; B \vdash l : C \quad \Gamma; C \vdash l' : D}{\Gamma; B \vdash ll' : D}\,Cut_1}{\Gamma; A \supset B \vdash u :: (ll') : D}\,L\supset$$

(2a) $\quad [\,]\langle x := v\rangle \to [\,]$

$$\dfrac{\Gamma; - \vdash v : A \qquad \dfrac{}{\Gamma, x : A; B \vdash [\,] : B}\,Ax}{\Gamma; B \vdash [\,]\langle x := v\rangle : B}\,Cut_2 \quad \to \quad \dfrac{}{\Gamma; B \vdash [\,] : B}\,Ax$$

(2b) $(u :: l)\langle x := v\rangle \to u\langle x := v\rangle :: l\langle x := v\rangle$

$$\dfrac{\Gamma; - \vdash v : A \quad \dfrac{\Gamma, x : A; - \vdash u : B \quad \Gamma, x : A; C \vdash l : D}{\Gamma, x : A; B \supset C \vdash u :: l : D}\, L\supset}{\Gamma; B \supset C \vdash (u :: l)\langle x := v\rangle : D}\, \mathit{Cut}_2$$

$$\to \quad \dfrac{\dfrac{\Gamma; - \vdash v : A \quad \Gamma, x : A; - \vdash u : B}{\Gamma; - \vdash u\langle x := v\rangle : B}\, \mathit{Cut}_4 \quad \dfrac{\Gamma; - \vdash v : A \quad \Gamma, x : A; C \vdash l : D}{\Gamma; C \vdash l\langle x := v\rangle : D}\, \mathit{Cut}_2}{\Gamma; B \supset C \vdash u\langle x := v\rangle :: l\langle x := v\rangle : D}\, L\supset$$

(3a) $(xl)l' \to x(ll')$

$$\dfrac{\dfrac{\Gamma, x : A; A \vdash l : B}{\Gamma, x : A; - \vdash xl : B}\, \mathit{Der} \quad \Gamma, x : A; B \vdash l' : C}{\Gamma, x : A; - \vdash (xl)l' : C}\, \mathit{Cut}_3$$

$$\to \quad \dfrac{\dfrac{\Gamma, x : A; A \vdash l : B \quad \Gamma, x : A; B \vdash l' : C}{\Gamma, x : A; A \vdash ll' : C}\, \mathit{Cut}_1}{\Gamma, x : A; - \vdash x(ll') : C}\, \mathit{Der}$$

(3b) $(\lambda y.t)[] \to \lambda y.t$

$$\dfrac{\Gamma; - \vdash \lambda y.t : A \quad \dfrac{}{\Gamma; A \vdash [] : A}\, \mathit{Ax}}{\Gamma; - \vdash (\lambda y.t)[] : A}\, \mathit{Cut}_3 \quad \to \quad \Gamma; - \vdash \lambda y.t : A$$

(4a) $(yl)\langle x := v\rangle \to yl\langle x := v\rangle \quad (y \not\equiv x)$

$$\dfrac{\Gamma, y : A; - \vdash v : B \quad \dfrac{\Gamma, x : B, y : A; A \vdash l : C}{\Gamma, x : B, y : A; - \vdash yl : C}\, \mathit{Der}}{\Gamma, y : A; - \vdash (yl)\langle x := v\rangle : C}\, \mathit{Cut}_4$$

$$\to \quad \dfrac{\dfrac{\Gamma, y : A; - \vdash v : B \quad \Gamma, x : B, y : A; A \vdash l : C}{\Gamma, y : A; A \vdash l\langle x := v\rangle : C}\, \mathit{Cut}_2}{\Gamma, y : A; - \vdash yl\langle x := v\rangle : C}\, \mathit{Der}$$

(4b) $(xl)\langle x := v\rangle \to vl\langle x := v\rangle$

$$\dfrac{\Gamma; - \vdash v : A \quad \dfrac{\Gamma, x : A; A \vdash l : B}{\Gamma, x : A; - \vdash xl : B}\, \mathit{Der}}{\Gamma; - \vdash (xl)\langle x := v\rangle : B}\, \mathit{Cut}_4$$

$$\to \quad \dfrac{\Gamma; - \vdash v : A \quad \dfrac{\Gamma; - \vdash v : A \quad \Gamma, x : A; A \vdash l : B}{\Gamma; A \vdash l\langle x := v\rangle : B}\, \mathit{Cut}_2}{\Gamma; - \vdash vl\langle x := v\rangle : B}\, \mathit{Cut}_3$$

(4c) $(\lambda y.t)\langle x := v\rangle \to \lambda y.t\langle x := v\rangle$

$$\dfrac{\Gamma; - \vdash v : A \qquad \dfrac{\Gamma, x : A, y : B; - \vdash t : C}{\Gamma, x : A; - \vdash \lambda y.t : B \supset C}\,R\supset}{\Gamma; - \vdash (\lambda y.t)\langle x := v\rangle : B \supset C}\,Cut_4$$

$$\to \quad \dfrac{\dfrac{\dfrac{}{\Gamma, y : B; - \vdash v : A}\,\textit{Lemma 11} \qquad \Gamma, x : A, y : B; - \vdash t : C}{\Gamma, y : B; - \vdash t\langle x := v\rangle : C}\,Cut_4}{\Gamma; - \vdash \lambda y.t\langle x := v\rangle : B \supset C}\,R\supset$$

(5a) $(ll')l'' \to l(l'l'')$

$$\dfrac{\dfrac{\Gamma; A \vdash l : B \qquad \Gamma; B \vdash l' : C}{\Gamma; A \vdash ll' : C}\,Cut_1 \qquad \Gamma; C \vdash l'' : D}{\Gamma; A \vdash (ll')l'' : D}\,Cut_1$$

$$\to \quad \dfrac{\Gamma; A \vdash l : B \qquad \dfrac{\Gamma; B \vdash l' : C \qquad \Gamma; C \vdash l'' : D}{\Gamma; B \vdash l'l'' : D}\,Cut_1}{\Gamma; A \vdash l(l'l'') : D}\,Cut_1$$

(5b) $(ll')\langle x := v\rangle \to l\langle x := v\rangle l'\langle x := v\rangle$

$$\dfrac{\Gamma; - \vdash v : A \qquad \dfrac{\Gamma, x : A; B \vdash l : C \qquad \Gamma, x : A; C \vdash l' : D}{\Gamma, x : A; B \vdash ll' : D}\,Cut_1}{\Gamma; B \vdash (ll')\langle x := v\rangle : D}\,Cut_2$$

$$\to \quad \dfrac{\dfrac{\Gamma; - \vdash v : A \qquad \Gamma, x : A; B \vdash l : C}{\Gamma; B \vdash l\langle x := v\rangle : C}\,Cut_2 \qquad \dfrac{\Gamma; - \vdash v : A \qquad \Gamma, x : A; C \vdash l' : D}{\Gamma; C \vdash l'\langle x := v\rangle : D}\,Cut_2}{\Gamma; B \vdash l\langle x := v\rangle l'\langle x := v\rangle : D}\,Cut_1$$

(5c) $(tl)l' \to t(ll')$

$$\dfrac{\dfrac{\Gamma; - \vdash t : A \qquad \Gamma; A \vdash l : B}{\Gamma; - \vdash tl : B}\,Cut_3 \qquad \Gamma; B \vdash l' : C}{\Gamma; - \vdash (tl)l' : C}\,Cut_3$$

$$\to \quad \dfrac{\Gamma; - \vdash t : A \qquad \dfrac{\Gamma; A \vdash l : B \qquad \Gamma; B \vdash l' : C}{\Gamma; A \vdash ll' : C}\,Cut_1}{\Gamma; - \vdash t(ll') : C}\,Cut_3$$

(5d) $(tl)\langle x := v\rangle \to t\langle x := v\rangle l\langle x := v\rangle$

$$\dfrac{\Gamma; - \vdash v : A \qquad \dfrac{\Gamma, x : A; - \vdash t : B \qquad \Gamma, x : A; B \vdash l : C}{\Gamma, x : A; - \vdash tl : C}\,Cut_3}{\Gamma; - \vdash (tl)\langle x := v\rangle : C}\,Cut_4$$

$$\to \quad \dfrac{\dfrac{\Gamma; - \vdash v : A \qquad \Gamma, x : A; - \vdash t : B}{\Gamma; - \vdash t\langle x := v\rangle : B}\,Cut_4 \qquad \dfrac{\Gamma; - \vdash v : A \qquad \Gamma, x : A; B \vdash l : C}{\Gamma; B \vdash l\langle x := v\rangle : C}\,Cut_2}{\Gamma; - \vdash t\langle x := v\rangle l\langle x := v\rangle : C}\,Cut_3$$

B Basic Properties of Pure Terms

Lemma 12. *Let t, l be pure terms. Then*

1. $l@[] \equiv l$,
2. $\{t\}[] \equiv t$.

Proof. 1. By induction on the structure of l.
2. By case analysis on the definition of $\{t\}[]$, using 1. ☐

Lemma 13. *Let l, l', l'' be pure terms. Then $l@(l'@l'') \equiv (l@l')@l''$.*

Proof. By induction on the structure of l. ☐

Lemma 14. *Let t, l, l' be pure terms. Then $\{\{t\}l\}l' \equiv \{t\}(l@l')$.*

Proof. By case analysis on the definition of $\{t\}l$.

(a) $t \equiv xl_0$. Then $\{\{xl_0\}l\}l' \equiv \{x(l_0@l)\}l' \equiv x((l_0@l)@l') \overset{13}{\equiv} x(l_0@(l@l')) \equiv \{xl_0\}(l@l')$.
(b) $t \equiv \lambda y.t_0$ and $l \equiv []$. Then $\{\{\lambda y.t_0\}[]\}l' \equiv \{\lambda y.t_0\}l' \equiv \{\lambda y.t_0\}([]@l')$.
(c) $t \equiv \lambda y.t_0$ and $l \equiv u :: l_0$. Then $\{\{\lambda y.t_0\}(u :: l_0)\}l' \equiv \{(\lambda y.t_0)(u :: l_0)\}l' \equiv (\lambda y.t_0)(u :: (l_0@l')) \equiv (\lambda y.t_0)((u :: l_0)@l') \equiv \{\lambda y.t_0\}((u :: l_0)@l')$.
(d) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $\{\{(\lambda y.t_0)(u :: l_0)\}l\}l' \equiv \{(\lambda y.t_0)(u :: (l_0@l))\}l' \equiv (\lambda y.t_0)(u :: ((l_0@l)@l')) \overset{13}{\equiv} (\lambda y.t_0)(u :: (l_0@(l@l'))) \equiv \{(\lambda y.t_0)(u :: l_0)\}(l@l')$.
☐

Lemma 15. *Let l, l', l_0 be pure terms with $l \to_\beta l'$. Then*

1. $l_0@l \to_\beta l_0@l'$,
2. $l@l_0 \to_\beta l'@l_0$.

Proof. 1. By induction on the structure of l_0.
2. By induction on the structure of l. ☐

Lemma 16. *Let t, t', l, l' be pure terms with $t \to_\beta t'$ and $l \to_\beta l'$. Then*

1. $\{t\}l \to_\beta \{t'\}l$,
2. $\{t\}l \to_\beta \{t\}l'$.

Proof. 1. By case analysis on the definition of $\{t\}l$.
   (a) $t \equiv xl_0$ and $l_0 \to_\beta l'_0$. Then $\{xl_0\}l \equiv x(l_0@l) \overset{15}{\to}_\beta x(l'_0@l) \equiv \{xl'_0\}l$.
   (b) $t \equiv \lambda y.t_0$, $t_0 \to_\beta t'_0$ and $l \equiv []$. Then $\{\lambda y.t_0\}[] \equiv \lambda y.t_0 \to_\beta \lambda y.t'_0 \equiv \{\lambda y.t'_0\}[]$.
   (c) $t \equiv \lambda y.t_0$, $t_0 \to_\beta t'_0$ and $l \equiv u :: l_0$. Then $\{\lambda y.t_0\}l \equiv (\lambda y.t_0)l \to_\beta (\lambda y.t'_0)l \equiv \{\lambda y.t'_0\}l$.
   (d) $t \equiv (\lambda y.t_0)(u :: l_0)$.
      i. The β-reduction is at the root, i.e., $(\lambda y.t_0)(u :: l_0) \to_\beta \{t_0[u/y]\}l_0$. Then $\{(\lambda y.t_0)(u :: l_0)\}l \equiv (\lambda y.t_0)(u :: (l_0@l)) \to_\beta \{t_0[u/y]\}(l_0@l) \overset{14}{\equiv} \{\{t_0[u/y]\}l_0\}l$.

ii. The β-reduction is internal, e.g., $l_0 \to_\beta l_0'$. Then $\{(\lambda y.t_0)(u :: l_0)\}l \equiv (\lambda y.t_0)(u :: (l_0@l)) \overset{15}{\to}_\beta (\lambda y.t_0)(u :: (l_0'@l)) \equiv \{(\lambda y.t_0)(u :: l_0')\}l$. The other cases are similar.

2. By case analysis on the definition of $\{t\}l$.
   (a) $t \equiv xl_0$. Then $\{xl_0\}l \equiv x(l_0@l) \overset{15}{\to}_\beta x(l_0@l') \equiv \{xl_0\}l'$.
   (b) $t \equiv \lambda y.t_0$ and $l \equiv u :: l_0$. Then $\{\lambda y.t_0\}l \equiv (\lambda y.t_0)l \to_\beta (\lambda y.t_0)l' \equiv \{\lambda y.t_0\}l'$.
   (c) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $\{(\lambda y.t_0)(u :: l_0)\}l \equiv (\lambda y.t_0)(u :: (l_0@l)) \overset{15}{\to}_\beta (\lambda y.t_0)(u :: (l_0@l')) \equiv \{(\lambda y.t_0)(u :: l_0)\}l'$. □

Lemma 17. *Let l, l', v be pure terms. Then $(l@l')[v/x] \equiv l[v/x]@l'[v/x]$.*

Proof. By induction on the structure of l. □

Lemma 18. *Let t, l, v be pure terms. Then $(\{t\}l)[v/x] \equiv \{t[v/x]\}l[v/x]$.*

Proof. By case analysis on the definition of $\{t\}l$.

(a) $t \equiv yl_0$ $(y \not\equiv x)$. Then $(\{yl_0\}l)[v/x] \equiv (y(l_0@l))[v/x] \equiv y(l_0@l)[v/x] \overset{17}{\equiv} y(l_0[v/x]@l[v/x]) \equiv \{yl_0[v/x]\}l[v/x] \equiv \{(yl_0)[v/x]\}l[v/x]$.
(b) $t \equiv xl_0$. Then $(\{xl_0\}l)[v/x] \equiv (x(l_0@l))[v/x] \equiv \{v\}(l_0@l)[v/x] \overset{17}{\equiv} \{v\}(l_0[v/x]@l[v/x]) \overset{14}{\equiv} \{\{v\}l_0[v/x]\}l[v/x] \equiv \{(xl_0)[v/x]\}l[v/x]$.
(c) $t \equiv \lambda y.t_0$ and $l \equiv []$. Then $(\{\lambda y.t_0\}[])[v/x] \equiv (\lambda y.t_0)[v/x] \equiv \lambda y.t_0[v/x] \equiv \{\lambda y.t_0[v/x]\}[] \equiv \{(\lambda y.t_0)[v/x]\}[][v/x]$.
(d) $t \equiv \lambda y.t_0$ and $l \equiv u :: l_0$. Then $(\{\lambda y.t_0\}(u :: l_0))[v/x] \equiv ((\lambda y.t_0)(u :: l_0))[v/x] \equiv (\lambda y.t_0[v/x])(u[v/x] :: l_0[v/x]) \equiv \{\lambda y.t_0[v/x]\}(u[v/x] :: l_0[v/x]) \equiv \{(\lambda y.t_0)[v/x]\}(u :: l_0)[v/x]$.
(e) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $(\{(\lambda y.t_0)(u :: l_0)\}l)[v/x] \equiv ((\lambda y.t_0)(u :: (l_0@l)))[v/x] \equiv (\lambda y.t_0[v/x])(u[v/x] :: (l_0@l)[v/x]) \overset{17}{\equiv} (\lambda y.t_0[v/x])(u[v/x] :: (l_0[v/x]@l[v/x])) \equiv \{(\lambda y.t_0[v/x])(u[v/x] :: l_0[v/x])\}l[v/x] \equiv \{((\lambda y.t_0)(u :: l_0))[v/x]\}l[v/x]$. □

Lemma 19. *Let t, l, v be pure terms.*

1. *If $x \notin FV(t)$, then $t[v/x] \equiv t$.*
2. *If $x \notin FV(l)$, then $l[v/x] \equiv l$.*

Proof. By simultaneous induction on the structure of t or l. □

Lemma 20 (Substitution lemma). *Let t, l, v, u be pure terms. If $x \not\equiv y$ and $x \notin FV(u)$, then*

1. *$t[v/x][u/y] \equiv t[u/y][v[u/y]/x]$,*
2. *$l[v/x][u/y] \equiv l[u/y][v[u/y]/x]$.*

Proof. By simultaneous induction on the structure of t or l.

1. (a) $t \equiv zl_0$ $(z \not\equiv x, y)$. Then $(zl_0)[v/x][u/y] \equiv (zl_0[v/x])[u/y] \equiv zl_0[v/x][u/y] \overset{IH}{\equiv} zl_0[u/y][v[u/y]/x] \equiv (zl_0)[u/y][v[u/y]/x]$.

(b) $t \equiv xl_0$. Then $(xl_0)[v/x][u/y] \equiv (\{v\}l_0[v/x])[u/y] \overset{18}{\equiv} \{v[u/y]\}l_0[v/x][u/y] \overset{IH}{\equiv} \{v[u/y]\}l_0[u/y][v[u/y]/x] \equiv (xl_0[u/y])[v[u/y]/x] \equiv (xl_0)[u/y][v[u/y]/x]$.

(c) $t \equiv yl_0$. Then $(yl_0)[v/x][u/y] \equiv (yl_0[v/x])[u/y] \equiv \{u\}l_0[v/x][u/y] \overset{IH}{\equiv} \{u\}l_0[u/y][v[u/y]/x] \overset{19}{\equiv} \{u[v[u/y]/x]\}l_0[u/y][v[u/y]/x] \overset{18}{\equiv} (\{u\}l_0[u/y])[v[u/y]/x] \equiv (yl_0)[u/y][v[u/y]/x]$.

(d) $t \equiv \lambda z.t_0$. Then $(\lambda z.t_0)[v/x][u/y] \equiv (\lambda z.t_0[v/x])[u/y] \equiv \lambda z.t_0[v/x][u/y] \overset{IH}{\equiv} \lambda z.t_0[u/y][v[u/y]/x] \equiv (\lambda z.t_0)[u/y][v[u/y]/x]$.

(e) $t \equiv (\lambda z.t_0)(u :: l_0)$. Then $((\lambda z.t_0)(u :: l_0))[v/x][u/y] \equiv ((\lambda z.t_0[v/x])(u[v/x] :: l_0[v/x]))[u/y] \equiv (\lambda z.t_0[v/x][u/y])(u[v/x][u/y] :: l_0[v/x][u/y]) \overset{IH}{\equiv} (\lambda z.t_0[u/y][v[u/y]/x])(u[u/y][v[u/y]/x] :: l_0[u/y][v[u/y]/x]) \equiv ((\lambda z.t_0)(u :: l_0))[u/y][v[u/y]/x]$.

2. (a) $l \equiv []$. Then $[][v/x][u/y] \equiv [] \equiv [][u/y][v[u/y]/x]$.

(b) $l \equiv u :: l_0$. Then $(u :: l_0)[v/x][u/y] \equiv (u[v/x] :: l_0[v/x])[u/y] \equiv u[v/x][u/y] :: l_0[v/x][u/y] \overset{IH}{\equiv} u[u/y][v[u/y]/x] :: l_0[u/y][v[u/y]/x] \equiv (u :: l_0)[u/y][v[u/y]/x]$.

□

Lemma 21. *Let t, t', l, l', v be pure terms with $t \to_\beta t'$ and $l \to_\beta l'$. Then*

1. *$t[v/x] \to_\beta t'[v/x]$,*
2. *$l[v/x] \to_\beta l'[v/x]$.*

Proof. By simultaneous induction on the structure of t or l, and case analysis.

1. (a) $t \equiv yl_0$ $(y \not\equiv x)$ and $l_0 \to_\beta l'_0$. Then $(yl_0)[v/x] \equiv yl_0[v/x] \overset{IH}{\to}_\beta yl'_0[v/x] \equiv (yl'_0)[v/x]$.

(b) $t \equiv xl_0$ and $l_0 \to_\beta l'_0$. Then $(xl_0)[v/x] \equiv \{v\}l_0[v/x] \overset{IH,16}{\longrightarrow}_\beta \{v\}l'_0[v/x] \equiv (xl'_0)[v/x]$.

(c) $t \equiv \lambda y.t_0$ and $t_0 \to_\beta t'_0$. Then $(\lambda y.t_0)[v/x] \equiv \lambda y.t_0[v/x] \overset{IH}{\to}_\beta \lambda y.t'_0[v/x] \equiv (\lambda y.t'_0)[v/x]$.

(d) $t \equiv (\lambda y.t_0)(u :: l_0)$.

i. The β-reduction is at the root, i.e., $(\lambda y.t_0)(u :: l_0) \to_\beta \{t_0[u/y]\}l_0$. Then $((\lambda y.t_0)(u :: l_0))[v/x] \equiv (\lambda y.t_0[v/x])(u[v/x] :: l_0[v/x]) \to_\beta \{t_0[v/x][u[v/x]/y]\}l_0[v/x] \overset{20}{\equiv} \{t_0[u/y][v/x]\}l_0[v/x] \overset{18}{\equiv} (\{t_0[u/y]\}l_0)[v/x]$.

ii. The β-reduction is internal, e.g., $l_0 \to_\beta l'_0$. Then $((\lambda y.t_0)(u :: l_0))[v/x] \equiv (\lambda y.t_0[v/x])(u[v/x] :: l_0[v/x]) \overset{IH}{\to}_\beta (\lambda y.t_0[v/x])(u[v/x] :: l'_0[v/x]) \equiv ((\lambda y.t_0)(u :: l'_0))[v/x]$. The other cases are similar.

2. (a) $l \equiv u :: l_0$ and $u \to_\beta u'$. Then $(u :: l_0)[v/x] \equiv u[v/x] :: l_0[v/x] \overset{IH}{\to}_\beta u'[v/x] :: l_0[v/x] \equiv (u' :: l_0)[v/x]$.

(b) $l \equiv u :: l_0$ and $l_0 \to_\beta l'_0$. Similar to the previous case. □

Lemma 22. *Let t, l, v, v' be pure terms with $v \to_\beta v'$. Then*

1. *$t[v/x] \overset{*}{\to}_\beta t[v'/x]$,*
2. *$l[v/x] \overset{*}{\to}_\beta l[v'/x]$.*

Proof. By simultaneous induction on the structure of t or l.

1. (a) $t \equiv yl_0$ $(y \not\equiv x)$. Then $(yl_0)[v/x] \equiv yl_0[v/x] \stackrel{IH*}{\to}_\beta yl_0[v'/x] \equiv (yl_0)[v'/x]$.
   (b) $t \equiv xl_0$. Then $(xl_0)[v/x] \equiv \{v\}l_0[v/x] \stackrel{16}{\to}_\beta \{v'\}l_0[v/x] \stackrel{IH,16*}{\longrightarrow}_\beta \{v'\}l_0[v'/x] \equiv (xl_0)[v'/x]$.
   (c) $t \equiv \lambda y.t_0$. Then $(\lambda y.t_0)[v/x] \equiv \lambda y.t_0[v/x] \stackrel{IH*}{\to}_\beta \lambda y.t_0[v'/x] \equiv (\lambda y.t_0)[v'/x]$.
   (d) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $((\lambda y.t_0)(u :: l_0))[v/x] \equiv (\lambda y.t_0[v/x])(u[v/x] :: l_0[v/x]) \stackrel{IH*}{\to}_\beta (\lambda y.t_0[v'/x])(u[v'/x] :: l_0[v'/x]) \equiv ((\lambda y.t_0)(u :: l_0))[v'/x]$.
2. (a) $l \equiv []$. Then $[][v/x] \equiv [] \equiv [][v'/x]$.
   (b) $l \equiv u :: l_0$. Then $(u :: l_0)[v/x] \equiv u[v/x] :: l_0[v/x] \stackrel{IH*}{\to}_\beta u[v'/x] :: l_0[v'/x] \equiv (u :: l_0)[v'/x]$. □

Lemma 23. *Let t, l, v, v' be pure terms with $v \to_\beta v'$.*

1. *If $x \in FV(t)$, then $t[v/x] \stackrel{+}{\to}_\beta t[v'/x]$.*
2. *If $x \in FV(l)$, then $l[v/x] \stackrel{+}{\to}_\beta l[v'/x]$.*

Proof. By a similar induction to the proof of Lemma 22. □

C Isomorphism between Pure Terms and λ-calculus

Lemma 24. $\Theta'(M, l@l') \equiv \Theta'(\Theta'(M, l), l')$.

Proof. By induction on the structure of l.

(a) $l \equiv []$. Then $\Theta'(M, []@l') \equiv \Theta'(M, l') \equiv \Theta'(\Theta'(M, []), l')$.
(b) $l \equiv u :: l_0$. Then $\Theta'(M, (u :: l_0)@l') \equiv \Theta'(M, u :: (l_0@l')) \equiv \Theta'(M\Theta(u), l_0@l') \stackrel{IH}{\equiv} \Theta'(\Theta'(M\Theta(u), l_0), l') \equiv \Theta'(\Theta'(M, u :: l_0), l')$. □

Lemma 25. $\Theta(\{t\}l) \equiv \Theta'(\Theta(t), l)$.

Proof. By case analysis on the definition of $\{t\}l$.

(a) $t \equiv xl_0$. Then $\Theta(\{xl_0\}l) \equiv \Theta(x(l_0@l)) \equiv \Theta'(x, l_0@l) \stackrel{24}{\equiv} \Theta'(\Theta'(x, l_0), l) \equiv \Theta'(\Theta(xl_0), l)$.
(b) $t \equiv \lambda y.t_0$ and $l \equiv []$. Then $\Theta(\{\lambda y.t_0\}[]) \equiv \Theta(\lambda y.t_0) \equiv \Theta'(\Theta(\lambda y.t_0), [])$.
(c) $t \equiv \lambda y.t_0$ and $l \equiv u :: l_0$. Then $\Theta(\{\lambda y.t_0\}(u :: l_0)) \equiv \Theta((\lambda y.t_0)(u :: l_0)) \equiv \Theta'(\lambda y.\Theta(t_0), u :: l_0) \equiv \Theta'(\Theta(\lambda y.t_0), u :: l_0)$.
(d) $t \equiv (\lambda y.t_0)(u :: l_0)$. Then $\Theta(\{(\lambda y.t_0)(u :: l_0)\}l) \equiv \Theta((\lambda y.t_0)(u :: (l_0@l))) \equiv \Theta'(\lambda y.\Theta(t_0), u :: (l_0@l)) \equiv \Theta'(\lambda y.\Theta(t_0), (u :: l_0)@l) \stackrel{24}{\equiv} \Theta'(\Theta'(\lambda y.\Theta(t_0), u :: l_0), l) \equiv \Theta'(\Theta((\lambda y.t_0)(u :: l_0)), l)$. □

Proposition 7. *$\Theta \circ \Psi = id$ and $\Psi \circ \Theta = id$.*

Proof. The first part is by induction on the structure of λ-terms.

(a) $M \equiv x$. Then $(\Theta \circ \Psi)(x) \equiv \Theta(x[]) \equiv \Theta'(x, []) \equiv x$.

(b) $M \equiv M_0 M_1$. Then $(\Theta \circ \Psi)(M_0 M_1) \equiv \Theta(\{\Psi(M_0)\}(\Psi(M_1) :: [])) \stackrel{25}{\equiv} \Theta'(\Theta(\Psi(M_0)), \Psi(M_1) :: []) \stackrel{IH}{\equiv} \Theta'(M_0, \Psi(M_1) :: []) \equiv \Theta'(M_0 \Theta(\Psi(M_1)), []) \stackrel{IH}{\equiv} \Theta'(M_0 M_1, []) \equiv M_0 M_1$.

(c) $M \equiv \lambda x.M_0$. Then $(\Theta \circ \Psi)(\lambda x.M_0) \equiv \Theta(\lambda x.\Psi(M_0)) \equiv \lambda x.\Theta(\Psi(M_0)) \stackrel{IH}{\equiv} \lambda x.M_0$.

For the second part we prove the following by simultaneous induction on the structure of pure terms:

1. $(\Psi \circ \Theta)(t) \equiv t$,
2. $(\Psi \circ \Theta')(M, l) \equiv \{\Psi(M)\}l$.

1. (a) $t \equiv xl$. Then $(\Psi \circ \Theta)(xl) \equiv \Psi(\Theta'(x, l)) \stackrel{IH}{\equiv} \{\Psi(x)\}l \equiv \{x[]\}l \equiv x([]@l) \equiv xl$.
   (b) $t \equiv \lambda x.t_0$. Then $(\Psi \circ \Theta)(\lambda x.t_0) \equiv \Psi(\lambda x.\Theta(t_0)) \equiv \lambda x.\Psi(\Theta(t_0)) \stackrel{IH}{\equiv} \lambda x.t_0$.
   (c) $t \equiv (\lambda x.t_0)(u :: l_0)$. Then $(\Psi \circ \Theta)((\lambda x.t_0)(u :: l_0)) \equiv \Psi(\Theta'(\lambda x.\Theta(t_0), u :: l_0)) \stackrel{IH}{\equiv} \{\Psi(\lambda x.\Theta(t_0))\}(u :: l_0) \equiv \{\lambda x.\Psi(\Theta(t_0))\}(u :: l_0) \stackrel{IH}{\equiv} \{\lambda x.t_0\}(u :: l_0) \equiv (\lambda x.t_0)(u :: l_0)$.
2. (a) $l \equiv []$. Then $(\Psi \circ \Theta')(M, []) \equiv \Psi(M) \stackrel{12}{\equiv} \{\Psi(M)\}[]$.
   (b) $l \equiv u :: l_0$. Then $(\Psi \circ \Theta')(M, u :: l_0) \equiv (\Psi \circ \Theta')(M\Theta(u), l_0) \stackrel{IH}{\equiv} \{\Psi(M\Theta(u))\}l_0 \equiv \{\{\Psi(M)\}(\Psi(\Theta(u)) :: [])\}l_0 \stackrel{IH}{\equiv} \{\{\Psi(M)\}(u :: [])\}l_0 \stackrel{14}{\equiv} \{\Psi(M)\}((u :: [])@l_0) \equiv \{\Psi(M)\}(u :: l_0)$. □

Lemma 26. $\Psi(M[N/x]) \equiv \Psi(M)[\Psi(N)/x]$.

Proof. By induction on the structure of M.

(a) $M \equiv y$ $(y \not\equiv x)$. Then $\Psi(y[N/x]) \equiv \Psi(y) \equiv y[] \stackrel{19}{\equiv} (y[])[\Psi(N)/x] \equiv \Psi(y)[\Psi(N)/x]$.

(b) $M \equiv x$. Then $\Psi(x[N/x]) \equiv \Psi(N) \stackrel{12}{\equiv} \{\Psi(N)\}[] \equiv (x[])[\Psi(N)/x] \equiv \Psi(x)[\Psi(N)/x]$.

(c) $M \equiv M_0 M_1$. Then

$$\begin{aligned}
\Psi((M_0 M_1)[N/x]) &\equiv \Psi(M_0[N/x] M_1[N/x]) \\
&\equiv \{\Psi(M_0[N/x])\}(\Psi(M_1[N/x]) :: []) \\
&\equiv \{\Psi(M_0)[\Psi(N)/x]\}(\Psi(M_1)[\Psi(N)/x] :: []) \quad \text{(by IH)} \\
&\equiv (\{\Psi(M_0)\}(\Psi(M_1) :: []))[\Psi(N)/x] \quad \text{(by Lemma 18)} \\
&\equiv \Psi(M_0 M_1)[\Psi(N)/x]
\end{aligned}$$

(d) $M \equiv \lambda y.M_0$. Then $\Psi((\lambda y.M_0)[N/x]) \equiv \Psi(\lambda y.M_0[N/x]) \equiv \lambda y.\Psi(M_0[N/x]) \stackrel{IH}{\equiv} \lambda y.\Psi(M_0)[\Psi(N)/x] \equiv (\lambda y.\Psi(M_0))[\Psi(N)/x] \equiv \Psi(\lambda y.M_0)[\Psi(N)/x]$. □

Lemma 27. $\Theta(t[u/x]) \equiv \Theta(t)[\Theta(u)/x]$.

Proof.

$$\begin{aligned}
\Theta(t[u/x]) &\equiv \Theta(\Psi(\Theta(t))[\Psi(\Theta(u))/x]) \quad \text{(by Proposition 7)} \\
&\equiv \Theta(\Psi(\Theta(t)[\Theta(u)/x])) \quad \text{(by Lemma 26)} \\
&\equiv \Theta(t)[\Theta(u)/x] \quad \text{(by Proposition 7)}
\end{aligned}$$

□

Lemma 28. *For any λ-terms M, M', if $M \to_\beta M'$ then $\Theta'(M,l) \to_\beta \Theta'(M',l)$.*

Proof. By induction on the structure of l. □

Theorem 4.

1. *For any λ-terms M, M', if $M \to_\beta M'$ then $\Psi(M) \to_\beta \Psi(M')$.*
2. *For any pure terms $t, t' \in \mathcal{T}_{\overline{\lambda}\mathbf{x}}$, if $t \to_\beta t'$ then $\Theta(t) \to_\beta \Theta(t')$.*

Proof. 1. By induction on the structure of M.

(a) $M \equiv (\lambda x.M_0)M_1 \to_\beta M_0[M_1/x] \equiv M'$. Then

$$\begin{array}{rcll} \Psi((\lambda x.M_0)M_1) & \equiv & \{\Psi(\lambda x.M_0)\}(\Psi(M_1) :: []) & \\ & \equiv & \{\lambda x.\Psi(M_0)\}(\Psi(M_1) :: []) & \\ & \equiv & (\lambda x.\Psi(M_0))(\Psi(M_1) :: []) & \\ & \to_\beta & \{\Psi(M_0)[\Psi(M_1)/x]\}[] & \\ & \equiv & \Psi(M_0)[\Psi(M_1)/x] & \text{(by Lemma 12 (2))} \\ & \equiv & \Psi(M_0[M_1/x]) & \text{(by Lemma 26)} \end{array}$$

(b) $M \equiv M_0M_1$ and $M_0 \to_\beta M_0'$. By the induction hypothesis, $\Psi(M_0) \to_\beta \Psi(M_0')$. Hence

$$\begin{array}{rcll} \Psi(M_0M_1) & \equiv & \{\Psi(M_0)\}(\Psi(M_1) :: []) & \\ & \to_\beta & \{\Psi(M_0')\}(\Psi(M_1) :: []) & \text{(by Lemma 16 (1))} \\ & \equiv & \Psi(M_0'M_1) & \end{array}$$

(c) $M \equiv M_0M_1$ and $M_1 \to_\beta M_1'$. Similar, using Lemma 16 (2).

(d) $M \equiv \lambda x.M_0$ and $M_0 \to_\beta M_0'$. By the induction hypothesis, $\Psi(M_0) \to_\beta \Psi(M_0')$. Hence $\Psi(\lambda x.M_0) \equiv \lambda x.\Psi(M_0) \to_\beta \lambda x.\Psi(M_0') \equiv \Psi(\lambda x.M_0')$.

2. We prove the following by simultaneous induction on the structure of t or l:

2.1. if $t \to_\beta t'$ then $\Theta(t) \to_\beta \Theta(t')$,

2.2. if $l \to_\beta l'$ then $\Theta'(M,l) \to_\beta \Theta'(M,l')$.

2.1. (a) $t \equiv xl_0$ and $l_0 \to_\beta l_0'$. By the induction hypothesis, $\Theta'(x,l_0) \to_\beta \Theta'(x,l_0')$. Hence $\Theta(xl_0) \equiv \Theta'(x,l_0) \to_\beta \Theta'(x,l_0') \equiv \Theta(xl_0')$.

(b) $t \equiv \lambda x.t_0$ and $t_0 \to_\beta t_0'$. By the induction hypothesis, $\Theta(t_0) \to_\beta \Theta(t_0')$. Hence $\Theta(\lambda x.t_0) \equiv \lambda x.\Theta(t_0) \to_\beta \lambda x.\Theta(t_0') \equiv \Theta(\lambda x.t_0')$.

(c) $t \equiv (\lambda x.t_0)(u :: l_0)$.

i. The β-reduction is at the root, i.e., $(\lambda x.t_0)(u :: l_0) \to_\beta \{t_0[u/x]\}l_0$. Then

$$\begin{array}{rcll} \Theta((\lambda x.t_0)(u :: l_0)) & \equiv & \Theta'(\lambda x.\Theta(t_0), u :: l_0) & \\ & \equiv & \Theta'((\lambda x.\Theta(t_0))\Theta(u), l_0) & \\ & \to_\beta & \Theta'(\Theta(t_0)[\Theta(u)/x], l_0) & \text{(by Lemma 28)} \\ & \equiv & \Theta'(\Theta(t_0[u/x]), l_0) & \text{(by Lemma 27)} \\ & \equiv & \Theta(\{t_0[u/x]\}l_0) & \text{(by Lemma 25)} \end{array}$$

ii. The β-reduction is internal, e.g., $t_0 \to_\beta t_0'$. By the induction hypothesis, $\Theta(t_0) \to_\beta \Theta(t_0')$. Hence

$$\begin{aligned} \Theta((\lambda x.t_0)(u :: l_0)) &\equiv \Theta'(\lambda x.\Theta(t_0), u :: l_0) \\ &\to_\beta \Theta'(\lambda x.\Theta(t_0'), u :: l_0) \quad \text{(by Lemma 28)} \\ &\equiv \Theta((\lambda x.t_0')(u :: l_0)) \end{aligned}$$

The other cases are similar, using the induction hypothesis.

2.2. (a) $l \equiv u :: l_0$ and $u \to_\beta u'$. By the induction hypothesis, $\Theta(u) \to_\beta \Theta(u')$. Hence

$$\begin{aligned} \Theta'(M, u :: l_0) &\equiv \Theta'(M\Theta(u), l_0) \\ &\to_\beta \Theta'(M\Theta(u'), l_0) \quad \text{(by Lemma 28)} \\ &\equiv \Theta'(M, u' :: l_0) \end{aligned}$$

(b) $l \equiv u :: l_0$ and $l_0 \to_\beta l_0'$. Similar, using the induction hypothesis. □

D Confluence of the Subcalculus x

In this section we show that each critical pair of the subcalculus `x` is joinable, following Appendix B of [9] with different terminology. We consider all pairs of rules R_1, R_2 where the LHS of R_1 unifies with a non-variable subterm of the LHS of R_2, in which case we say that R_1 *overlaps* with R_2.

1. Rule (1a) overlaps with (5a).

$$\begin{array}{ccc} ([]l)l' & \xrightarrow{(5a)} & [](ll') \\ {\scriptstyle (1a)}\downarrow & & \downarrow{\scriptstyle (1a)} \\ ll' & = & ll' \end{array}$$

2. Rule (1a) overlaps with (5b).

$$\begin{array}{ccc} ([]l)\langle x := v\rangle & \xrightarrow{(5b)} & []\langle x := v\rangle l\langle x := v\rangle \\ {\scriptstyle (1a)}\downarrow & & \downarrow{\scriptstyle (2a)} \\ l\langle x := v\rangle & \xleftarrow[(1a)]{} & []l\langle x := v\rangle \end{array}$$

3. Rule (1b) overlaps with (5a).

$$\begin{array}{ccccc} ((u :: l)l')l'' & \xrightarrow{(5a)} & (u :: l)(l'l'') & \xrightarrow{(1b)} & u :: (l(l'l'')) \\ {\scriptstyle (1b)}\downarrow & & & & \| \\ (u :: (ll'))l'' & \xrightarrow[(1b)]{} & u :: ((ll')l'') & \xrightarrow[(5a)]{} & u :: (l(l'l'')) \end{array}$$

4. Rule (1b) overlaps with (5b).

$$\begin{array}{ccc}
((u :: l)l')\langle x := v\rangle & \xrightarrow{(5b)} & (u :: l)\langle x := v\rangle l'\langle x := v\rangle \\
{\scriptstyle (1b)}\downarrow & & \downarrow{\scriptstyle (2b)} \\
(u :: (ll'))\langle x := v\rangle & & (u\langle x := v\rangle :: l\langle x := v\rangle)l'\langle x := v\rangle \\
{\scriptstyle (2b)}\downarrow & & \downarrow{\scriptstyle (1b)} \\
u\langle x := v\rangle :: (ll')\langle x := v\rangle & \xrightarrow[(5b)]{} & u\langle x := v\rangle :: (l\langle x := v\rangle l'\langle x := v\rangle)
\end{array}$$

5. Rules (2a), (2b) have no overlaps.
6. Rule (3a) overlaps with (5c).

$$\begin{array}{ccccc}
((xl)l')l'' & \xrightarrow{(5c)} & (xl)(l'l'') & \xrightarrow{(3a)} & x(l(l'l'')) \\
{\scriptstyle (3a)}\downarrow & & & & \parallel \\
(x(ll'))l'' & \xrightarrow[(3a)]{} & x((ll')l'') & \xrightarrow[(5a)]{} & x(l(l'l''))
\end{array}$$

7. Rule (3a) overlaps with (5d). There are two cases.
7.1.

$$\begin{array}{ccc}
((xl)l')\langle x := v\rangle & \xrightarrow{(5d)} & (xl)\langle x := v\rangle l'\langle x := v\rangle \\
{\scriptstyle (3a)}\downarrow & & \downarrow{\scriptstyle (4b)} \\
(x(ll'))\langle x := v\rangle & & (vl\langle x := v\rangle)l'\langle x := v\rangle \\
{\scriptstyle (4b)}\downarrow & & \downarrow{\scriptstyle (5c)} \\
v(ll')\langle x := v\rangle & \xrightarrow[(5b)]{} & v(l\langle x := v\rangle l'\langle x := v\rangle)
\end{array}$$

7.2. Let $y \not\equiv x$.

$$\begin{array}{ccc}
((yl)l')\langle x := v\rangle & \xrightarrow{(5d)} & (yl)\langle x := v\rangle l'\langle x := v\rangle \\
{\scriptstyle (3a)}\downarrow & & \downarrow{\scriptstyle (4a)} \\
(y(ll'))\langle x := v\rangle & & (yl\langle x := v\rangle)l'\langle x := v\rangle \\
{\scriptstyle (4a)}\downarrow & & \downarrow{\scriptstyle (3a)} \\
y(ll')\langle x := v\rangle & \xrightarrow[(5b)]{} & y(l\langle x := v\rangle l'\langle x := v\rangle)
\end{array}$$

8. Rule (3b) overlaps with (5c).

$$\begin{array}{ccc}
((\lambda y.t)[])l & \xrightarrow{(5c)} & (\lambda y.t)([]l) \\
{\scriptstyle (3b)}\downarrow & & \downarrow{\scriptstyle (1a)} \\
(\lambda y.t)l & = & (\lambda y.t)l
\end{array}$$

9. Rule (3b) overlaps with (5d).

$$
\begin{array}{ccc}
((\lambda y.t)[])\langle x := v\rangle & \xrightarrow{(5d)} & (\lambda y.t)\langle x := v\rangle[]\langle x := v\rangle \\
{\scriptstyle (3b)}\Big\downarrow & & \Big\downarrow{\scriptstyle (2a)} \\
(\lambda y.t)\langle x := v\rangle & & (\lambda y.t)\langle x := v\rangle[] \\
{\scriptstyle (4c)}\Big\downarrow & & \Big\downarrow{\scriptstyle (4c)} \\
\lambda y.t\langle x := v\rangle & \xleftarrow[(3b)]{} & (\lambda y.t\langle x := v\rangle)[]
\end{array}
$$

10. Rules (4a), (4b) and (4c) have no overlaps.
11. Rule (5a) overlaps with itself.

$$
\begin{array}{ccccc}
((ll')l'')l''' & \xrightarrow{(5a)} & (ll')(l''l''') & \xrightarrow{(5a)} & l(l'(l''l''')) \\
{\scriptstyle (5a)}\Big\downarrow & & & & \Big\| \\
(l(l'l''))l''' & \xrightarrow[(5a)]{} & l((l'l'')l''') & \xrightarrow[(5a)]{} & l(l'(l''l'''))
\end{array}
$$

12. Rule (5a) overlaps with (5b).

$$
\begin{array}{ccc}
((ll')l'')\langle x := v\rangle & \xrightarrow{(5b)} & (ll')\langle x := v\rangle l''\langle x := v\rangle \\
{\scriptstyle (5a)}\Big\downarrow & & \Big\downarrow{\scriptstyle (5b)} \\
(l(l'l''))\langle x := v\rangle & & (l\langle x := v\rangle l'\langle x := v\rangle)l''\langle x := v\rangle \\
{\scriptstyle (5b)}\Big\downarrow & & \Big\downarrow{\scriptstyle (5a)} \\
l\langle x := v\rangle(l'l'')\langle x := v\rangle & \xrightarrow[(5b)]{} & l\langle x := v\rangle(l'\langle x := v\rangle l''\langle x := v\rangle)
\end{array}
$$

13. Rule (5b) has no (further) overlaps.
14. Rule (5c) overlaps with itself.

$$
\begin{array}{ccccc}
((tl)l')l'' & \xrightarrow{(5c)} & (tl)(l'l'') & \xrightarrow{(5c)} & t(l(l'l'')) \\
{\scriptstyle (5c)}\Big\downarrow & & & & \Big\| \\
(t(ll'))l'' & \xrightarrow[(5c)]{} & t((ll')l'') & \xrightarrow[(5a)]{} & t(l(l'l''))
\end{array}
$$

15. Rule (5c) overlaps with (5d).

$$
\begin{array}{ccc}
((tl)l')\langle x := v\rangle & \xrightarrow{(5d)} & (tl)\langle x := v\rangle l'\langle x := v\rangle \\
{\scriptstyle (5c)}\big\downarrow & & \big\downarrow{\scriptstyle (5d)} \\
(t(ll'))\langle x := v\rangle & & (t\langle x := v\rangle l\langle x := v\rangle)l'\langle x := v\rangle \\
{\scriptstyle (5d)}\big\downarrow & & \big\downarrow{\scriptstyle (5c)} \\
t\langle x := v\rangle(ll')\langle x := v\rangle & \xrightarrow[(5b)]{} & t\langle x := v\rangle(l\langle x := v\rangle l'\langle x := v\rangle)
\end{array}
$$

16. Rule (5d) has no (further) overlaps.