Semantic Equivalence between the Linear Call-by-Value Calculus and the Value Substitution Calculus (Technical Note)

February 2025

Pablo Barenbaum[1], Delia Kesner[2] and Mariana Milicich[2†]

[1] Universidad Nacional de Quilmes (CONICET), and Instituto de Ciencias de la Computación, UBA
Argentina
[2] Université Paris Cité, CNRS, IRIF
France

Abstract

In this technical note, we show that the Value Substitution Calculus (VSC) and the Linear Call-by-Value (LCBV) are observationally equivalent. We rely on the non-idempotent intersection type system $\mathcal{V}$ to establish such a equivalence, as it is already known that $\mathcal{V}$-typability characterises VSC-termination. The technical development focuses on the characterisation of LCBV-termination by means of system $\mathcal{V}$.

1 Preliminary Notions

In this section, we recall Accattoli and Paolini's *Value Substitution Calculus* [1] and give an alternative specification of the *Linear Call-by-Value* (LCBV) [2]. Moreover, we recall a measure for LCBV-terms from [2] and show that substitution steps in LCBV-reduction decrease (Lemma 2).

We recall also the type system $\mathcal{V}$ [3], inspired by Ehrhard's system [5].

1.1 Syntax

We start by giving the syntax of both calculi in this work. Given a denumerable set of **variables** $(x, y, z, \ldots)$, the syntax for describing the sets of **terms** $(t, u, \ldots)$, **substitution contexts** $(\mathtt{L}, \mathtt{L}', \ldots)$, and **values** $(v, w, \ldots)$ is specified by the following grammar:

$$t, u ::= x \mid \lambda x.\, t \mid t\, u \mid t[x\backslash u] \qquad \mathtt{L} ::= \Diamond \mid \mathtt{L}[x\backslash t] \qquad v ::= x \mid \lambda x.\, t$$

The set of terms includes **variables**, **abstractions**, **applications**, and **closures** $t[x\backslash u]$, representing an **explicit substitution** (ES) $[x\backslash u]$ on a term t.

Free and **bound occurrences** of variables are defined as usual, where free occurrences of x in t are bound in $t[x\backslash u]$. Terms are considered up to α-renaming of bound variables. Substitution contexts are lists of ESs, and we write $t\mathtt{L}$ for the **replacement** of the hole $\Diamond$ in $\mathtt{L}$ by t, which may capture the free variables of t. For example, if $\mathtt{L} = [x\backslash y\, y][y\backslash z]$ and $t = x$, then $t\mathtt{L} = x[x\backslash y\, y][y\backslash z]$. We write $t\{x := u\}$ for the **(capture-avoiding) substitution** of the free

† *This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945332.*

occurrences of x by u in t. The sets of **free variables** of a **term** ($\mathsf{fv}(t)$) is defined as expected. The set of **reachable variables** of a term t is written $\mathsf{rv}(t)$ and defined as:

$$\begin{array}{lcllcl} \mathsf{rv}(x) & := & \{x\} & \mathsf{rv}(\lambda x.\, t) & := & \emptyset \\ \mathsf{rv}(t\, u) & := & \mathsf{rv}(t) \cup \mathsf{rv}(u) & \mathsf{rv}(t[x\backslash u]) & := & (\mathsf{rv}(t) \setminus \{x\}) \cup \mathsf{rv}(u) \end{array}$$

Evaluation contexts used in both calculi are defined by the following grammar:

$$\mathtt{W} ::= \Diamond \mid \mathtt{W}\, t \mid t\, \mathtt{W} \mid \mathtt{W}[x\backslash t] \mid t[x\backslash \mathtt{W}]$$

And we write $\mathtt{W}\langle t\rangle$ for the term resulting from replacing the hole $\Diamond$ in $\mathtt{W}$ by t.

1.2 The Value Substitution Calculus

Reduction in the Value Substitution Calculus (VSC) is defined by the two following rewriting rules, closed by arbitrary weak evaluation contexts:

$$(\lambda x.\, t)\mathtt{L}\, u \to_{\mathsf{db}} t[x\backslash u]\mathtt{L} \qquad t[x\backslash v\mathtt{L}] \to_{\mathsf{sv}} t\{x := v\}\mathtt{L}$$

The db-rule ("$\mathsf{d}$istant $\mathsf{b}$eta") is a β-like rule that applies an abstraction $\lambda x.\, t$ to an argument u, creating an ES $[x\backslash u]$ that binds x to the argument u. This rule acts at a *distance*, meaning an arbitrary list of ESs ($\mathtt{L}$) may be in between the abstraction and its argument. For example, $(\lambda x.\, \lambda y.\, x)\, t\, u \to_{\mathsf{db}} (\lambda y.\, x)[x\backslash t]\, u \to_{\mathsf{db}} x[y\backslash u][x\backslash t]$.

The sv-rule ("$\mathsf{s}$ubstitution of $\mathsf{v}$alues") fires an ES when x is bound to a value v surrounded by a context $\mathtt{L}$. This rule performs the meta-level substitution of x by v, potentially erasing or producing many copies of v, while it extrudes the substitution context $\mathtt{L}$, which must remain shared. For example, the step $x[y\backslash z] \to_{\mathsf{sv}} x\{y := z\} = x$ results in the erasure of the value z, while $(x\, x)[x\backslash(\lambda y.\, z)[z\backslash t]] \to_{\mathsf{sv}} ((\lambda y.\, z)\, \lambda y.\, z)[z\backslash t]$ produces two copies of the value $\lambda y.\, z$ but keeps a single copy of the ES $[z\backslash t]$. Note that a term like $x[x\backslash y\, y]$ is in VSC-normal form because $y\, y$ is not a value.

1.3 The Linear Call-by-Value Calculus

We start by giving an alternative specification of the LCBV calculus (w.r.t. [2]) using a more conventional notation. Reduction in LCBV is defined by the two following rewriting rules, closed by arbitrary weak evaluation contexts that is:

$$(\lambda x.\, t)\mathtt{L}\, u \to_{\mathsf{db}} t[x\backslash u]\mathtt{L} \qquad \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}] \to_{\mathsf{lsv}} \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}$$

We say a term t is LCBV**-terminating** if there is no infinite LCBV-reduction sequence starting at t.

1.3.1 Term measures

Given a term t and a variable x, the **potential number of occurences of** x **in** t, written $\#_x(t)$, is a natural number defined as 0 if $x \notin \mathsf{fv}(t)$, and otherwise is defined recursively as follows:

$$\begin{array}{rclrcl} \#_x(x) & := & 1 & \#_x(t\, u) & := & \#_x(t) + \#_x(u) \\ \#_x(\lambda y.\, t) & := & 0 & \#_x(t[y\backslash u]) & := & \#_x(t) + \#_x(u) \cdot (1 + \#_y(t)) \end{array}$$

The **measure of a term** t is written $\#(t)$ and defined as:

$$\begin{array}{rclrcl} \#(x) & := & 0 & \#(t\, u) & := & \#(t) + \#(u) \\ \#(\lambda x.\, t) & := & 0 & \#(t[x\backslash u]) & := & \#(t) + \#_x(t) + \#(u) \cdot (1 + \#_x(t)) \end{array}$$

Remark 1. Let $\mathtt{W}$ be a weak evaluation context, x be any variable and v be any value. Then, $\#(\mathtt{W}\langle x\rangle) = \#(\mathtt{W}\langle v\rangle)$.

Let $\varphi : \mathsf{Var} \to \mathbb{N}$. Given a substitution context $\mathtt{L}$, the **potential number of occurrences of x in $\mathtt{L}$ under φ**, written $\#_x^\varphi(\mathtt{L})$, is recursively defined as follows:

$$\#_x^\varphi(\Diamond) := \varphi(x) \qquad \#_x^\varphi(\mathtt{L}'[y\backslash t]) := \#_x^\varphi(\mathtt{L}') + \#_x(t) \cdot (1 + \#_y^\varphi(\mathtt{L}'))$$

We define the **measure of $\mathtt{L}$ under φ**, written $\#^\varphi(\mathtt{L})$, as follows:

$$\#^\varphi(\Diamond) := 0 \qquad \#^\varphi(\mathtt{L}'[x\backslash t]) := \#^\varphi(\mathtt{L}') + \#_x^\varphi(\mathtt{L}') + \#(t) \cdot (1 + \#_x^\varphi(\mathtt{L}'))$$

Lemma 2. *If $t \to_{\mathsf{lsv}} t'$, then $\#(t) > \#(t')$.*

Proof. We proceed by induction on $t \to_{\mathsf{lsv}} t'$. We only show the base case, that is, when the surrounding evaluation context is empty. The inductive cases are straightforward.

Then, $t = \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}] \to_{\mathsf{lsv}} \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L} = t'$, so that

$$\begin{aligned}
\#(\mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}]) &= \#(\mathtt{W}\langle x\rangle) + \#_x(\mathtt{W}\langle x\rangle) + \#(v\mathtt{L}) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) \\
&= \#(\mathtt{W}\langle x\rangle) + \#_x(\mathtt{W}\langle x\rangle) + (\#(v) + \#^{\varphi_v}(\mathtt{L})) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) \quad \text{(By [2, Lemma C.4 (2)])} \\
&= \#(\mathtt{W}\langle v\rangle) + \#_x(\mathtt{W}\langle x\rangle) + \#(v) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) + \#^{\varphi_{\mathtt{W}\langle v\rangle}}(\mathtt{L}) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) \quad \text{(By Remark 1)} \\
&= \#(\mathtt{W}\langle v\rangle) + \#_x(\mathtt{W}\langle x\rangle) + \#(v) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) + \#^{\varphi_{\mathtt{W}\langle v\rangle}}(\mathtt{L}) \cdot (1 + (\#_x(\mathtt{W}\langle v\rangle) + 1)) \\
&> \#(\mathtt{W}\langle v\rangle) + \#_x(\mathtt{W}\langle x\rangle) + \#(v) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) + \#^{\varphi_{\mathtt{W}\langle v\rangle}}(\mathtt{L}) \cdot (1 + \#_x(\mathtt{W}\langle v\rangle)) \\
&\geq \#(\mathtt{W}\langle v\rangle) + \#_x(\mathtt{W}\langle x\rangle) + \#(v) \cdot (1 + \#_x(\mathtt{W}\langle x\rangle)) + \#^{\varphi_{\mathtt{W}\langle v\rangle[x\backslash v]}}(\mathtt{L}) \quad \text{(By [2, Lemma C.5 (2)])} \\
&= \#(\mathtt{W}\langle v\rangle) + (\#_x(\mathtt{W}\langle v\rangle) + 1) + \#(v) \cdot (1 + (\#_x(\mathtt{W}\langle v\rangle) + 1)) + \#^{\varphi_{\mathtt{W}\langle v\rangle[x\backslash v]}}(\mathtt{L}) \\
&> \#(\mathtt{W}\langle v\rangle) + \#_x(\mathtt{W}\langle v\rangle) + \#(v) \cdot (1 + \#_x(\mathtt{W}\langle v\rangle)) + \#^{\varphi_{\mathtt{W}\langle v\rangle[x\backslash v]}}(\mathtt{L}) \\
&= \#(\mathtt{W}\langle v\rangle[x\backslash v]) + \#^{\varphi_{\mathtt{W}\langle v\rangle[x\backslash v]}}(\mathtt{L}) \\
&= \#(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}) \quad \text{(By [2, Lemma C.4 (2)])}
\end{aligned}$$

□

1.4 The Type System $\mathcal{V}$

Let us consider a set of base types denoted by α. The grammar of types is given by:

$$\begin{array}{rrcll}
\textbf{(Types)} & \sigma, \tau & ::= & \alpha \mid \mathcal{M} \mid \mathcal{M} \to \sigma & \\
\textbf{(Multi-Types)} & \mathcal{M}, \mathcal{N} & ::= & [\sigma_i]_{i \in I} & \text{where } I \text{ is a finite set}
\end{array}$$

Typing environments $\Gamma, \Delta, \ldots$ are functions mapping variables to multi-types. The **domain** of an environment Γ is $\mathsf{dom}(\Gamma) := \{x \mid \Gamma(x) \neq [\,]\}$, and $\emptyset$ denotes the empty typing environment, mapping each variable to $[\,]$.

The **union of multi-types**, written $\mathcal{M}_1 + \mathcal{M}_2$, is a multiset of types defined as expected, where $[\,]$ is the neutral element. For typing environments $(\Gamma_i)_{i\in I}$, we write $+_{i\in I}\Gamma_i$ for the environment mapping each variable x to $+_{i\in I}\Gamma_i(x)$, where $\Gamma + \Delta$ and $\Gamma +_{j\in J} \Delta_j$ are particular instances of the general notation. When $\mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta) = \emptyset$ we write $\Gamma; \Delta$ instead of $\Gamma + \Delta$ to emphasise that the domains of Γ and Δ are disjoint. As a consequence, $\Gamma; x : [\,]$ is identical to Γ.

Type assumptions are denoted $x : \mathcal{M}$, meaning that the environment assigns $\mathcal{M}$ to x, and $[\,]$ to any other variable.

Typing judgements in system $\mathcal{V}$ are of the form $\Gamma \vdash t : \sigma$, where Γ is a type environment, t is a term, and σ is a type.

The **typing rules** of system $\mathcal{V}$ are:

$$\frac{}{x : \mathcal{M} \vdash x : \mathcal{M}}\text{VAR} \qquad \frac{(\Gamma_i; x : \mathcal{M}_i \vdash t : \sigma_i)_{i \in I}}{+_{i \in I}\Gamma_i \vdash \lambda x. t : [\mathcal{M}_i \to \sigma_i]_{i \in I}}\text{ABS}$$

$$\frac{\Gamma \vdash t : [\mathcal{M} \to \sigma] \quad \Delta \vdash u : \mathcal{M}}{\Gamma + \Delta \vdash t\, u : \sigma}\text{APP} \qquad \frac{\Gamma; x : \mathcal{M} \vdash t : \sigma \quad \Delta \vdash u : \mathcal{M}}{\Gamma + \Delta \vdash t[x\backslash u] : \sigma}\text{ES}$$

We write $\Phi \triangleright \Gamma \vdash t : \sigma$ to denote there is a **(typing) derivation** Φ in system $\mathcal{V}$ concluding with the typing judgement $\Gamma \vdash t : \sigma$. We say a term t is **$\mathcal{V}$-typable** if there is a derivation for typing t.

Lemma 3 (Relevance). *If $\Phi \triangleright \Gamma \vdash t : \mathcal{T}$, then $\mathsf{rv}(t) \subseteq \mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(t)$.*

Proof. By a straightforward induction on typing derivations. □

Let Φ be a type derivation with conclusion $\Gamma \vdash t : \mathcal{T}$. Then, the **measure** of Φ is noted by $\mathtt{meas}(\Phi)$, and is a pair of natural numbers given by $\langle \mathsf{sz}(\Phi), \#(t)\rangle$, where $\mathsf{sz}(\Phi)$ is recursively defined as follows:

- $\mathsf{sz}\left(\frac{}{x : \mathcal{T} \vdash x : \mathcal{T}}\text{VAR}\right) = |\mathcal{T}|$
- $\mathsf{sz}\left(\frac{(\Phi_i)_{i \in I}}{\Gamma \vdash \lambda x. t : [\alpha_i]_{i \in I}}\text{ABS}\right) = |I| + \sum_{i \in I} \mathsf{sz}(\Phi_i)$
- $\mathsf{sz}\left(\frac{\Phi_1 \quad \Phi_2}{\Gamma \vdash t\, u : \mathcal{T}}\text{APP}\right) = 1 + \mathsf{sz}(\Phi_1) + \mathsf{sz}(\Phi_2)$
- $\mathsf{sz}\left(\frac{\Phi_1 \quad \Phi_2}{\Gamma \vdash t[x\backslash u] : \mathcal{T}}\text{ES}\right) = 1 + \mathsf{sz}(\Phi_1) + \mathsf{sz}(\Phi_2)$ if $x \in \mathsf{rv}(t)$
- $\mathsf{sz}\left(\frac{\Phi_1 \quad \Phi_2}{\Gamma \vdash t[x\backslash u] : \mathcal{T}}\text{ES}\right) = \mathsf{sz}(\Phi_1) + \mathsf{sz}(\Phi_2)$ otherwise

2 Characterisation of lcbv-termination through $\mathcal{V}$-typability

In this section, we show that $\mathcal{V}$-typability characterises LCBV-termination. To do so, we prove that the type system $\mathcal{V}$ is **sound** and **complete** with respect to LCBV-reduction.

2.1 Soundness

We start addressing the soundness of system $\mathcal{V}$ with respect to the LCBV calculus, which states that $\mathcal{V}$-typability implies LCBV-termination. This result rely on a subject reduction property based in turn on a substitution lemma.

Remark 4. If $\Phi \rhd \Gamma \vdash v : [\,]$, then $\mathsf{sz}(\Phi) = 0$.

Remark 5. Let t be a term. Then, $t = \mathtt{W}\langle x\rangle$ if and only if $x \in \mathsf{rv}(t)$.

Lemma 6 (Splitting/Merging)**.** *The following are equivalent:*

1. $\Phi \rhd \Gamma \vdash v : \mathcal{M}_1 + \mathcal{M}_2$
2. *There exist typing derivations* Φ_1, Φ_2 *and typing contexts* Γ_1, Γ_2 *such that* $\Phi_1 \rhd \Gamma_1 \vdash v : \mathcal{M}_1$*, and* $\Phi_2 \rhd \Gamma_2 \vdash v : \mathcal{M}_2$*, and* $\Gamma = \Gamma_1 + \Gamma_2$*.*

Moreover, $\mathsf{sz}(\Phi) = \mathsf{sz}(\Phi_1) + \mathsf{sz}(\Phi_2)$

Proof. See [4, Lemmas 4.17 and 4.20]. □

Lemma 7 (Partial Substitution)**.** *Let x be a variable, v be a value such that $x \notin \mathsf{fv}(v)$, and let $\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma; x : \mathcal{M} \vdash \mathtt{W}\langle x\rangle : \sigma$. Then, there exists a splitting $\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$ such that, for all $\Phi_v \rhd \Delta \vdash v : \mathcal{M}_1$ there exists a typing derivation $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma + \Delta; x : \mathcal{M}_2 \vdash \mathtt{W}\langle v\rangle : \sigma$. Moreover,* $\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}(\Phi_v) - |\mathcal{M}_1|$.

Proof. We proceed by induction on $\mathtt{W}$, showing only the cases $\mathtt{W} = \diamond$ and $\mathtt{W} = \mathtt{W}_1\, t$, as the remaining cases can be shown reasoning in a similar way.

- $\mathtt{W} = \diamond$. Then, $\mathtt{W}\langle x\rangle = x$ and $\mathtt{W}\langle v\rangle = v$. The typing derivation is of the form $\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma; x : \mathcal{M} \vdash x : \sigma$, so necessarily $\Gamma = \emptyset$ and $\sigma = \mathcal{M}$, as $\Phi_{\mathtt{W}\langle x\rangle}$ can only be obtained from the axiom VAR. We take $\mathcal{M}_1 := \mathcal{M}$ and $\mathcal{M}_2 := [\,]$, yielding the splitting $\mathcal{M} = \mathcal{M} + [\,]$, and $\Delta = \Delta; x : [\,]$ by definition. So we are done, since for any $\Phi_v \rhd \Delta \vdash v : \mathcal{M}$ we yield $\Phi_{\mathtt{W}\langle v\rangle} := \Phi_v$.

  Moreover, $\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = \mathsf{sz}(\Phi_v) = |\mathcal{M}| + \mathsf{sz}(\Phi_v) - |\mathcal{M}| = \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}(\Phi_v) - |\mathcal{M}|$.

- $\mathtt{W} = \mathtt{W}_1\, t$. Then, $\Phi_{\mathtt{W}\langle x\rangle}$ is of the form:

  $$\dfrac{\Phi_{\mathtt{W}_1\langle x\rangle} \rhd \Gamma_1; x : \mathcal{M}'_1 \vdash \mathtt{W}_1\langle x\rangle : [\mathcal{N} \to \sigma] \qquad \Phi_t \rhd \Gamma_2; x : \mathcal{M}'_2 \vdash t : \mathcal{N}}{\Gamma_1 + \Gamma_2; x : (\mathcal{M}'_1 + \mathcal{M}'_2) \vdash \mathtt{W}_1\langle x\rangle\, t : \sigma}\,\text{APP}$$

  where $\Gamma = \Gamma_1 + \Gamma_2$ and $\mathcal{M} = \mathcal{M}'_1 + \mathcal{M}'_2$. By the *i.h.* on $\Phi_{\mathtt{W}_1\langle x\rangle}$, there exists a splitting $\mathcal{M}'_1 = \mathcal{M}_{11} + \mathcal{M}_{12}$ such that for all $\Phi_v \rhd \Delta \vdash v : \mathcal{M}_{11}$ there exists a typing derivation $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma_1 + \Delta; x : \mathcal{M}_{12} \vdash \mathtt{W}\langle v\rangle : \sigma$ such that $\mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) = \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathsf{sz}(\Phi_v) - |\mathcal{M}_{11}|$.

  Applying the rule APP with $\Phi_{\mathtt{W}_1\langle v\rangle}$ and Φ_t as premises, we build the derivation $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma_1 + \Delta + \Gamma_2; x : \mathcal{M}_{12} + \mathcal{M}'_2 \vdash \mathtt{W}_1\langle v\rangle\, t : \mathcal{T}$, where we let $\mathcal{M}_1 := \mathcal{M}_{11}$ and $\mathcal{M}_2 := \mathcal{M}_{12} + \mathcal{M}'_2$. Note that $\Gamma_1 + \Delta + \Gamma_2 = \Gamma + \Delta$.

  Moreover, $\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = 1 + \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) + \mathsf{sz}(\Phi_t) =_{i.h.} 1 + \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathsf{sz}(\Phi_v) - |\mathcal{M}_{11}| + \mathsf{sz}(\Phi_t) = \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}(\Phi_v) - |\mathcal{M}_1|$.

- $\mathtt{W} = \mathtt{W}_1[y\backslash t]$. Then, we can assume $y \neq x$ by α-conversion, and $\Phi_{\mathtt{W}\langle x\rangle}$ is of the form

  $$\dfrac{\Phi_{\mathtt{W}_1\langle x\rangle} \rhd \Gamma_1; x : \mathcal{M}'_1; y : \mathcal{L} \vdash \mathtt{W}_1\langle x\rangle : \sigma \qquad \Phi_t \rhd \Gamma_2; x : \mathcal{M}'_2 \vdash t : \mathcal{L}}{\Gamma_1 + \Gamma_2; x : (\mathcal{M}'_1 + \mathcal{M}'_2) \vdash \mathtt{W}_1\langle x\rangle[y\backslash t] : \sigma}\,\text{ES}$$

  where $\Gamma = \Gamma_1 + \Gamma_2$ and $\mathcal{M} = \mathcal{M}'_1 + \mathcal{M}'_2$.

  Since $x \notin \mathsf{fv}(v)$, we can then apply the *i.h.* on $\mathtt{W}_1$, yielding the splitting $\mathcal{M}'_1 = \mathcal{M}_{11} + \mathcal{M}_{12}$ such that for all $\Phi_v \rhd \Delta \vdash v : \mathcal{M}_{11}$ there exists a typing derivation

$\Phi_{\mathtt{W}_1\langle v\rangle} \rhd ((\Gamma_1; y : \mathcal{L}) + \Delta); x : \mathcal{M}_{12} \vdash \mathtt{W}_1\langle v\rangle : \sigma$. Moreover, $\mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) = \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v\right) - |\mathcal{M}_{11}|$.

Note that $y \notin \mathsf{fv}(v)$ by α-conversion, thus $y \notin \mathsf{dom}(\Delta)$ by Lemma 3, so we can write $((\Gamma_1; y : \mathcal{L}) + \Delta)$ as $(\Gamma_1 + \Delta); y : \mathcal{L}$.

Applying the rule ES with $\Phi_{\mathtt{W}_1\langle v\rangle}$ and Φ_t as premises, we build the derivation $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma_1 + \Gamma_2 + \Delta; x : \mathcal{M}_{12} + \mathcal{M}'_2 \vdash \mathtt{W}_1\langle v\rangle[y\backslash t] : \sigma$, where we let $\mathcal{M}_1 := \mathcal{M}_{11}$ and $\mathcal{M}_2 := \mathcal{M}_{12} + \mathcal{M}'_2$. Note that $\Gamma_1 + \Delta + \Gamma_2 = \Gamma + \Delta$.

Moreover, if $y \in \mathsf{rv}(\mathtt{W}\langle x\rangle)$, then in particular $y \in \mathsf{rv}(\mathtt{W}_1\langle x\rangle)$, since we can assume $y \notin \mathsf{fv}(t)$ by α-conversion. Then, we yield $y \in \mathsf{rv}(\mathtt{W}_1\langle v\rangle)$, and we obtain the size of $\Phi_{\mathtt{W}\langle v\rangle}$ as follows:

$$\begin{aligned}
\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) &=_{y\in\mathsf{rv}(\mathtt{W}_1\langle v\rangle)} 1 + \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) + \mathsf{sz}\left(\Phi_t\right) \\
&=_{i.h.} 1 + \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v\right) - |\mathcal{M}_{11}| + \mathsf{sz}\left(\Phi_t\right) \\
&=_{y\in\mathsf{rv}(\mathtt{W}_1\langle x\rangle)} \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v\right) - |\mathcal{M}_{11}|
\end{aligned}$$

Otherwise, $y \notin \mathsf{rv}(\mathtt{W}\langle x\rangle)$, so in particular $y \notin \mathsf{rv}(\mathtt{W}_1\langle x\rangle)$. Then, we yield $y \notin \mathsf{rv}(\mathtt{W}_1\langle v\rangle)$, and we obtain the size of $\Phi_{\mathtt{W}\langle v\rangle}$ as follows:

$$\begin{aligned}
\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) &=_{y\notin\mathsf{rv}(\mathtt{W}_1\langle v\rangle)} \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) + \mathsf{sz}\left(\Phi_t\right) \\
&=_{i.h.} \mathsf{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v\right) - |\mathcal{M}_{11}| + \mathsf{sz}\left(\Phi_t\right) \\
&=_{y\notin\mathsf{rv}(\mathtt{W}_1\langle x\rangle)} \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v\right) - |\mathcal{M}_{11}|
\end{aligned}$$

□

Lemma 8 (Weighted Subject Reduction)**.** *Let $\Phi \rhd \Gamma \vdash t : \sigma$ and $t \to_{\text{LCBV}} t'$. Then, there exists a derivation Φ' such that $\Phi' \rhd \Gamma \vdash t' : \sigma$ and moreover $\mathsf{sz}(\Phi) \geq \mathsf{sz}(\Phi')$ and $\mathtt{meas}(\Phi) >_{\mathtt{lex}} \mathtt{meas}(\Phi')$*[1].

Proof. By induction on $t \to_{\text{LCBV}} t'$. We only show the base case $t \to_{\mathsf{lsv}} t'$, as the case $t \to_{\mathsf{db}} t'$ is analogous to [4, Lemma 3.4]. Moreover, the inductive cases are straightforward.

Let $t = \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}]$ and $t' = \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}$. We proceed by induction on $\mathtt{L}$.

- $\mathtt{L} = \diamond$. Then, Φ has the form:

$$\frac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_1; x : \mathcal{M} \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \Phi_v \rhd \Gamma_2 \vdash v : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash \mathtt{W}\langle x\rangle[x\backslash v] : \sigma}\,\text{ES}$$

  where $\Gamma = \Gamma_1 + \Gamma_2$.

  Let $\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$ be the splitting obtained by Lemma 7 on $\Phi_{\mathtt{W}\langle x\rangle}$. Then, by Lemma 6 on Φ_v, there exist typing derivations $\Phi_v^1 \rhd \Gamma_{21} \vdash v : \mathcal{M}_1$ and $\Phi_v^2 \rhd \Gamma_{22} \vdash v : \mathcal{M}_2$ such that $\mathsf{sz}\left(\Phi_v\right) = \mathsf{sz}\left(\Phi_v^1\right) + \mathsf{sz}\left(\Phi_v^1\right)$.

  Applying Lemma 7 on $\Phi_{\mathtt{W}\langle x\rangle}$ and Φ_v^1, we yield a derivation $\Phi_{\mathtt{W}\langle v\rangle} \rhd (\Gamma_1 + \Gamma_{21}); x : \mathcal{M}_2 \vdash \mathtt{W}\langle v\rangle : \sigma$ and $\mathsf{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = \mathsf{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathsf{sz}\left(\Phi_v^1\right) - |\mathcal{M}_1|$.

  Thus, we conclude by building the following derivation Φ':

$$\frac{\Phi_{\mathtt{W}\langle v\rangle} \rhd (\Gamma_1 + \Gamma_{21}); x : \mathcal{M}_2 \vdash \mathtt{W}\langle v\rangle : \sigma \qquad \Phi_v^2 \rhd \Gamma_{22} \vdash v : \mathcal{M}_2}{\Gamma_1 + \Gamma_2 \vdash \mathtt{W}\langle v\rangle[x\backslash v] : \sigma}\,\text{ES}$$

[1] Where $\mathtt{lex}$ stands for the lexicographical order, as the size of a typing derivation is a pair of natural numbers.

To calculate $\mathtt{sz}(\Phi)$ and $\mathtt{sz}(\Phi')$, we analyse two cases, and considering $x \in \mathsf{rv}(\mathtt{W}\langle x\rangle)$ by Remark 5.

– If $x \in \mathsf{rv}(\mathtt{W}\langle v\rangle)$, then

$$\begin{aligned}\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}(\Phi_v)\\ &= 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v^1\right) + \mathtt{sz}\left(\Phi_v^2\right)\\ &\geq 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v^1\right) - |\mathcal{M}_1| + \mathtt{sz}\left(\Phi_v^2\right)\\ &= 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) + \mathtt{sz}\left(\Phi_v^2\right)\\ &=_{x\in\mathsf{rv}(\mathtt{W}\langle v\rangle)} \mathtt{sz}(\Phi')\end{aligned}$$

– If $x \notin \mathsf{rv}(\mathtt{W}\langle v\rangle)$, then

$$\begin{aligned}\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}(\Phi_v)\\ &= 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v^1\right) + \mathtt{sz}\left(\Phi_v^2\right)\\ &> \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v^1\right) - |\mathcal{M}_1| + \mathtt{sz}\left(\Phi_v^2\right)\\ &= \mathtt{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) + \mathtt{sz}\left(\Phi_v^2\right)\\ &=_{x\notin\mathsf{rv}(\mathtt{W}\langle v\rangle)} \mathtt{sz}(\Phi')\end{aligned}$$

We conclude since $\mathtt{sz}(\Phi) \geq \mathtt{sz}(\Phi')$ and $\#(t) > \#(t')$ (Lemma 2) imply $\mathtt{meas}(\Phi) >_{\mathtt{lex}} \mathtt{meas}(\Phi')$.

- $\mathtt{L} = \mathtt{L}'[y\backslash s]$. Then, Φ has the form:

$$\dfrac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma'; x : \mathcal{M} \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \dfrac{\Phi_{v\mathtt{L}'} \rhd \Delta_1; y : \mathcal{N} \vdash v\mathtt{L}' : \mathcal{M} \qquad \Phi_s \rhd \Delta_2 \vdash s : \mathcal{N}}{\Delta_1 + \Delta_2 \vdash v\mathtt{L}'[y\backslash s] : \mathcal{M}}\,\mathrm{ES}}{\Gamma' + \Delta_1 + \Delta_2 \vdash \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'[y\backslash s]] : \sigma}\,\mathrm{ES}$$

Where $\Gamma = \Gamma' + \Delta_1 + \Delta_2$. Moreover, we can assume $y \notin \mathsf{fv}(\mathtt{W}\langle x\rangle)$ by α-conversion.

We now build the following derivation Ψ:

$$\dfrac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma'; x : \mathcal{M} \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \Phi_{v\mathtt{L}'} \rhd \Delta_1; y : \mathcal{N} \vdash v\mathtt{L}' : \mathcal{M}}{\Gamma' + (\Delta_1; y : \mathcal{N}) \vdash \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'] : \sigma}\,\mathrm{ES}$$

Moreover, $\mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'] \to_{\mathsf{lsv}} \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}'$. Then, we can apply the *i.h.*, yielding Ψ' such that $\Psi' \rhd \Gamma' + (\Delta_1; y : \mathcal{N}) \vdash \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}' : \sigma$ and moreover $\mathtt{sz}(\Psi) \geq \mathtt{sz}(\Psi')$ and $\mathtt{meas}(\Psi) >_{\mathtt{lex}} \mathtt{meas}(\Psi')$.

Then, we build Φ' as follows:

$$\dfrac{\Psi' \rhd \Gamma' + (\Delta_1; y : \mathcal{N}) \vdash \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}' : \sigma \qquad \Phi_s \rhd \Delta_2 \vdash s : \mathcal{N}}{\Gamma' + \Delta_1 + \Delta_2 \vdash \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}'[y\backslash s] : \sigma}\,\mathrm{ES}$$

Note that we can assume $y \notin \mathsf{dom}(\Gamma')$ by α-conversion and Lemma 3.

To calculate $\mathtt{sz}(\Phi)$ and $\mathtt{sz}(\Phi')$, we analyse two cases, and considering $x \in \mathsf{rv}(\mathtt{W}\langle x\rangle)$ by Remark 5.

– If $y \in \mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')$, recall that $y \notin \mathsf{fv}(\mathtt{W}\langle x\rangle)$ by α-conversion, so $y \notin \mathsf{rv}(\mathtt{W}\langle x\rangle)$. Then

$$\begin{aligned}
\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle),y\in\mathsf{rv}(v\mathtt{L}')} 1+\mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right)+1+\mathtt{sz}(\Phi_{v\mathtt{L}'})+\mathtt{sz}(\Phi_s)\\
&= 1+\mathtt{sz}(\Psi)+\mathtt{sz}(\Phi_s)\\
&\geq_{i.h.} 1+\mathtt{sz}(\Psi')+\mathtt{sz}(\Phi_s)\\
&=_{y\in\mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')} \mathtt{sz}(\Phi')
\end{aligned}$$

– If $y \notin \mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')$, then in particular $y \notin \mathsf{rv}(v\mathtt{L}')$. Thus

$$\begin{aligned}
\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle),y\notin\mathsf{rv}(v\mathtt{L}')} 1+\mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right)+\mathtt{sz}(\Phi_{v\mathtt{L}'})+\mathtt{sz}(\Phi_s)\\
&= \mathtt{sz}(\Psi)+\mathtt{sz}(\Phi_s)\\
&\geq_{i.h.} \mathtt{sz}(\Psi')+\mathtt{sz}(\Phi_s)\\
&=_{y\notin\mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')} \mathtt{sz}(\Phi')
\end{aligned}$$

We conclude since $\mathtt{sz}(\Phi) \geq \mathtt{sz}(\Phi')$ and $\#(t) > \#(t')$ (Lemma 2) imply $\mathtt{meas}(\Phi) >_{\mathtt{lex}} \mathtt{meas}(\Phi')$.

□

Proposition 9 (Soundness of System $\mathcal{U}$)**.** *Let t be a $\mathcal{V}$-typable term. Then, t is* LCBV*-terminating.*

Proof. Straightforward by Lemma 8. □

2.2 Completeness

We now prove the completeness of system $\mathcal{V}$ with respect to the LCBV calculus, which states that LCBV-termination implies $\mathcal{V}$-typability. The proof for this result follows well-understood techniques, requiring a subject expansion property based in turn on an anti-partial substitution lemma.

Lemma 10 (Anti-Partial Substitution)**.** *Let $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma; x : \mathcal{M} \vdash \mathtt{W}\langle v\rangle : \sigma$, where $x \notin \mathsf{fv}(v)$. Then, there exist typing derivations $\Phi_{\mathtt{W}\langle x\rangle}$ and Φ_v, typing environments $\Gamma_{\mathtt{W}\langle x\rangle}$ and Δ, and a multi-type $\mathcal{N}$ satisfying:*

1. $\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{\mathtt{W}\langle x\rangle}; x : \mathcal{M}+\mathcal{N} \vdash \mathtt{W}\langle x\rangle : \sigma$
2. $\Phi_v \rhd \Delta \vdash v : \mathcal{N}$
3. $\Gamma = \Gamma_{\mathtt{W}\langle x\rangle} + \Delta$

Moreover, $\mathtt{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}(\Phi_v) - |\mathcal{N}|$.

Proof. By induction on $\mathtt{W}$. We detail the cases when $\mathtt{W} = \diamond$, $\mathtt{W} = \mathtt{W}_1\, t$, and $\mathtt{W} = \mathtt{W}_1[x\backslash t]$, as all the remaining inductive cases are similar, and follow easily from the *i.h.*.

- $\mathtt{W} = \diamond$. Then, $\mathtt{W}\langle v\rangle = v$ and $\mathtt{W}\langle x\rangle = x$. The typing derivation is of the form $\Phi_{\mathtt{W}\langle v\rangle} \rhd \Gamma; x : \mathcal{M} \vdash v : \sigma$, so it is the case that σ is a multi-type, *i.e.*, $\sigma = \mathcal{N}$. Since $x \notin \mathsf{fv}(v)$, then necessarily $\mathcal{M} = [\,]$ by Lemma 3. We take $\Gamma_{\mathtt{W}\langle x\rangle} := \emptyset$, $\Delta := \Gamma$, and $\mathcal{N}$ so that $\Phi_v := \Phi_{\mathtt{W}\langle v\rangle}$ and moreover $\Phi_{\mathtt{W}\langle x\rangle} \rhd x : \mathcal{N} \vdash x : \mathcal{N}$ by rule VAR.

  Furthermore, $\mathtt{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = \mathtt{sz}(\Phi_v) = |\mathcal{N}| + \mathtt{sz}(\Phi_v) - |\mathcal{N}| = \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}(\Phi_v) - |\mathcal{N}|$.

- $\mathtt{W} = \mathtt{W}_1\, t$. Then, the typing derivation is of the form

$$\frac{\Phi_{\mathtt{W}_1\langle v\rangle} \rhd \Gamma_1; x : \mathcal{M}_1 \vdash \mathtt{W}_1\langle v\rangle : [\mathcal{L} \to \sigma] \qquad \Phi_t \rhd \Gamma_2; x : \mathcal{M}_2 \vdash t : \mathcal{L}}{\Gamma_1 + \Gamma_2; x : (\mathcal{M}_1 + \mathcal{M}_2) \vdash \mathtt{W}_1\langle v\rangle\, t : \sigma}\text{APP}$$

  where $\Gamma = \Gamma_1 + \Gamma_2$ and $\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$.

  Since $x \notin \mathsf{fv}(v)$, we can then apply the *i.h.* on $\mathtt{W}_1$, yielding typing derivations $\Phi_{\mathtt{W}_1\langle x\rangle}$ and Φ_v, typing environments $\Gamma_{\mathtt{W}_1\langle x\rangle}$ and Δ, and a multi-type $\mathcal{N}$ satisfying:

  1.1 $\Phi_{\mathtt{W}_1\langle x\rangle} \rhd \Gamma_{\mathtt{W}_1\langle x\rangle}; x : (\mathcal{M}_1 + \mathcal{N}) \vdash \mathtt{W}_1\langle x\rangle : [\mathcal{L} \to \sigma]$

  1.2 $\Phi_v \rhd \Delta \vdash v : \mathcal{N}$

  1.3 $\Gamma_1 = \Gamma_{\mathtt{W}_1\langle x\rangle} + \Delta$

  And moreover, $\mathtt{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) = \mathtt{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) - |\mathcal{N}|$. Then, taking the typing environments $\Gamma_{\mathtt{W}\langle x\rangle} := \Gamma_{\mathtt{W}_1\langle x\rangle} + \Gamma_2$ and Δ, and the type $\mathcal{N}$, we yield the following:

  1. $\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{\mathtt{W}\langle x\rangle}; x : (\mathcal{M} + \mathcal{N}) \vdash \mathtt{W}_1\langle x\rangle\, t : \sigma$, by applying rule APP to $\Phi_{\mathtt{W}_1\langle x\rangle}$ and the conclusion of Φ_t.
  2. $\Phi_v \rhd \Delta \vdash v : \mathcal{N}$
  3. $\Gamma = \Gamma_1 + \Gamma_2 =_{i.h.} \Gamma_{\mathtt{W}_1\langle x\rangle} + \Delta + \Gamma_2 = \Gamma_{\mathtt{W}\langle x\rangle} + \Delta$

  And moreover, $\mathtt{sz}\left(\Phi_{\mathtt{W}\langle v\rangle}\right) = 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}_1\langle v\rangle}\right) + \mathtt{sz}\left(\Phi_t\right) =_{i.h.} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}_1\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_t\right) + \mathtt{sz}\left(\Phi_v\right) - |\mathcal{N}| = \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) - |\mathcal{N}|$.

□

Lemma 11 (Weighted Subject Expansion). *Let $t \to_{\text{LCBV}} t'$ and $\Phi' \rhd \Gamma \vdash t' : \sigma$. Then, there exists a typing derivation Φ such that $\Phi \rhd \Gamma \vdash t : \sigma$ and moreover $\mathtt{sz}(\Phi) \geq \mathtt{sz}(\Phi')$ and $\mathtt{meas}(\Phi) >_{\mathtt{lex}} \mathtt{meas}(\Phi')$.*

Proof. By induction on $t \to_{\text{LCBV}} t'$.

We only show the base case $t \to_{\mathsf{lsv}} t'$, as the reasoning for the base case $t \to_{\mathsf{db}} t'$ is analogous to the one in Lemma 8, and inductive cases are straightforward.

- Let $t = \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}] \to_{\mathsf{lsv}} \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L} = t'$. We proceed by induction on $\mathtt{L}$.

  - $\mathtt{L} = \diamond$. Then, Φ' has the form:

$$\frac{\Phi'_{\mathtt{W}\langle v\rangle} \rhd \Gamma_1; x : \mathcal{M} \vdash \mathtt{W}\langle v\rangle : \sigma \qquad \Phi'_v \rhd \Gamma_2 \vdash v : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash \mathtt{W}\langle v\rangle[x\backslash v] : \mathcal{T}}\text{ES}$$

    where $\Gamma = \Gamma_1 + \Gamma_2$.

    We can assume by α-conversion that $x \notin \mathsf{fv}(v)$. By Lemma 10 on $\Phi'_{\mathtt{W}\langle v\rangle}$, we yield typing derivations $\Phi_{\mathtt{W}\langle x\rangle}$ and Φ_v, typing environments $\Gamma_{\mathtt{W}\langle x\rangle}$ and Δ, and a multi-type $\mathcal{N}$ satisfying

    1. $\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{\mathtt{W}\langle x\rangle}; x : (\mathcal{M} + \mathcal{N}) \vdash \mathtt{W}\langle x\rangle : \sigma$
    2. $\Phi_v \rhd \Delta \vdash v : \mathcal{N}$
    3. $\Gamma_1 = \Gamma_{\mathtt{W}\langle x\rangle} + \Delta$

Moreover, $\mathtt{sz}\left(\Phi'_{\mathtt{W}\langle v\rangle}\right) = \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) - |\mathcal{N}|$. We can apply Lemma 6 to Φ'_v and Φ_v, yielding $\Phi''_v \rhd \Gamma_2 + \Delta \vdash v : \mathcal{M} + \mathcal{N}$. Moreover, $\mathtt{sz}\left(\Phi''_v\right) = \mathtt{sz}\left(\Phi'_v\right) + \mathtt{sz}\left(\Phi_v\right)$. Then, we build Φ as follows:

$$\dfrac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{\mathtt{W}\langle x\rangle}; x : (\mathcal{M} + \mathcal{N}) \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \Phi''_v \rhd \Gamma_2 + \Delta \vdash v : \mathcal{M} + \mathcal{N}}{\Gamma_{\mathtt{W}\langle x\rangle} + \Gamma_2 + \Delta \vdash \mathtt{W}\langle x\rangle[x\backslash v] : \sigma}\,\mathrm{ES}$$

By Remark 5, $x \in \mathsf{rv}(\mathtt{W}\langle x\rangle)$, and consider two cases, depending on whether $x \in \mathsf{rv}(\mathtt{W})$.

* If $x \in \mathsf{rv}(\mathtt{W})$, then

$$\begin{aligned}
\mathtt{sz}\left(\Phi\right) &=_{x \in \mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi''_v\right) \\
&= 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) + \mathtt{sz}\left(\Phi'_v\right) \\
&\geq 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) + \mathtt{sz}\left(\Phi'_v\right) - |\mathcal{N}| \\
&= 1 + \mathtt{sz}\left(\Phi'_{\mathtt{W}\langle v\rangle}\right) + \mathtt{sz}\left(\Phi'_v\right) \\
&=_{x \in \mathsf{rv}(\mathtt{W})} \mathtt{sz}\left(\Phi'\right)
\end{aligned}$$

* Otherwise, $x \notin \mathsf{rv}(\mathtt{W})$, then

$$\begin{aligned}
\mathtt{sz}\left(\Phi\right) &=_{x \in \mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi''_v\right) \\
&= 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) + \mathtt{sz}\left(\Phi'_v\right) \\
&\geq 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}\left(\Phi_v\right) + \mathtt{sz}\left(\Phi'_v\right) - |\mathcal{N}| \\
&> \mathtt{sz}\left(\Phi'_{\mathtt{W}\langle v\rangle}\right) + \mathtt{sz}\left(\Phi'_v\right) \\
&=_{x \notin \mathsf{rv}(\mathtt{W})} \mathtt{sz}\left(\Phi'\right)
\end{aligned}$$

We conclude since $\mathtt{sz}\left(\Phi\right) \geq \mathtt{sz}\left(\Phi'\right)$ in the first case, and $\mathtt{sz}\left(\Phi\right) > \mathtt{sz}\left(\Phi'\right)$ in the second case. Moreover, $\#(t) > \#(t')$ by Lemma 2, so $\mathtt{meas}\left(\Phi\right) >_{\mathtt{lex}} \mathtt{meas}\left(\Phi'\right)$.

– $\mathtt{L} = \mathtt{L}'[y\backslash s]$. Then, Φ' has the form:

$$\dfrac{\Phi'_1 \rhd \Gamma_1; y : \mathcal{L} \vdash \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}' : \sigma \qquad \Phi'_s \rhd \Gamma_2 \vdash s : \mathcal{L}}{\Gamma_1 + \Gamma_2 \vdash \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}'[y\backslash s] : \sigma}\,\mathrm{ES}$$

Where $\Gamma = \Gamma_1 + \Gamma_2$. Moreover, we can assume $y \notin \mathsf{fv}(\mathtt{W}\langle v\rangle)$ by α-conversion.
Since $\mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'] \rightarrow_{\mathsf{lsv}} \mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}'$, we can then apply the *i.h.* on Φ'_1, yielding Φ_1 such that $\Phi_1 \rhd \Gamma_1; y : \mathcal{L} \vdash \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'] : \sigma$ and moreover, $\mathtt{sz}\left(\Phi_1\right) \geq \mathtt{sz}\left(\Phi'_1\right)$ and $\mathtt{meas}\left(\Phi_1\right) >_{\mathtt{lex}} \mathtt{meas}\left(\Phi'_1\right)$.
The conclusion of Φ_1 can only be obtained as follows:

$$\dfrac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{11}; x : \mathcal{N} \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \Phi_{v\mathtt{L}'} \rhd \Gamma_{12}; y : \mathcal{L} \vdash v\mathtt{L}' : \mathcal{N}}{(\Gamma_{11} + \Gamma_{12}); y : \mathcal{L} \vdash \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'] : \sigma}\,\mathrm{ES}$$

where $\Gamma_1 = \Gamma_{11} + \Gamma_{12}$. Since $y \notin \mathsf{fv}(\mathtt{W}\langle v\rangle)$ and $x \neq y$ by α-conversion, we can assume that $y \notin \mathsf{fv}(\mathtt{W}\langle x\rangle)$ as well. Therefore, $y \notin \mathsf{dom}(\Gamma_{11})$ by Lemma 3.
We then build Φ as follows:

$$\dfrac{\Phi_{\mathtt{W}\langle x\rangle} \rhd \Gamma_{11}; x : \mathcal{N} \vdash \mathtt{W}\langle x\rangle : \sigma \qquad \dfrac{\Phi_{v\mathtt{L}'} \rhd \Gamma_{12}; y : \mathcal{L} \vdash v\mathtt{L}' : \mathcal{N} \qquad \Phi'_s \rhd \Gamma_2 \vdash s : \mathcal{L}}{\Gamma_{12} + \Gamma_2 \vdash v\mathtt{L}'[y\backslash s] : \mathcal{N}}\,\mathrm{ES}}{\Gamma_{11} + \Gamma_{12} + \Gamma_2 \vdash \mathtt{W}\langle x\rangle[x\backslash v\mathtt{L}'[y\backslash s]] : \sigma}\,\mathrm{ES}$$

Moreover, to calculate $\mathtt{sz}(\Phi)$, we consider two cases depending on whether $y \in \mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')$.

* If $y \in \mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')$, then in particular, $y \in \mathsf{rv}(v\mathtt{L}')$ and thus

$$\begin{aligned}\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle), y\in\mathsf{rv}(v\mathtt{L}')} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + 1 + \mathtt{sz}(\Phi_{v\mathtt{L}'}) + \mathtt{sz}(\Phi'_s)\\ &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}(\Phi_1) + \mathtt{sz}(\Phi'_s)\\ &\geq_{i.h.} 1 + \mathtt{sz}(\Phi'_1) + \mathtt{sz}(\Phi'_s)\\ &=_{y\in\mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')} \mathtt{sz}(\Phi')\end{aligned}$$

* Otherwise, $y \notin \mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')$, so in particular, $y \notin \mathsf{rv}(v\mathtt{L}')$, and thus

$$\begin{aligned}\mathtt{sz}(\Phi) &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle), y\notin\mathsf{rv}(v\mathtt{L}')} 1 + \mathtt{sz}\left(\Phi_{\mathtt{W}\langle x\rangle}\right) + \mathtt{sz}(\Phi_{v\mathtt{L}'}) + \mathtt{sz}(\Phi'_s)\\ &=_{x\in\mathsf{rv}(\mathtt{W}\langle x\rangle)} 1 + \mathtt{sz}(\Phi_1) + \mathtt{sz}(\Phi'_s)\\ &\geq_{i.h.} 1 + \mathtt{sz}(\Phi'_1) + \mathtt{sz}(\Phi'_s)\\ &>_{y\notin\mathsf{rv}(\mathtt{W}\langle v\rangle[x\backslash v]\mathtt{L}')} \mathtt{sz}(\Phi')\end{aligned}$$

We conclude since $\mathtt{sz}(\Phi) \geq \mathtt{sz}(\Phi')$ in the first case, and $\mathtt{sz}(\Phi) > \mathtt{sz}(\Phi')$ in the second case. Moreover, $\#(t) > \#(t')$ by Lemma 2, so $\mathtt{meas}(\Phi) >_{\mathtt{lex}} \mathtt{meas}(\Phi')$.

□

We establish a lemma showing that LCBV-normal forms are typable to prove completeness of system $\mathcal{V}$ with respect to LCBV evaluation. For that, we first define a grammar of terms characterising LCBV-normal forms.

$$\begin{array}{lcl}\mathtt{vr} & ::= & x \mid \mathtt{vr}[x\backslash\mathtt{ne}] \mid \mathtt{vr}[x\backslash\mathtt{no}]\ (x \notin \mathsf{rv}(\mathtt{vr}))\\ \mathtt{ne} & ::= & \mathtt{vr}\,\mathtt{no} \mid \mathtt{ne}\,\mathtt{no} \mid \mathtt{ne}[x\backslash\mathtt{ne}] \mid \mathtt{ne}[x\backslash\mathtt{no}]\ (x \notin \mathsf{rv}(\mathtt{ne}))\\ \mathtt{ab} & ::= & \lambda x.\, t \mid \mathtt{ab}[x\backslash\mathtt{ne}] \mid \mathtt{ab}[x\backslash\mathtt{no}]\ (x \notin \mathsf{rv}(\mathtt{ab}))\\ \mathtt{no} & ::= & \mathtt{vr} \mid \mathtt{ne} \mid \mathtt{ab}\end{array}$$

We now define the predicates $\mathsf{abs}(\_)$, $\mathsf{app}(\_)$, and $\mathsf{var}(\_)$ that we use for the characterisation of LCBV-normal forms. Let t be any term. Then, the predicate $\mathsf{abs}(t)$ holds if and only $t = (\lambda x.\, u)\mathtt{L}$ for some variable x, some term u and some substitution context $\mathtt{L}$, the predicate $\mathsf{app}(t)$ holds if and only $t = (u\, s)\mathtt{L}$ for some terms u, s and some substitution context $\mathtt{L}$, and the predicate $\mathsf{var}(t)$ holds if and only $t = x\mathtt{L}$ for some variable x, and some substitution context $\mathtt{L}$.

Lemma 12 (Characterisation of LCBV-Normal Forms)**.** *Let t be a term. Then, $t \in \mathtt{no}$ if and only if t is* LCBV*-irreducible.*

Proof. We prove simultaneously the following statements:

1. $t \in \mathtt{vr}$ if and only if t is LCBV-irreducible and $\mathsf{var}(t)$
2. $t \in \mathtt{ne}$ if and only if t is LCBV-irreducible and $\mathsf{app}(t)$
3. $t \in \mathtt{ab}$ if and only if t is LCBV-irreducible and $\mathsf{abs}(t)$
4. $t \in \mathtt{no}$ if and only if t is LCBV-irreducible

1. We first prove the *only if* statement, by induction on $t \in \mathtt{vr}$:

- $t = x$. Then, x is LCBV-irreducible since there are no rules to reduce variables, and by definition, $\mathsf{var}(x)$ trivially holds.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{ne}$. Applying the *i.h.* (1) on t_1, then t_1 is LCBV-irreducible and $\mathsf{var}(t_1)$. We thus conclude $\mathsf{var}(t)$.

  Applying the *i.h.* (2) on t_2, then t_2 is LCBV-irreducible and $\mathsf{app}(t_2)$, so even if $x \in \mathsf{fv}(t_1)$, t cannot LCBV-reduce by performing an lsv-step. Since t_1 and t_2 are LCBV-irreducible, we conclude t to be LCBV-irreducible.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{vr}$, $t_2 \in \mathtt{no}$, and $x \notin \mathsf{rv}(t_1)$. Applying the *i.h.* (1) on t_1, then t_1 is LCBV-irreducible and $\mathsf{var}(t_1)$. We thus conclude $\mathsf{var}(t)$. Applying the *i.h.* (4) on t_2, then t_2 is LCBV-irreducible.

  Since $x \notin \mathsf{rv}(t_1)$, t cannot LCBV-reduce by performing an lsv-step by Remark 5. And since t_1 and t_2 are LCBV-irreducible, we then conclude that t is LCBV-irreducible.

We now prove the *if* statement, by induction on t. Cases $t = \lambda x. u$ and $t = t_1\, t_2$ do not hold, as these terms do not satisfy $\mathsf{var}(t)$.

- $t = x$. Straightforward by definition of the grammar of $\mathtt{vr}$.
- $t = t_1[x\backslash t_2]$. Then, both t_1 and t_2 are LCBV-irreducible, since otherwise t would LCBV-reduce.

  Moreover, $\mathsf{var}(t_1)$ since $\mathsf{var}(t)$ holds. Then, $t_1 \in \mathtt{vr}$ by the *i.h.* (1) on t_1. Also, $t_2 \in \mathtt{no}$ by the *i.h.* (4) on t_2.

  On the other hand, there are two cases: $\mathsf{app}(t_2)$, or $x \notin \mathsf{rv}(t_1)$ since otherwise t_2 would be of the form $v\mathsf{L}$, and t would reduce by performing an lsv-step. Then, if the former, $t_2 \in \mathtt{ne}$ by the *i.h.* (2) on t_2. Therefore, we conclude $t_1[x\backslash t_2] \in \mathtt{vr}$ with $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{ne}$. If the latter, we conclude $t_1[x\backslash t_2] \in \mathtt{vr}$ with $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{no}$, and $x \notin \mathsf{rv}(t_1)$.

2. We first prove the *only if* statement, by induction on $t \in \mathtt{ne}$:

- $t = t_1\, t_2$, where $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{no}$. Applying the *i.h.* (1) on t_1, then t_1 is LCBV-irreducible and $\mathsf{var}(t_1)$. Then, t does not reduce by performing a db-step. Applying the *i.h.* (4) on t_2, then t_2 is LCBV-irreducible. We then conclude t to be LCBV-irreducible, and by the form t has, then $\mathsf{app}(t)$.
- $t = t_1\, t_2$, where $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{no}$. Applying the *i.h.* (2) on t_1, then t_1 is LCBV-irreducible and $\mathsf{app}(t_1)$. Then, t does not reduce by performing a db-step. Applying the *i.h.* (4) on t_2, then t_2 is LCBV-irreducible. We then conclude t to be LCBV-irreducible, and by the form t has, then $\mathsf{app}(t)$.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{ne}$. Applying the *i.h.* (2) on t_1, then t_1 is LCBV-irreducible and $\mathsf{app}(t_1)$. We thus conclude $\mathsf{app}(t)$.

  Applying the *i.h.* (2) on t_2, then t_2 is LCBV-irreducible and $\mathsf{app}(t_2)$, so even if $x \in \mathsf{fv}(t_1)$, t cannot LCBV-reduce by performing an lsv-step. And since t_1 and t_2 are LCBV-irreducible, we conclude t to be LCBV-irreducible.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{ne}$, $t_2 \in \mathtt{no}$, and $x \notin \mathsf{rv}(t_1)$. Applying the *i.h.* (2) on t_1, then t_1 is LCBV-irreducible and $\mathsf{app}(t_1)$. We thus conclude $\mathsf{app}(t)$. Applying the *i.h.* (4) on t_2, then t_2 is LCBV-irreducible.

  Since $x \notin \mathsf{rv}(t_1)$, then t cannot LCBV-reduce by performing an lsv-step by Remark 5. And since t_1 and t_2 are LCBV-irreducible, we then conclude that t is LCBV-irreducible.

We now prove the *if* statement, by induction on t. Cases $t = x$ and $t = \lambda x.\, u$ do not hold, as these terms do not satisfy $\mathsf{app}(t)$.

- $t = t_1\, t_2$. Then, both t_1 and t_2 are LCBV-irreducible, since otherwise t would LCBV-reduce. By the *i.h.* (4) on t_2, then $t_2 \in \mathtt{no}$. Moreover, $\neg\mathsf{abs}(t_1)$ since otherwise t would reduce by performing a db-step. And it is easy to note that syntactically, t_1 satisfy either $\mathsf{var}(t_1)$ or $\mathsf{app}(t_1)$. If the former, then $t_1 \in \mathtt{vr}$ by *i.h.* (1). If the latter, then $t_1 \in \mathtt{ne}$ by *i.h.* (2). Either way, we conclude $t_1\, t_2 \in \mathtt{ne}$, so we are done.
- $t = t_1[x\backslash t_2]$. Then, both t_1 and t_2 are LCBV-irreducible, since otherwise t would LCBV-reduce.

  Moreover, $\mathsf{app}(t_1)$ since $\mathsf{app}(t)$ holds. Then, $t_1 \in \mathtt{ne}$ by the *i.h.* (2) on t_1. Also, $t_2 \in \mathtt{no}$ by the *i.h.* (4) on t_2.

  On the other hand, there are two cases: $\mathsf{app}(t_2)$, or $x \notin \mathsf{rv}(t_1)$ since otherwise t_2 would be of the form $v\mathtt{L}$, and t would reduce by performing an lsv-step. Then, if the former, $t_2 \in \mathtt{ne}$ by the *i.h.* (2) on t_2. Therefore, we conclude $t_1[x\backslash t_2] \in \mathtt{ne}$ with $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{ne}$. If the latter, we conclude $t_1[x\backslash t_2] \in \mathtt{ne}$ with $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{no}$, and $x \notin \mathsf{rv}(t_1)$.

3. We first prove the *only if* statement, by induction on $t \in \mathtt{ab}$:

- $t = \lambda x.\, u$. Then, t is LCBV-irreducible since there are no rules to reduce abstractions. Moreover, $\mathsf{abs}(t)$ trivially holds.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{ab}$ and $t_2 \in \mathtt{ne}$. Applying the *i.h.* (3) on t_1, then t_1 is LCBV-irreducible and $\mathsf{abs}(t_1)$, so that we conclude $\mathsf{abs}(t)$. Applying the *i.h.* (2) on t_2, then t_2 is LCBV-irreducible and $\mathsf{app}(t_2)$. Then, t cannot LCBV-reduce by performing an lsv-step. And since t_1 and t_2 are LCBV-irreducible, we conclude t to be LCBV-irreducible.
- $t = t_1[x\backslash t_2]$, where $t_1 \in \mathtt{ab}$ and $t_2 \in \mathtt{no}$ and $x \notin \mathsf{rv}(t_1)$. Applying the *i.h.* (3) on t_1, then t_1 is LCBV-irreducible and $\mathsf{abs}(t_1)$, so that we conclude $\mathsf{abs}(t)$. Applying the *i.h.* (4) on t_2, then t_2 is LCBV-irreducible. If $t_2 \in \mathtt{ne}$, then we proceed as the previous case. Otherwise, if $t_2 \in \mathtt{vr}$ or $t_2 \in \mathtt{ab}$, then t cannot LCBV-reduce by performing an lsv-step by Remark 5, as $x \notin \mathsf{rv}(t_1)$. And since t_1 and t_2 are LCBV-irreducible, t must also be LCBV-irreducible.

We now prove the *if* statement. Since $\mathsf{abs}(t)$, then $t = (\lambda x.\, u)\mathtt{L}$. We proceed by induction on $\mathtt{L}$.

- $\mathtt{L} = \Diamond$. Then, $t = \lambda x.\, u$. Straightforward by definition of the grammar $\mathtt{ab}$.
- $\mathtt{L} = \mathtt{L}'[y\backslash s]$. Then, $t = (\lambda x.\, u)\mathtt{L}'[y\backslash s]$ and since $(\lambda x.\, u)\mathtt{L}'[y\backslash s]$ is LCBV-irreducible, so is $t' = (\lambda x.\, u)\mathtt{L}'$. Moreover, $\mathsf{abs}(t')$ holds. Applying the *i.h.* (3) on t', then $t' \in \mathtt{ab}$.

  On the other hand, s is LCBV-irreducible since t is LCBV-irreducible. Applying the *i.h.* (4) on s, then $s \in \mathtt{no}$. We now consider two cases, depending on whether $y \in \mathsf{rv}(t')$. If $y \notin \mathsf{rv}(t')$, then $t' \in \mathtt{ab}$, and $s \in \mathtt{no}$ imply $t'[y\backslash s] = t \in \mathtt{ab}$. If $y \in \mathsf{rv}(t')$, then $\mathsf{app}(s)$ since otherwise $t = t'[y\backslash s]$ would be LCBV-irreducible. Applying the *i.h.* (2) on s, we yield $s \in \mathtt{ne}$, which implies $t = t'[y\backslash s] \in \mathtt{ab}$.

4. We first prove the *only if* statement, by induction on $t \in \mathtt{no}$:

- $t \in \mathtt{vr}$. Then, t is LCBV-irreducible by Item 1.

- $t \in \mathtt{ne}$. Then, t is LCBV-irreducible by Item 2.
- $t \in \mathtt{ab}$. Then, t is LCBV-irreducible by Item 3.

We now prove the *if* statement, by cases.

- If $\mathsf{var}(t)$, then $t \in \mathtt{vr}$ by the *i.h.* (1). We conclude since $\mathtt{vr} \subseteq \mathtt{no}$.
- If $\mathsf{app}(t)$, then $t \in \mathtt{ne}$ by the *i.h.* (2). We conclude since $\mathtt{ne} \subseteq \mathtt{no}$.
- If $\mathsf{abs}(t)$, then $t \in \mathtt{ab}$ by the *i.h.* (3). We conclude since $\mathtt{ab} \subseteq \mathtt{no}$.

□

Lemma 13 (LCBV-Normal Forms are $\mathcal{V}$-Typable)**.** *Let t be a* LCBV*-normal form. Then, t is $\mathcal{V}$-typable.*

Proof. We prove simultaneously the following statements:

1. If $t \in \mathtt{vr}$, then for every multi-type $\mathcal{M}$ there exist Φ and Γ such that $\Phi \rhd \Gamma \vdash t : \mathcal{M}$.
2. If $t \in \mathtt{ne}$, then for every type σ there exist Φ and Γ such that $\Phi \rhd \Gamma \vdash t : \sigma$.
3. If $t \in \mathtt{ab}$, then there exist Φ and Γ such that $\Phi \rhd \Gamma \vdash t : [\,]$.
4. If $t \in \mathtt{no}$, then there exist Φ, Γ, and $\mathcal{M}$ such that $\Phi \rhd \Gamma \vdash t : \mathcal{M}$.

Moreover, in the four cases, for any variable x, if $x \notin \mathsf{rv}(t)$, then $x \notin \mathsf{dom}(\Gamma)$.

1. We proceed by induction on $t \in \mathtt{vr}$.
   - $t = x$. Let $\mathcal{M}$ by any multi-type. Then, we yield $\Phi \rhd x : \mathcal{M} \vdash x : \mathcal{M}$ by rule VAR. So, it is sufficient to take $\Gamma := x : \mathcal{M}$.
   - $t = t_1[x \backslash t_2]$, where $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{ne}$. Let $\mathcal{M}$ by any multi-type. By the *i.h.* (1) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \rhd \Gamma_1 \vdash t_1 : \mathcal{M}$. By the *i.h.* (2) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \rhd \Gamma_2 \vdash t_2 : \Gamma_1(x)$. Let us write $\Gamma_1 = \Gamma_1'; x : \Gamma_1(x)$. Then, applying rule ES, we yield $\Phi \rhd \Gamma_1' + \Gamma_2 \vdash t_1[x \backslash t_2] : \mathcal{M}$, so that $\Gamma := \Gamma_1' + \Gamma_2$.
   - $t = t_1[x \backslash t_2]$, where $t_1 \in \mathtt{vr}$, $t_2 \in \mathtt{no}$, and $x \notin \mathsf{rv}(t_1)$. Let $\mathcal{M}$ by any multi-type. If $t_2 \in \mathtt{ne}$, then we proceed as in the previous case. If $t_2 \in \mathtt{vr}$ or $t_2 \in \mathtt{ab}$, then by the *i.h.* (1) or (3) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \rhd \Gamma_2 \vdash t_2 : [\,]$. By the *i.h.* (1) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \rhd \Gamma_1 \vdash t_1 : \mathcal{M}$. Moreover, given that $x \notin \mathsf{rv}(t_1)$, then $x \notin \mathsf{dom}(\Gamma_1)$ also holds by the *i.h.*, and thus $\Gamma_1 = \Gamma_1; x : [\,]$. Then, applying the rule ES, we yield $\Phi \rhd \Gamma_1 + \Gamma_2 \vdash t_1[x \backslash t_2] : [\,]$, so that $\Gamma := \Gamma_1 + \Gamma_2$.
2. We proceed by induction on $t \in \mathtt{ne}$.
   - $t = t_1\, t_2$, where $t_1 \in \mathtt{vr}$ and $t_2 \in \mathtt{no}$. Let σ be any type. By the *i.h.* (4) on t_2, there exist Φ_2, Γ_2, and $\mathcal{M}$ such that $\Phi_2 \rhd \Gamma_2 \vdash t_2 : \mathcal{M}$. By the *i.h.* (1) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \rhd \Gamma_1 \vdash t_1 : [\mathcal{M} \to \sigma]$. Then, taking $\Gamma := \Gamma_1 + \Gamma_2$, we yield $\Phi \rhd \Gamma_1 + \Gamma_2 \vdash t_1\, t_2 : \sigma$ by rule APP.
   - $t = t_1\, t_2$, where $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{no}$. Analogous to the previous case.
   - $t = t_1[x \backslash t_2]$, where $t_1 \in \mathtt{ne}$ and $t_2 \in \mathtt{ne}$. Let σ be any type. By the *i.h.* (2) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \rhd \Gamma_1 \vdash t_1 : \sigma$. By the *i.h.* (2) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \rhd \Gamma_2 \vdash t_2 : \Gamma_1(x)$. Let us write $\Gamma_1 = \Gamma_1'; x : \Gamma_1(x)$. Then, applying the rule ES, we yield $\Phi \rhd \Gamma_1' + \Gamma_2 \vdash t_1[x \backslash t_2] : \sigma$, so that $\Gamma := \Gamma_1' + \Gamma_2$.

- $t = t_1[x\backslash t_2]$, where $t_1 \in \texttt{ne}$, $t_2 \in \texttt{no}$, and $x \notin \mathsf{rv}(t_1)$. Let σ be any type. If $t_2 \in \texttt{ne}$, then we proceed as in the previous case.

  If $t_2 \in \texttt{vr}$, then by the *i.h.* (2) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \triangleright \Gamma_1 \vdash t_1 : \sigma$. Let $\Gamma_1 = \Gamma_1'; x : \mathcal{M}$. By the *i.h.* (1) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \triangleright \Gamma_2 \vdash t_2 : \mathcal{M}$. Then, applying the rule ES, we yield $\Phi \triangleright \Gamma_1' + \Gamma_2 \vdash t_1[x\backslash t_2] : \sigma$, so that $\Gamma := \Gamma_1' + \Gamma_2$.

  If $t_2 \in \texttt{ab}$, then by the *i.h.* (3) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \triangleright \Gamma_2 \vdash t_2 : [\,]$. By the *i.h.* (2) on t_1, there exist Φ_1 and Γ_1 such that $\Phi_1 \triangleright \Gamma_1 \vdash t_1 : \sigma$. Given that $x \notin \mathsf{rv}(t_1)$, then $x \notin \mathsf{dom}(\Gamma_1)$, and thus $\Gamma_1 = \Gamma_1; x : [\,]$. Then, applying the rule ES, we yield $\Phi \triangleright \Gamma_1 + \Gamma_2 \vdash t_1[x\backslash t_2] : \sigma$, so that $\Gamma := \Gamma_1 + \Gamma_2$.

3. We proceed by induction on $t \in \texttt{ab}$.

   - $t = \lambda x. u$. Then, the statement holds with typing rule ABS applied to an empty set of premises, *i.e.*, $\Gamma := \emptyset$.
   - $t = t_1[x\backslash t_2]$, where $t_1 \in \texttt{ab}$ and $t_2 \in \texttt{ne}$. By the *i.h.* (3), there exist Φ_1 and Γ_1 such that $\Phi_1 \triangleright \Gamma_1 \vdash t_1 : [\,]$. By the *i.h.* (2) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \triangleright \Gamma_2 \vdash t_2 : \Gamma_1(x)$. Let us write $\Gamma_1 = \Gamma_1'; x : \Gamma_1(x)$. Then, applying rule ES, we yield $\Phi \triangleright \Gamma_1' + \Gamma_2 \vdash t_1[x\backslash t_2] : [\,]$, so that $\Gamma := \Gamma_1' + \Gamma_2$.
   - $t = t_1[x\backslash t_2]$, where $t_1 \in \texttt{ab}$ and $t_2 \in \texttt{no}$ and $x \notin \mathsf{rv}(t_1)$. If $t_2 \in \texttt{ne}$, then we proceed as in the previous case.

     If $t_2 \in \texttt{vr}$ or $t_2 \in \texttt{ab}$, then by the *i.h.* (1) or (3) on t_2, there exist Φ_2 and Γ_2 such that $\Phi_2 \triangleright \Gamma_2 \vdash t_2 : [\,]$. Moreover, given that $x \notin \mathsf{rv}(t_1)$, then $x \notin \mathsf{dom}(\Gamma_1)$ also holds by the *i.h.*, and thus $\Gamma_1 = \Gamma_1; x : [\,]$. Then, applying the rule ES, we yield $\Phi \triangleright \Gamma_1 + \Gamma_2 \vdash t_1[x\backslash t_2] : [\,]$, so that $\Gamma := \Gamma_1 + \Gamma_2$.

4. We proceed by induction on $t \in \texttt{no}$.

   - $t \in \texttt{vr}$. By Item 1, for any type σ, there exist Φ and Γ such that $\Phi \triangleright \Gamma \vdash t : \sigma$, so we are done.
   - $t \in \texttt{ne}$. By Item 2, for any type σ, there exist Φ and Γ such that $\Phi \triangleright \Gamma \vdash t : \sigma$, so we are done.
   - $t \in \texttt{ab}$. By Item 3, there exist Φ and Γ such that $\Phi \triangleright \Gamma \vdash t : [\,]$, so we are done.

□

Proposition 14 (Completeness of System $\mathcal{V}$)**.** *Let t be a* LCBV*-terminating term. Then, t is $\mathcal{V}$-typable.*

Proof. The proof follows from Lemmas 11 and 13. □

Combining the results of soundness and completeness, we yield:

Theorem 15 (System $\mathcal{V}$ characterises LCBV-termination)**.** *A term is $\mathcal{V}$-typable if and only it is* LCBV*-terminating.*

Proof. The *only if* direction holds by Proposition 9, while the *if* direction follows from Proposition 14. □

We already know that system $\mathcal{V}$ characterises VSC-termination [3, Theorem 5]. Therefore, we conclude this section with the main result in this note, establishing a semantic equivalence between LCBV and VSC:

Theorem 16. *A term terminates in* VSC *if and only if it terminates in* LCBV*.*

References

[1] Beniamino Accattoli and Luca Paolini. Call-by-value solvability, revisited. In Tom Schrijvers and Peter Thiemann, editors, *Functional and Logic Programming - 11th International Symposium, FLOPS 2012, Kobe, Japan, May 23-25, 2012. Proceedings*, volume 7294 of *Lecture Notes in Computer Science*, pages 4–16. Springer, 2012.

[2] Pablo Barenbaum, Delia Kesner, and Mariana Milicich. Useful evaluation: Syntax and semantics (technical report), 2025.

[3] Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, and Andrés Viso. The bang calculus revisited. In Keisuke Nakano and Konstantinos Sagonas, editors, *Functional and Logic Programming - 15th International Symposium, FLOPS 2020, Akita, Japan, September 14-16, 2020, Proceedings*, volume 12073 of *Lecture Notes in Computer Science*, pages 13–32. Springer, 2020.

[4] Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, and Andrés Viso. The bang calculus revisited. *Inf. Comput.*, 293:105047, 2023.

[5] Thomas Ehrhard. Collapsing non-idempotent intersection types. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France*, volume 16 of *LIPIcs*, pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.