January 2000

DALI: an untyped CBV operational semantics and equational theory for datatypes with binders (Technical Development)

Emir Paalic

Tim Sheard

Walid Taha

DALI: An Untyped CBV Operational Semantics and Equational Theory for Datatypes with Binders (Technical Development)

Emir Pašalić, Tim Sheard*, Walid Taha**

Oregon Graduate Institute *and* Chalmers University of Technology

Abstract. This report presents the basic definitions and formal development of DALI, a novel extension of the CBV lambda calculus. DALI is based on a proposal by Miller, and provides an elegant and well-behaved notion of object-variables. The notion is *elegant* because it avoids any explicit need for a "gensym" or "newname" operation that is often needed in such systems. The notion is *well-behaved* in that it induces a computationally adequate equational theory.
Our development follows the one employed in Taha's thesis [4] for MetaML[7, 5]. The development was easy to adapt, and some of the major lemmas and proofs remained essentially unchanged. This success is further evidence of the robustness of both the observations of Takahashi on the notion of parallel reduction and the original "standardisation" development by Plotkin for the CBV and CBN lambda calculus.

* The research reported in this paper was supported by the USAF Air Materiel Command, contract # F19628-96-C-0161, NSF Grants CCR-9803880 and IRI-9625462, and the Department of Defense.

** Supported by a Postdoctoral Fellowship, funded by the Swedish Research Council for Engineering Sciences (TFR), grant number 221-96-403.

Table of Contents

1 Preliminaries

1.1 A Note Regarding the First Revision

This document presents the technical development of a core calculus of DALI, a CBV language which provides support for datatypes with bindings. The proofs are given in as much detail as the space would allow it.

Due to the time constraints, many typographical and stylistic changes we wished to make in order to make the document more readable and compact are still missing. We intend to address these problems regarding our presentation in a forthcoming revision. e.p.

The document is organized as follows: Section 2 presents definitions of the various notions that will be used throughtout the proof. Section 3 proves various basic properties of those notions. Section 4 provides the general proof of confluence, taken directly from [5]. Section 5 proves necessary lemmas for DALI that are required for the general argument in section 4. Section 6 gives a general argument for soundness of DALI, adapted directly from [5]. Section 7 provides proofs for DALI-specific lemmas required in Section 6.

1.2 Notes on the Technical Development

The development proceeds in a number of steps. Some of these steps introduce auxiliary constructs that are useful in constructing the proofs of properties in which we are interested. These constructs include well-known notions, such as *complete development* and parallel reduction, which are recast in the setting of DALI.

Each new notion usually has certain basic properties associated with it, about which we prove lemmas (Section 3). These basic properties, beside being useful in latter proofs, also ensure that the definitions of various notions are sensible and correspond to their equivalents in more well-known calculi.

We introduce a big-step semantics (or evaluation function where the order of evaluation of subterms is deterministic), and a reduction semantics (where the order in which the rules are applied is immaterial).

The main result of this paper is that reduction preserves evalaution. That is, applying any sequence of reduction rules to a term does not change the value to which it evaluates. Thus the reduction calculus can be used to transform one program into another program with equivalent meaning.

Our development includes the following steps:

- We define a core language (DALI, Section 2) that exhibits only the essential features and properties we wish to develop theoretically.
- We define a big-step semantics ($\hookrightarrow$), a deterministic partial function that defines the notion of evaluation of DALI programs.
- We define a reduction semantics (λ^d), based on primitive notions of reduction (e.g., β), lifted into arbitrary contexts to obtain a compatible reduction relation ($\longrightarrow$),as well as its reflexive transitive closure ($\longrightarrow^*$).
- We define *parallel reduction*($\gg$), for DALI, a relation between terms that allows multiple reductions to be performed at the same time, nondeterministically.
- We define *left reduction*($\longmapsto$), a function on terms that performs the work of big-step semantics in a number of small steps.
- We prove important properties of these constructs:
  1. λ^d is equivalent to parallel reduction. This allows us to replace reduction semantics with parallel reductions in our proofs, since parallel reduction is considerably less dificult to reason about than λ^d.
  2. We prove equivalence of the transitive closure of left reduction ($\longmapsto^*$) and big-step evaluation. Similarly to parallel reduction, this allows us to reason about $\longmapsto^*$ instead of $\hookrightarrow$ in our proofs which simplifies the development.
- The first major result of our work on DALI is the proof of confluence of λ^d. (Sections 4 and 5) We prove that parallel reduction is confluent, which by equivalence of the transitive closures of λ^d and parallel reduction allows us to conclude that λ^d is confluent as well.
- Based on Taha's soundness proof of MetaML [5], the partiality of left reduction function induces a partitioning of the set of expressions, called expression classes. The three term classes are inductively defined and are called *values*, *workables,* and *stucks*.
  1. *Workables* are expressions on which left reduction is defined, i.e., can be advanced by left reduction. In other words, workables may left-reduce, in one or more steps, to either values, stucks or other workables.

2. *Values* are expressions to which workables may left-reduce in one or more steps but on which left reduction cannot proceed further. They correspond exactly to the set of values defined in Section 2.
3. *Stucks* are expressions that cannot be advanced by left reduction, but are not values.

- A further development is to prove certain monotonic properties of parallel reduction with respect to term classes. In particular, values and stucks can be only further reduced to other values or stucks respectively, and most importantly, if a result of the parallel reduction is a value, it must have been obtained either from another value, or from a workable.
- Finally, the main result of this paper, the soundness of reduction semantics, is reduced to proving two goals

1. A (terminating) evaluation of a term to a value implies a finite sequence of λ^d steps in which the original term reduces to the same value.
2. A finite sequence of reductions of a term e to a value v implies that there exists a value v' to which the original term evaluates, and which can be then reduced in some finite number of steps to the original value v.

The first goal is not difficult to prove by straightforard induction. The second goal, however, requires several transformations:

1. Using the equivalence of evaluation and left reduction's reflexive transitive closure, and the equivalence of λ^d and parallel reduction, the goal is restated in terms of left reduction and parallel reduction.
2. Three important lemmas are used in proving this goal: *push-back*,*transition* and *permutation*. These lemmas allow rearranging of the order of left and parallel reductions in finite reduction chains until the goal is reached.

The formulation of the soundness proof technique is most directly due to Taha's work on MetaML[4,5]. In proving soundness of DALI, we were able to directly directly reuse both the structure and certain key lemmas of this proof. For others, it was necessary to re-prove them in the context of the new language, but their basic formulation remained unchanged.

2 Definitions

2.1 $\mathbb{X}, \mathbb{Z}, \mathbb{F}, \mathbb{E}, \mathbb{C}, \mathbb{V}, \mathbb{B}, \mathbb{W}, \mathbb{S}$ Set Definitions

We will use a **BNF-like** notation for specifying a number of inductively defined sets that will be used throughout this report. Two non-standard notations will be used for conciseness:

- **Lambda terms with patterns** are written $\lambda^{f \in \{f_1,...,f_n\}}(f\ \ x_f).e_f$, meaning, a sequence of simple lambda abstractions $\lambda x_1.e_1, ..., \lambda x_n.e_n$ corresponding to branches of a lambda abstraction indexed by the various tags $f_1, ..., f_n$.
- **Set difference** is written $\mathbb{V} \setminus v$, meaning, all possible elements of $\mathbb{V}$ except the ones matching the pattern particular pattern v. For example, if $\mathbb{N}$ is the set of naturals, then $n+1$ is a pattern that matches naturals greater than zero, and $\mathbb{N} \setminus (n+1)$ is simply the natural number zero. This notation allows us to avoid having a definition of stuck terms $\mathbb{S}$ that is quadratic in the number of constructs in expressions $\mathbb{E}$.

We will make use of **Barendregt's variable convention**, and state the set of bound and free variables in expressions occurring in any formula or statement should be taken by the reader to be distinct.

Definition 1

$$
\begin{array}{rlll}
x \in \mathbb{X} & \textit{Normal variables} & := & \textit{Infinite set of names} \\
z \in \mathbb{Z} & \textit{Object variables} & := & \textit{Infinite set of names} \\
f \in \mathbb{F} & \textit{Tags} & := & \textit{Infinite set of names containing } \mathsf{True} \textit{ and } \mathsf{False} \\
F \subset \mathbb{F} & \textit{Tag sets} & = & \textit{Finite subsets of } \mathbb{F} \\
e \in \mathbb{E} & \textit{Expressions} & := & () \mid x \mid \lambda x.e \mid e\ e \mid (e,e) \mid \pi_1\ e \mid \pi_2\ e \mid f\ e \mid \lambda^{f \in F}(f\ x_f).e_f \mid \\
 & & & \#z \mid \#z \Rightarrow e \mid \lambda(\#z \Rightarrow x).e \mid \mathsf{isOVar}\ e \mid e =_{\#} e \\
C \in \mathbb{C} & \textit{Contexts} & := & [] \mid \lambda x.C \mid C\ e \mid e\ C \mid (e,C) \mid (C,e) \mid \pi_1\ C \mid \pi_2\ C \mid \\
 & & & \lambda^{f \in F - \{f'\}}((f_i\ x_i).e_i) + +(f'\ x).C \mid f\ C \mid (\#z \Rightarrow C) \mid \lambda(\#z \Rightarrow x).C \\
 & & & \mid \mathsf{isOVar}\ C \mid C =_{\#} e \mid e =_{\#} C \\
b \in \mathbb{B} & \textit{Based Values} & := & () \mid (b,b) \mid f\ b \mid \#z \mid \#z \Rightarrow b \\
v \in \mathbb{V} & \textit{Values} & := & () \mid \lambda x.e \mid (v,v) \mid f\ v \mid \lambda^{f \in F} f\ x_f.e_f \mid \#z \mid \#z \Rightarrow v \mid \lambda(\#z \Rightarrow x).e \\
w \in \mathbb{W} & \textit{Workables} & := & (\lambda x.e)\ v \mid (\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \mid w\ e \mid v\ w \mid f\ w \mid \#z \Rightarrow w \mid \\
 & & & (w,e) \mid (v,w) \mid \pi_n\ w \mid \pi_n(v,v) \\
 & & & \mid w = e \mid \#z = w \mid \#z = \#z' \\
 & & & \mid \mathsf{isOVar}\ v \mid \mathsf{isOVar}\ w \\
s \in \mathbb{S} & \textit{Stucks} & := & x \mid s\ e \mid v\ s \mid (s,e) \mid (v,s) \mid \pi_n\ s \mid \pi_n\ (\mathbb{V} \setminus (v,v)) \mid f\ s \\
 & & & \mid (\lambda^{f \in F} f\ x_f.e_f)\ (\mathbb{V} \setminus f\ v) \mid (\#z \Rightarrow s) \\
 & & & \mid (\mathbb{V} \setminus \#z) = e \mid \#z = (\mathbb{V} \setminus \#z) \mid s = e \mid \#z = s \\
 & & & \mid (\lambda(\#z \Rightarrow x).e)\ (\mathbb{V} \setminus (\#z \Rightarrow b)) \\
 & & & \mid (\mathbb{V} \setminus \lambda^e)\ v \text{ where } \lambda^e = \lambda x.e + \lambda^{f \in F} f\ x_f.e_f + \lambda(\#z \Rightarrow x).e \\
 & & & \mid \mathsf{isOVar}\ s \\
\rho \in \mathbb{R} & \textit{Reductions} & := & \beta_1 \mid \pi_1 \mid \pi_2 \mid \beta_2 \mid \beta_3 \mid \# \mid \delta_{\mathsf{isOVar}}
\end{array}
$$

2.2 $\boxed{\_[\_]\colon \mathbb{C} \times \mathbb{E} \to \mathbb{E}}$ Context filling

Definition 2

$$
\begin{array}{c}
[\,][e'] = e' \quad (\lambda x.C)[e'] = \lambda x.(C[e']) \quad (\lambda \#z \Rightarrow x.C)[e'] = (\lambda \#z \Rightarrow x.C[e']) \quad (C\ e)[e'] = C[e']\ e \quad (e\ C)[e'] = e\ C[e'] \\
(\#z \Rightarrow C)[e'] = (\#z \Rightarrow C[e']) \quad (C =_{\#} e)[e'] = C[e'] = e \quad (e =_{\#} C)[e'] = e = C[e'] \\
(e,C)[e'] = (e,C[e']) \quad (C,e)[e'] = (C[e'],e) \quad (\pi_1\ C)[e'] = \pi_1\ (C[e']) \quad (\pi_2\ C)[e'] = \pi_2\ (C[e']) \quad (f\ C)[e'] = f\ (C[e']) \\
(\lambda^{f \in F - \{f'\}}((f\ x_i).e_i) + +(f'\ x).C)[e'] = (\lambda^{f \in F - \{f'\}}((f\ x_i).e_i) + +(f'\ x).C[e']) \\
(\mathsf{isOVar}\ C)[e'] = \mathsf{isOVar}\ C[e']
\end{array}
$$

2.3 $\boxed{\_[\_:=\_]\colon \mathbb{E} \times \mathbb{X} \times \mathbb{E} \to \mathbb{E}}$ and $\boxed{\_[\#\_:=\_]\colon \mathbb{B} \times \mathbb{Z} \times \mathbb{X} \to \mathbb{E}}$ Substitution

Definition 3

$$
\begin{array}{rl}
()[x:=e_3] = () & \\
x[x:=e_3] = e_3 & \\
x'[x:=e_3] = x', \quad x \neq x' & \\
\lambda x'.e[x:=e_3] = \lambda x'.(e[x:=e_3]) & \\
e_1\ e_2[x:=e_3] = (e_1[x:=e_3])\ (e_2[x:=e_3]) & \\
(e_1,e_2)[x:=e_3] = (e_1[x:=e_3], e_2[x:=e_3]) & \qquad ()[\#z:=x] = () \\
\pi_1\ e\ [x:=e_3] = \pi_1\ (e[x:=e_3]) & \qquad f\ b[\#z:=x] = f\ (b[\#z:=x]) \\
\pi_2\ e\ [x:=e_3] = \pi_2\ (e[x:=e_3]) & \qquad \#z[\#z:=x] = x \\
f\ e_1[x:=e_3] = f\ (e_1[x:=e_3]) & \qquad \#z'[\#z:=x] = \#z', \quad \#z' \neq \#z \\
\lambda^{f \in F}(f x_f).e_f[x:=e_3] = \lambda^{f \in F}(f x'_f).(e_f[x:=e_3]) & \qquad \#z' \Rightarrow b[\#z:=x] = z' \Rightarrow (b[\#z:=x]) \\
\#z[x:=e_3] = \#z & \\
\#z' \Rightarrow e[x:=e_3] = \#z' \Rightarrow (e[x:=e_3]) & \\
\lambda(\#\_ \Rightarrow x').e[x:=e_3] = \lambda(\#\_ \Rightarrow x').(e[x:=e_3]) & \\
e_1 =_{\#} e_2[x:=e_3] = (e_1[x:=e_3]) =_{\#} (e_2[x:=e_3]) & \\
\mathsf{isOVar}\ e[x:=e_3] = \mathsf{isOVar}(e[x:=e_3]) &
\end{array}
$$

Remark 4 *Note that the above definition of substitution uses Barendregt's variable convention, and thus no explicit side-conditions on renaming substitutions over binding constructs are necessary.*

2.4 $\boxed{\_ \longrightarrow \_ \_ : \mathbb{E} \times \mathbb{R} \times \mathbb{E} \to \mathbb{E}}$ Notions of Reduction

Definition 5

$$
\begin{array}{rcll}
(\lambda x.e)\ v & \longrightarrow_{\beta_1} & e[x := v] & \\
\pi_1\ (v_1, v_2) & \longrightarrow_{\pi_1} & v_1 & \\
\pi_2\ (v_1, v_2) & \longrightarrow_{\pi_2} & v_2 & \\
(\lambda^{i \in L - \{k\}} (f\ x_i).e_i)\ (k\ v) & \longrightarrow_{\beta_2} & e_k[x_k := e] & \\
(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) & \longrightarrow_{\beta_3} & e[x := \lambda x.(b[\#z := x])] & \\
\#z =_{\#} \#z & \longrightarrow_{\#} & \mathsf{True}() & \\
\#z_1 =_{\#} \#z_2 & \longrightarrow_{\#} & \mathsf{False}() & \textit{if}\ \ z_1 \neq z_2 \\
\mathsf{isOVar}\ \#z & \longrightarrow_{\mathsf{isOVar}} & \mathsf{True}() & \\
\mathsf{isOVar}\ v & \longrightarrow_{\mathsf{isOVar}} & \mathsf{False}() & \textit{if}\ v \neq \#z
\end{array}
$$

2.5 $\boxed{\_ \longrightarrow \_ \subset \mathbb{E} \times \mathbb{E}}$ and $\boxed{\_ \longrightarrow^* \_ \subset \mathbb{E} \times \mathbb{E}}$ Compatible Reduction and the Reduction Relation

Definition 6

$$
\frac{e_1 \longrightarrow_\rho e_2}{C[e_1] \longrightarrow C[e_2]} \rho \in \mathbb{R} \qquad \frac{}{e \longrightarrow^* e} \qquad \frac{e_1 \longrightarrow e_2 \quad e_2 \longrightarrow^* e_3}{e_1 \longrightarrow^* e_3}
$$

2.6 Notation

Notation 7 (Relation Composition) *For any two relations $\oplus$ and $\otimes$, we write $a \oplus b \otimes c$ as a shorthand for $(a \oplus b) \wedge (b \otimes c)$.*

2.7 $\boxed{\_ \gg \_ \subseteq \mathbb{E} \times \mathbb{E}}$ Parallel Reduction

In order to prove the two key lemmas presented in this section, we will need to reason by induction on the "complexity" of parallel reduction. Thus, we will use the following definition of parallel reduction with an associated complexity measure. Where complexity is not relevant, we will simply omit it to avoid unnecessary clutter.

2.8 $\boxed{\_ \overset{M}{\gg} \_ : \mathbb{E} \to \mathbb{E}}$ Parallel Reduction with Complexity

$$\frac{}{() \overset{0}{\gg} ()} \qquad \frac{}{x \overset{0}{\gg} x} \qquad \frac{e_1 \overset{N}{\gg} e_2}{\lambda x.e_1 \overset{N}{\gg} \lambda x.e_2} \qquad \frac{e_1 \overset{M}{\gg} e_3 \quad e_2 \overset{N}{\gg} e_4}{e_1\ e_2 \overset{M+N}{\gg} e_3\ e_4} \qquad \frac{e_1 \overset{M}{\gg} e_3 \quad v_1 \overset{N}{\gg} v_2}{(\lambda x.e_1)\, v_1 \overset{M+\#(x,e_3)N+1}{\gg} e_3[x := v_2]}$$

$$\frac{e_1 \overset{M}{\gg} e_2}{(\lambda(\#z \Rightarrow x).e_1)\ (\#z \Rightarrow b) \overset{M+\#(x,e_2)N+1}{\gg} e_2[x := \lambda x.b[\#z := x]]} \qquad \frac{e_k \overset{M}{\gg} e'_k \quad v_1 \overset{N}{\gg} v_2}{(\lambda^{i \in F} f_i\ x_i.e_i)\ (f_k\ v_1) \overset{M+\#(x_k,e'_k)N+1}{\gg} e'_k[x_k := v_2]}$$

$$\frac{e_1 \overset{M}{\gg} e_3 \quad e_2 \overset{N}{\gg} e_4}{(e_1, e_2) \overset{M+N}{\gg} (e_2, e_4)} \qquad \frac{e_1 \overset{M}{\gg} e_2}{\pi_1\ e_1 \overset{M}{\gg} \pi_1\ e_2} \qquad \frac{e_1 \overset{M}{\gg} e_2}{\pi_2\ e_1 \overset{M}{\gg} \pi_2\ e_2} \qquad \frac{v_1 \overset{N}{\gg} v'_1}{\pi_1\ (v_1, v_2) \overset{N+1}{\gg} v'_1} \qquad \frac{v_2 \overset{N}{\gg} v'_2}{\pi_2\ (v_1, v_2) \overset{N+1}{\gg} v'_2}$$

$$\frac{e_1 \overset{M}{\gg} e_2}{f\ e_1 \overset{M}{\gg} f\ e_2} \qquad \frac{e_f \overset{N_f}{\gg} e'_f}{\lambda^{f \in F} f\ x_f.e_f \overset{\Sigma N_f}{\gg} \lambda^{f \in F} f\ x_f.e'_f} \qquad \frac{}{\#z \overset{0}{\gg} \#z} \qquad \frac{e_1 \overset{M}{\gg} e_2}{(\#z \Rightarrow e_1) \overset{M}{\gg} (\#z \Rightarrow e_2)} \qquad \frac{e_1 \overset{N}{\gg} e_2}{\lambda(\#z \Rightarrow x).e_1 \overset{N}{\gg} \lambda(\#z \Rightarrow x).e_2}$$

$$\frac{}{\#z =_{\#} \#z \overset{1}{\gg} \mathsf{True}()} \qquad \frac{\#z_1 \neq \#z_2}{\#z_1 =_{\#} \#z_2 \overset{1}{\gg} \mathsf{False}()} \qquad \frac{e_1 \overset{M}{\gg} e_2 \quad e_2 \overset{N}{\gg} e_4}{e_1 =_{\#} e_2 \overset{M+N}{\gg} e_2 =_{\#} e_4}$$

$$\frac{e \overset{X}{\gg} e'}{\mathsf{isOVar}\ e \overset{X}{\gg} \mathsf{isOVar}\ e'} \qquad \frac{v \neq \#z}{\mathsf{isOVar}\ v \overset{1}{\gg} \mathsf{False}()}$$

$$\frac{}{\mathsf{isOVar}\ \#z \overset{1}{\gg} \mathsf{True}()}$$

where $\#(x, e)$ is the number of free occurrences of the variable x in the term e.

2.9 $\boxed{!\_ : \mathbb{E} \to \mathbb{E}}$ Complete Development

Definition 8

$$\begin{aligned}
!x &\equiv x \\
!(\lambda x.e) &\equiv \lambda x.!e \\
!(e_1\ e_2) &\equiv !e_1\ !e_2 \ \textit{if}\ e_1\ e_2 \not\equiv \begin{cases} (\lambda x.e_3)v_4 \\ (\lambda^{i \in F \cup \{k\}} (f\ x_f.e_f))\ (k\ v) \\ (\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \end{cases} \\
!((\lambda x.e)\ v) &\equiv !e[x := !v] \\
!(e_1, e_2) &\equiv (!e_1, !e_2) \\
!(\pi_n\ e) &\equiv \pi_n\ !e \quad e \not\equiv (v_1, v_2) \\
!(\pi_1\ (v_1, v_2)) &\equiv !v_1 \\
!(\pi_2\ (v_1, v_2)) &\equiv !v_2 \\
!(f\ e) &\equiv f\ !e \\
!((\lambda^{f \in F \cup \{k\}} f\ x_f.e_f)\ (kv)) &\equiv !e_k[x_k := !v] \\
!\#z &\equiv \#z \\
!(\#z \Rightarrow e) &\equiv \#z \Rightarrow !e \\
!((\lambda(\#z \Rightarrow x).e)(\#z \Rightarrow b)) &\equiv !e[x := \lambda x'.b[\#z := x']] \\
!(e_1 =_{\#} e_2) &\equiv !e_1 =_{\#} !e_2 \quad \textit{if}\ e_1, e_2 \notin \mathbb{Z} \\
!(\#z = \#z) &\equiv \mathsf{True}() \\
!(\#z_1 = \#z_2) &\equiv \mathsf{False}() \quad \textit{if}\ \#z_1 \neq \#z_2 \\
!(\mathsf{isOVar} e) &\equiv \mathsf{isOVar} !e \quad \textit{if} e \notin \mathbb{V} \\
!(\mathsf{isOVar} \#z) &\equiv True() \\
!(\mathsf{isOVar} v) &\equiv False() \quad \textit{if} e \neq \#z
\end{aligned}$$

2.10 $\boxed{\_ \hookrightarrow \_ : \mathbb{E} \to \mathbb{E}}$ Big-Step Semantics

Definition 9

$$\frac{}{() \hookrightarrow ()} \qquad \frac{}{\lambda x.e \hookrightarrow \lambda x.e} \qquad \frac{\begin{array}{l} e_1 \hookrightarrow \lambda x.e \\ e_2 \hookrightarrow e_3 \\ e[x := e_3] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4} \qquad \frac{\begin{array}{l} e_1 \hookrightarrow \lambda^{f \in F \cup \{k\}}\ (f\ x_f).e_f \\ e_2 \hookrightarrow k\ e_4 \\ e_k[x := e_4] \hookrightarrow e_5 \end{array}}{e_1\ e_2 \hookrightarrow e_5} \qquad \frac{\begin{array}{l} e_1 \hookrightarrow \lambda(\#\_ \Rightarrow x).e \\ e_2 \hookrightarrow \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4}$$

$$\frac{e_1 \hookrightarrow e_3 \quad e_2 \hookrightarrow e_4}{(e_1, e_2) \hookrightarrow (e_3, e_4)} \qquad \frac{e \hookrightarrow (e_3, e_4)}{\pi_1\ e \hookrightarrow e_3} \qquad \frac{e \hookrightarrow (e_3, e_4)}{\pi_2\ e \hookrightarrow e_4} \qquad \frac{e_1 \hookrightarrow e_2}{f_k\ e_1 \hookrightarrow f_k\ e_2} \qquad \frac{}{\lambda^{f \in F} f\ x_f.e_f \hookrightarrow \lambda^{i \in F} f\ x_f.e_f}$$

$$\frac{}{\#z \hookrightarrow \#z} \qquad \frac{e_1 \hookrightarrow e_2}{\#z \Rightarrow e_1 \hookrightarrow \#z \Rightarrow e_2} \qquad \frac{}{\lambda(z \Rightarrow x).e \hookrightarrow \lambda(z \Rightarrow x).e} \qquad \frac{\begin{array}{l} e_1 \hookrightarrow \#z \\ e_2 \hookrightarrow \#z \end{array}}{e_1 =_\# e_2 \hookrightarrow \mathsf{True}()} \qquad \frac{\begin{array}{ll} e_1 \hookrightarrow \#z_1 & \\ e_2 \hookrightarrow \#z_2 & z_1 \neq z_2 \end{array}}{e_1 =_\# e_2 \hookrightarrow \mathsf{False}()}$$

$$\frac{e \hookrightarrow \#z}{\mathsf{isOVar}\ e \hookrightarrow \mathsf{True}()} \qquad \frac{e \hookrightarrow v \quad v \neq \#z}{\mathsf{isOVar}\ e \hookrightarrow \mathsf{False}()}$$

Remark 10 *Note that the use of b in the definition of the semantics of the application of a case over object-binders is an expensive runtime check. However, we expect that it should be possible to eliminate the need for this check by an appropriate type system that restricts "analysable" terms to base values.*

2.11 $\boxed{\_ \longmapsto \_ : \mathbb{E} \to \mathbb{E}}$ Left Reduction

The notion of left reduction is intended to capture precisely the reductions performed by the big-step semantics, in a small-step manner. Note that the simplicity of the definition depends on the fact that the partial function being defined is *not* defined on values. That is, we expect that there is no e such that $v \longmapsto e$.

Lemma 22 says that the set of workables characterises exactly the set of terms that can be advanced by left reduction.

Definition 11

$$\frac{}{(\lambda x.e)\ v \longmapsto e[x := v]} \qquad \frac{e_1 \longmapsto e_1'}{e_1\ e_2 \longmapsto e_1'\ e_2} \qquad \frac{e \longmapsto e'}{v\ e_2 \longmapsto v\ e'} \qquad \frac{e_1 \longmapsto e_1'}{(e_1, e_2) \longmapsto (e_1', e_2)} \qquad \frac{e \longmapsto e'}{(v, e) \longmapsto (v, e')}$$

$$\frac{e \longmapsto e'}{\pi_n\ e \longmapsto \pi_n e'} \qquad \frac{}{\pi_1\ (v_1, v_2) \longmapsto v_1}$$

$$\frac{}{\pi_2\ (v_1, v_2) \longmapsto v_2} \qquad \frac{}{(\lambda^{f \in L \cup \{k\}}\ (f\ x_f).e_f)\ (k\ v) \longmapsto e_k[x_k := v]} \qquad \frac{e \longmapsto e'}{f\ e \longmapsto f\ e'}$$

$$\frac{e \longmapsto e'}{\#z \Rightarrow e \longmapsto \#z \Rightarrow e'} \qquad \frac{}{(\lambda(\#z \Rightarrow x).e)\ (\#z' \Rightarrow b) \longmapsto e[x := \lambda x.b[\#z' := x]]}$$

$$\frac{e_1 \longmapsto e_1'}{e_1 =_\# e_2 \longmapsto e_1' =_\# e_2} \qquad \frac{e \longmapsto e'}{\#z' =_\# e \longmapsto \#z' =_\# e'} \qquad \frac{}{\#z =_\# z \longmapsto \mathsf{True}()} \qquad \frac{\#z \neq \#z'}{\#z =_\# z' \longmapsto \mathsf{False}()}$$

$$\frac{e \longmapsto e'}{\mathsf{isOVar}\ e \longmapsto \mathsf{isOVar}\ e'} \qquad \frac{}{\mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()} \qquad \frac{v \neq \#z}{\mathsf{isOVar}\ v \longmapsto \mathsf{False}()}$$

3 Basic Properties

3.1 Substitution

Lemma 12 (Basic Properties of Substitution) $\forall e, e_1, e_2 \in \mathbb{E}$.

1. $x \neq y \wedge x \notin FV(e_2) \Longrightarrow$

$$e[x := e_1][y := e_2] = e[y := e_2][x := e_1[x := e_2]]$$

2. $\#x \neq \#y \wedge \#x \notin FV(e_2) \Longrightarrow$

$$e[\#x := e_1][\#y := e_2] = e[\#y := e_2][\#x := e_1[\#x := e_2]]$$

Proof (Lemma 12). The proof easily follows by structural induction on the expression $e\Box$.

3.2 Parallel Reduction

Lemma 13 (Compatibility of Parallel Reduction) $\forall C \in \mathbb{C}.\, \forall e_1, e_2 \in \mathbb{E}.$

$$e_1 \gg e_2 \implies C[e_1] \gg C[e_2]$$

Proof (Lemma 13). Prof is by structural induction on the context C.

1. $[][e_1] \gg [][e_2]$
2. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\lambda x.C)[e1] \gg (\lambda x.C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
3. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2] \quad e \gg e}{(C\ e)[e_1] \gg (C\ e)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
4. $[\,]_{\gg} \Uparrow \dfrac{e \gg e \quad C[e_1] \gg C[e_2]}{(e\ C)[e_1] \gg (e\ C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
5. $[\,]_{\gg} \Uparrow \dfrac{e \gg e \quad C[e_1] \gg C[e_2]}{(e, C)[e_1] \gg (e, C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
6. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2] \quad e \gg e}{(C, e)[e_1] \gg (C, e)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
7. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\pi_1\ C)[e_1] \gg (\pi_1\ C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
8. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\pi_2\ C)[e_1] \gg (\pi_2\ C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
9. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\lambda^{f \in F - \{f'\}}((f_i\ x_i).e_i) +\!\!+ (f'\ x).C)[e_1] \gg (\lambda^{f \in F - \{f'\}}((f_i\ x_i).e_i) +\!\!+ (f'\ x).C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
10. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(f\ C)[e_1] \gg (f\ C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
11. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\#z \Rightarrow C)[e_1] \gg (\#z \Rightarrow C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
12. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\lambda(\#z \Rightarrow x).C)[e_1] \gg (\lambda(\#z \Rightarrow x).C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
13. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2] \quad e \gg e}{(C =_{\#} e)[e_1] \gg (C =_{\#} e)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
14. $[\,]_{\gg} \Uparrow \dfrac{e \gg e \quad C[e_1] \gg C[e_2]}{(e = \#C)[e_1] \gg (e = \#C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$
15. $[\,]_{\gg} \Uparrow \dfrac{C[e_1] \gg C[e_2]}{(\mathsf{isOVar}\ C)[e_1] \gg (\mathsf{isOVar}\ C)[e_2]}\ {}_{IH_{\gg},[;]}\Downarrow$

$\Box$ [e.p.]

Lemma 14 (Parallel Reduction Properties) $\forall e_1 \in \mathbb{E}.$

1. $\forall b \in \mathbb{B}, e \in \mathbb{E}.\, b \gg e \Longrightarrow e \equiv b$
2. $\forall e \in \mathbb{E}.\, e \gg e.$

3. $\forall \rho. \forall e_1, e_2 \in \mathbb{E}.\ e_1 \longrightarrow_\rho e_2 \Longrightarrow e_1 \gg e_2$
4. $\forall e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow e_1 \gg e_2$
5. $\forall e_2 \in \mathbb{E}. e_1 \gg e_2 \Longrightarrow e_1 \longrightarrow^* e_2$
6. $\forall e_3 \in \mathbb{E}. e_2, e_4 \in \mathbb{E}. e_1 \gg e_3, e_2 \gg e_4 \implies e_1[y{:=}e_2] \gg e_3[y{:=}e_4]$.

Remark 15 *Note that from 1 it follows that*

1. $b \longrightarrow e \Longrightarrow e \equiv b$
2. $!b \equiv b$
3. $b \longmapsto^* e \Longrightarrow e \equiv b$

Proof (Lemma 14).

Part 1 is by a simple induction over the derivation of $b \in \mathbb{B}$. Part 2 is by a simple structural induction on e. Part 3 is by a case analysis over ρ.

1. β_1

$$(\lambda x.e)\ v \longrightarrow_{\beta_1} e[x{:=}v] \text{ and } \frac{e \gg e \quad v \gg v}{(\lambda x.e)\ v \gg e[x{:=}v]} \gg, \text{by } 14.2 \Downarrow$$

2. π_1

$$\pi_1\ (v_1, v_2) \longrightarrow_{\pi_1} v_1 \text{ and } \frac{v_1 \gg v_1}{\pi_1\ (v_1, v_2) \gg v_1} \gg \Downarrow$$

3. π_2

$$\pi_2\ (v_1, v_2) \longrightarrow_{\pi_1} v_2 \text{ and } \frac{v_2 \gg v_2}{\pi_1\ (v_1, v_2) \gg v_2} \gg \Downarrow$$

4. β_2

$$(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \longrightarrow_{\beta_2} e[x{:=}\lambda x.b[\#z{:=}x]] \text{ and } \frac{e \gg e}{(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \gg e[x{:=}\lambda x.b[\#z{:=}x]]} \gg, \text{by } 14.2 \Downarrow$$

5. β_3

$$(\lambda^{i \in \{k\} \cup L}(f_i\ x_i).e_i)\ (f_k\ v) \rightarrow_{\beta_3} e_k[x_k{:=}e] \text{ and } \frac{e_n \gg e_n \quad v \gg v}{(\lambda^{i \in \{k\} \cup L}(f_i\ x_i).e_i)\ (f_k\ v) \gg e_n[x_k{:=}v]} \gg, \text{by } 14.2 \Downarrow$$

6. $\#$

$$\#z =_\# \#z \longrightarrow_\# \mathsf{True}() \quad \text{and} \quad \#z =_\# \#z \gg \mathsf{True}$$

$$\frac{\#z \neq \#z'}{\#z = \#z' \longrightarrow_\# \mathsf{False}()} \quad \text{and} \quad \frac{\#z \neq \#z'}{\#z =_\# \#z' \gg \mathsf{False}()} \gg \Downarrow$$

7. δ_{isOVar}

$$\mathsf{isOVar}\ \#z \longrightarrow \mathsf{True}() \text{ and } \frac{}{\mathsf{isOVar}\ \#z \gg \mathsf{True}}$$

8. δ_{isOVar}

$$\mathsf{isOVar}\ v \longrightarrow \mathsf{False}() \text{ where } v \neq \#z, \text{ and } \frac{v \neq \#z}{\mathsf{isOVar}\ v \gg \mathsf{False}()}\ .$$

Proof (Property 4 of Lemma 14). If we look at the definition of $\longrightarrow$, we notice that if $e_1 \longrightarrow e_2$, then there must exist some context C, so that $e_1 = C[e']$, $e_2 = C[e'']$, and $e' \longrightarrow_\rho e''$.

Thus, to show that $e_1 \longrightarrow e_2 \Longrightarrow e_1 \gg e_2$, it is enough to prove that for any context $C \in \mathbb{C}$, $C[e'] \longrightarrow C[e''] \Longrightarrow C[e'] \gg C[e'']$. This, however, follows directly from Lemma 13 (Compatibility of Parallel Reduction).

Proof (Property 5 of Lemma 14). $\forall e_1 \in \mathbb{E}. \forall e_2 \in \mathbb{E}. e_1 \gg e_2 \implies e_1 \longrightarrow^* e_2$ By induction on the height of the derivation $e_1 \gg e_2$.

1. $() \gg ()$and $() \longrightarrow^0 ()$
2. $x \gg x$ and $x \longrightarrow^0 x$.
3. $$\gg\Uparrow \frac{e \gg e'}{\lambda x.e \gg \lambda x.e'} \overset{IH}{\Longrightarrow} \frac{e \longrightarrow^* e'}{\lambda x.e \longrightarrow^* \lambda x.e'} \text{ comp.} \longrightarrow^* \Downarrow$$
4. $$\gg\Uparrow \frac{\begin{array}{c} e_1 \gg e_1' \\ e_2 \gg e_2' \end{array}}{e_1\ e_2 \gg e_1'\ e_2'} \overset{IH}{\Longrightarrow} \frac{\begin{array}{c} e_1 \longrightarrow^* e_1' \\ e_2 \longrightarrow^* e_2' \end{array}}{e_1\ e_2 \longrightarrow^* e_1'\ e_2'} \text{ comp.} \longrightarrow^* \Downarrow$$
5. $$\gg\Uparrow \frac{\begin{array}{c} v \gg v' \\ e \gg e' \end{array}}{(\lambda x.e)\ v \gg e'[x := v']} \overset{IH}{\Longrightarrow} \frac{\begin{array}{c} v \longrightarrow^* v' \\ e \longrightarrow^* e' \end{array}}{(\lambda x.e)\ v \longrightarrow^* (\lambda x.e')\ v \longrightarrow^* (\lambda x.e')\ v' \longrightarrow_{\beta_1} e'[x := v']} \text{ comp.} \longrightarrow^* \Downarrow$$
6. $$\gg\Uparrow \frac{\begin{array}{c} b \gg b' \\ e \gg e' \end{array}}{(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \gg e'[x := \lambda x'.b'[\#z := x']]} \overset{IH \Downarrow \ IH \Downarrow}{\begin{array}{c} b \longrightarrow^* b' \\ e \longrightarrow^* e' \end{array}} \frac{}{(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \longrightarrow^* (\lambda(\#z \Rightarrow x).e')\ (\#z \Rightarrow b') \longrightarrow_{\beta_2} e'[x := \lambda x.b'[\#z := x]]} \text{ comp.} \longrightarrow^* \Downarrow$$
7. $$\gg\Uparrow \frac{\begin{array}{c} e_k \gg e_k' \\ v \gg v' \end{array}}{(\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f)\ (k\ v) \gg e_k'[x := v']} \overset{IH \Downarrow \ IH \Downarrow}{\begin{array}{c} e_k \longrightarrow^* e_k' \\ v \longrightarrow^* v' \end{array}} \frac{}{(\lambda^{f \in F}\ f\ x_f.e_f | k\ x_k.e_k)\ (k\ v) \longrightarrow^* (\lambda^{f \in F}\ f\ x_f.e_f | k\ x_k.e_k')\ (k\ v') \longrightarrow_{\beta_3} e_k'[x := v']} \text{ comp.} \longrightarrow^* \Downarrow$$
8. $$\gg\Uparrow \frac{\begin{array}{c} e_1 \gg e_1' \\ e_2 \gg e_2' \end{array}}{(e_1, e_2) \gg (e_1', e_2')} \overset{IH}{\Longrightarrow} \frac{\begin{array}{c} e_1 \longrightarrow^* e_1' \\ e_2 \longrightarrow^* e_2' \end{array}}{(e_1, e_2) \longrightarrow^* (e_1', e_2')} \text{ comp.} \longrightarrow^* \Downarrow$$
9. $$\gg\Uparrow \frac{e \gg e'}{\pi_1\ e \gg \pi_1\ e'} \overset{IH}{\Longrightarrow} \frac{e \longrightarrow^* e'}{\pi_1\ e \longrightarrow^* \pi_1\ e'} \text{ comp.} \longrightarrow^* \Downarrow$$
10. $$\gg\Uparrow \frac{e \gg e'}{\pi_2\ e \gg \pi_2\ e'} \overset{IH}{\Longrightarrow} \frac{e \longrightarrow^* e'}{\pi_2; e \longrightarrow^* \pi_2; e'} \text{ comp.} \longrightarrow^* \Downarrow$$
11. $$\Uparrow\gg \frac{v_1 \gg v_1'}{\pi_1\ (v_1, v_2) \gg v_1'} \overset{IH}{\Longrightarrow} \frac{v_1 \longrightarrow^* v_1'}{\pi_1\ (v_1, v_2) \longrightarrow_{\beta_{\pi_1}} v_1 \longrightarrow^* v_1'} \text{ comp.} \longrightarrow^* \Downarrow$$
12. $$\Uparrow\gg \frac{v_2 \gg v_2'}{\pi_2\ (v_1, v_2) \gg v_2'} \overset{IH}{\Longrightarrow} \frac{v_2 \longrightarrow^* v_2'}{\pi_2\ (v_1, v_2) \longrightarrow_{\beta_{\pi_2}} v_2 \longrightarrow^* v_2'} \text{ comp.} \longrightarrow^* \Downarrow$$
13. $$\gg\Uparrow \frac{e \gg e'}{f\ e \gg f\ e'} \overset{IH}{\Longrightarrow} \frac{e \longrightarrow^* e'}{f\ e \longrightarrow^* f\ e'} \text{ comp.} \longrightarrow^* \Downarrow$$
14. $$\gg\Uparrow \frac{e_f \gg e_f'}{\lambda^{f \in F}(f\ x_f).e_f \gg \lambda^{f \in F}(f\ x_f).e_f'} \overset{IH}{\Longrightarrow} \frac{e_f \longrightarrow^* e_f'}{\lambda^{f \in F}(f\ x_f).e_f \longrightarrow^* \lambda^{f \in F}(f\ x_f).e_f'} \text{ comp.} \longrightarrow^* \Downarrow$$
15. $\#z \gg \#z$ and, $\#z \longrightarrow^0 \#z$

16. $\gg\Uparrow \dfrac{e \gg e'}{\#z \Rightarrow e \gg \#z \Rightarrow e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* e'}{\#z \Rightarrow e \longrightarrow^* \#z \Rightarrow e'}$

17. $\gg\Uparrow \dfrac{e \gg e'}{\lambda(\#z \Rightarrow x).e \gg \lambda(\#z \Rightarrow x).e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* e'}{\lambda(\#z \Rightarrow x).e \longrightarrow^* \lambda(\#z \Rightarrow x).e'}$ comp. $\dashrightarrow^*\Downarrow$

18. $\gg\Uparrow \dfrac{\begin{array}{c} e_1 \gg e_1' \\ e_2 \gg e_2' \end{array}}{e_1 =_\# e_2 \gg e_1' =_\# e_2'} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e_1 \longrightarrow^* e_1' \\ e_2 \longrightarrow^* e_2' \end{array}}{e_1 =_\# e_2 \longrightarrow^* e_1' =_\# e_2'}$ comp. $\dashrightarrow^*\Downarrow$

19. $\gg\Uparrow \dfrac{\#z = \#z'}{\#z = \#z' \gg \mathsf{False}()}$ and $\#z = \#z' \longrightarrow_\# \mathsf{False}()$

20. $\dfrac{\#z = \#z'}{\#z = \#z' \gg \mathsf{True}()}$ and$\#z = \#z' \longrightarrow_\# \mathsf{True}()$

21. $\gg\Uparrow \dfrac{e \gg e'}{\mathsf{isOVar}\ e \gg \mathsf{isOVar} e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* e;}{\mathsf{isOVar}\ e \longrightarrow^* \mathsf{isOVar}\ e'}$ comp. $\dashrightarrow^*\Downarrow$

22. $\dfrac{}{\mathsf{isOVar}\ \#z \gg \mathsf{True}()}$ and $\mathsf{isOVar}\ \#z \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{True}()$.

23. $\dfrac{v \neq \#z}{\mathsf{isOVar}\ v \gg \mathsf{False}()}$ and $\mathsf{isOVar}\ v \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{False}()$

□

Proof (Property 6 of Lemma 14). The property is stated as follows:

$$\forall e_1, e_2, e_3, e_4 \in \mathbb{E}.\ e_1 \gg e_3,\ e_2 \gg e_4 \Longrightarrow e_1[y := e_2] \gg e_3[y := e_4]$$

The substitution property for parallel reduction without complexity follows directly from the substitution property for parallel reduction with complexity (Lemma 18 on page 14).

Remark 16 *From Lemma 14, parts 4 and 5 above we can see that that* $\gg^* = \longrightarrow^*$.

Proof (Remark 16). By induction on the derivations of $\longrightarrow^*$ and $\gg^*$, and has two parts.

- $e_1 \longrightarrow^* e_2 \Longrightarrow e_1 \gg^* e_2$

$$\gg^*\Uparrow \dfrac{\begin{array}{c} e_1 \longrightarrow u \overset{14.4}{\Longrightarrow} e_1 \gg u \\ u \longrightarrow^* e_2 \overset{IH}{\Longrightarrow} u \gg^* e_2 \end{array}}{e_1 \longrightarrow^* e_2} \gg^*\Downarrow$$

- $e_1 \gg^* e_2 \Longrightarrow e_1 \longrightarrow^* e_2$
  Assuming that $e_1 \gg^* e_2$, it must be the case, by definition of $\gg^*$, that there exists some u_1, such that $e_1 \gg u_1$ and $u_1 \gg^* e_2$. By previous property 14.5, then $e_1 \longrightarrow^* u_1$. But for that to be true, there must exist some u' such that $e_1 \longrightarrow u'$ and $u' \longrightarrow^* u_1$.
  To show that $e_1 \longrightarrow^* e_2$, there must exist some u, such that $e_1 \longrightarrow u$ and $u \longrightarrow^* e_2$. Let u' be this u. Now, we know that $e_1 \longrightarrow u$ and $u \longrightarrow^* u_1$. Since $u_1 \gg^* e_2$, then $u_1 \longrightarrow^* e_2$ by the induction hypothesis. Since $u \longrightarrow^* u_1$ and $u_1 \longrightarrow^* e_2$, by transitivity of $\longrightarrow^*$ we have that $u \longrightarrow^* e_2$. Therefore $u \longrightarrow^* e_2$ and we are done.□

Remark 17 (Substitution with Complexity) *We have already shown that parallel reduction without complexity is equivalent (in many steps) to normal reduction (in many steps). The same result applies to parallel reduction with complexity.*

3.3 Substitution Lemma for Parallel Reduction with Complexity

Lemma 18 (Substitution for Parallel Reduction with Complexity) $\forall e_4, e_5, e_6, e_7 \in \mathbb{E},\ X, Y \in \mathbb{N}.$

$$e_4 \overset{X}{\gg} e_5 \wedge e_6^0 \overset{Y}{\gg} e_7 \Longrightarrow (\exists Z \in \mathbb{N}.\, e_4[x := e_6^0] \overset{Z}{\gg} e_5[x := e_7] \wedge Z \leq X + \#(x, e_5)Y).$$

Proof (Lemma 18). Proof is by induction on the derivation $e_4 \overset{Y}{\gg} e_5$. Assumption: $e_6 \overset{Y}{\gg} e_7$.

1. $\dfrac{}{() \overset{0}{\gg} ()}$ and $\exists Z = 0.()[y := e_6] \overset{0}{\gg} ()[y := e_7] \wedge Z \leq 0.$

2. $\dfrac{}{x \overset{0}{\gg} x}$

   (a) If $x = y$ then $\exists Z = Y.x[y := e_6] \overset{Z}{\gg} x[y := e_7] \wedge Z \leq Y$

   (b) If $x \neq y$ then $\exists Z = 0.x[y := e_6] \overset{Z}{\gg} x[y := e_7] \wedge Z \leq 0$

3. $\dfrac{e_1 \overset{N}{\gg} e_2}{\lambda x.e_1 \overset{N}{\gg} \lambda x.e_2}$ By the induction hypothesis, $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_2[y := e_7] \wedge Z \leq N + \#(y, e_2)Y$ By Barendregt's assumption $x \notin FV(e_6, e_7)$, so $\exists Z = Z_1.\lambda x.e_1[x := x][y := e_6] \overset{Z}{\gg} \lambda x.e_2[x := x][y := e_7] \wedge Z \leq N + \#(y, \lambda x.e_2)Y$

4. $\dfrac{e_1 \overset{M}{\gg} e_3 \quad e_2 \overset{N}{\gg} e_4}{e_1\ e_2 \overset{M+N}{\gg} e_3\ e_4}$. By the induction hypothesis, we obtain the following

   (a) $\exists Z_1.\, e_1[y := e_6] \overset{Z_1}{\gg} e_3[y := e_7] \wedge Z_1 \leq M + \#(y, e_3)$

   (b) $\exists Z_2.\, e_2[y := e_6] \overset{Z_2}{\gg} e_4[y := e_7] \wedge Z_2 \leq N + \#(y, e_4)$

   Then, $\exists Z = Z_1 + Z_2.(e_1\ e_2)[y := e_6] \overset{Z}{\gg} (e_3\ e_4)[Y := e_7] \wedge Z \leq (M + N) + \#(y, e_3\ e_4)Y$

5. $\dfrac{e_1 \overset{M}{\gg} e_3 \quad v_1 \overset{N}{\gg} v_2}{(\lambda x.e_1)\, v_1 \overset{M+\#(x,e_3)N+1}{\gg} e_3[x := v_2]}$ By the induction hypothesis, we obtain:

   (a) $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_3[y := e_7] \wedge Z_1 \leq M + \#(y, e_3)Y$

   (b) $\exists Z_2.v_1[y := e_6] \overset{Z_2}{\gg} v_2[y := e_7] \wedge Z_2 \leq N + \#(y, v_2)Y$

   Using the Barendregt's assumption and definition of substitution, the goal can be stated as follows: $\exists Z.$

   $$(\lambda x.e_1[y := e_6])\ (v_1[y := e_6]) \overset{Z}{\gg} e_2[x := v_2][y := e_7]$$

   By property of substitution (Lemma 12) that is equivalent to: $\exists Z.$

   $$(\lambda x.e_1[y := e_6])\ (v[y := e_6]) \overset{Z}{\gg} e_2[y := e_7][x := v'[y := e_7]]$$

   By Barendregt's assumption x is not free in any therms other than e_1 and e_2, and inparticular in v'. From this we can simplify the goal further, and by a series of arithmetical manipulations obtain: $\exists Z = Z_1 + \#(x, e')Z_2 + 1.\ ((\lambda x.e_1)\ v)[y := e_6] \overset{Z}{\gg} e_2[x := v_2][y := e_7] \wedge Z \leq M + \#(x, e_2)N + 1 + \#(y, e'[x := v'])Y.$

6. $\dfrac{e_1 \overset{M}{\gg} e_2}{(\lambda(\#z \Rightarrow x).e_1)\ (\#z \Rightarrow b) \overset{M+1}{\gg} e_2[x := \lambda x.b[\#z := x]]}$

   By the induction hypothess: $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_2[y := e_7] \wedge Z_1 \leq M + \#(y, e_2)Y$. The goal is: $\exists Z.$

   $$(((\lambda \#z \Rightarrow x).e_1)\ (\#z \Rightarrow b)) \overset{Z}{\gg} e_2[x := \lambda x'.b[\#z := x']][y := e_7] \wedge Z \leq M + 1 + \#(y, e_2[x := \lambda x'.b[\#z := x']])$$

   Since $FV(b) = \emptyset$, this can be simplified to

   $$(((\lambda \#z \Rightarrow x).e_1)\ (\#z \Rightarrow b)) \overset{Z}{\gg} e_2[x := \lambda x'.b[\#z := x']][y := e_7] \wedge Z \leq M + 1 + \#(y, e_2)$$

   Then further, by "permutation" of subsitution:

   $$(((\lambda \#z \Rightarrow x).e_1)\ (\#z \Rightarrow b)) \overset{Z}{\gg} e_2[y := e_7][x := \lambda x'.b[\#z := x']] \wedge Z \leq M + 1 + \#(y, e_2)$$

   Now the goal follows easily when $Z = Z_1 + 1$.

7. $\dfrac{e_k \overset{M}{\gg} e'_k \quad v_1 \overset{N}{\gg} v_2}{(\lambda^{f\in F\cup\{k\}} f\ x_f.e_f)\ (f_k\ v_1) \overset{M+\#(x_k,e'_k)N+1}{\gg} e'_k[x_k := v_2]}$ By the induction hypothesis we have

   (a) $\exists Z_1.\, e_k[y := e_6] \overset{Z_1}{\gg} e'_k[y := e_7] \wedge Z_1 \leq M + \#(y, e'_k)Y$

   (b) $\exists Z_2.\, v_1[y := e_6] \overset{Z_2}{\gg} v_2[y := e_7] \wedge Z_1 \leq N + \#(y, v_2)Y$

   Using Barendregt's assumption and the definition of substitution, the goal can be restated as follows: $\exists Z.$

   $$((\lambda^{i\in F\cup\{k\}} f\ x_f.e_f)\ (f_k\ v_1))[y := e_6] \overset{Z}{\gg} e'_k[x := v_2][y := e_7] \wedge Z \leq (M + \#(x, e'_k)N + 1) + \#(y, e'_k[x := v_2])Y$$

   By property of subsitution (Lemma 12), we obtain $\exists Z.$

   $$((\lambda^{i\in F\cup\{k\}} f\ x_f.e_f)\ (f_k\ v_1))[y := e_6] \overset{Z}{\gg} e'_k[y := e_7][x := v_2[y := e_7]] \wedge Z \leq (M + \#(x, e'_k)N + 1) + \#(y, e'_k[x := v_2])Y$$

   By the induction hypothesis, properties of substitution and arithmetic the above goal follows when $Z = Z_1 + \#(x, e'_k)Z_2 + 1$.

8. $\dfrac{e_1 \overset{M}{\gg} e_3 \quad e_2 \overset{N}{\gg} e_4}{(e_1, e_2) \overset{M+N}{\gg} (e_2, e_4)}$ By the induction hypothesis, we obtain

   (a) $\exists Z_1.\, e_1[y := e_6] \overset{Z_1}{\gg} e_3[y := e_7] \wedge Z_1 \leq M + \#(y, e_3)Y$

   (b) $\exists Z_2.\, e_2[y := e_6] \overset{Z_2}{\gg} e_4[y := e_7] \wedge Z_2 \leq N + \#(y, e_4)Y$

   Then, it follows $\exists Z = Z_1 + Z_2.\, (e_1,\ e_2)[y := e_7] \overset{Z}{\gg} (e_3\ e_4)[y := e_7] \wedge Z \leq (M + N) + \#(y, (e_3, e_4))Y$

9. $\dfrac{e_1 \overset{M}{\gg} e_2}{\pi_1\ e_1 \overset{M}{\gg} \pi_1\ e_2}$ By the induction hypothesis we obtain

   $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_2[y := e_7] \wedge Z_1 \leq M + \#(y, e_2)Y$. Then, $\exists Z = Z_1.(\pi_1\ e_1)[y := e_6] \overset{Z}{\gg} (\pi_1\ e_2)[y := e_7] \wedge Z \leq M + \#(y, e_2)Y$.

10. For the derivation $\dfrac{e_1 \overset{M}{\gg} e_2}{\pi_2\ e_1 \overset{M}{\gg} \pi_2\ e_2}$ By the induction hypothesis we obtain

    $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_2[y := e_7] \wedge Z_1 \leq M + \#(y, e_2)Y$. Then, $\exists Z = Z_1.(\pi_2\ e_1)[y := e_6] \overset{Z}{\gg} (\pi_2\ e_2)[y := e_7] \wedge Z \leq M + \#(y, e_2)Y$.

11. $\dfrac{v_1 \overset{N}{\gg} v'_1}{\pi_1\ (v_1, v_2) \overset{N+1}{\gg} v'_1}$

    By the induction hypothesis we obtain: $\exists Z_1.\, v_1[y := e_6] \overset{Z_1}{\gg} v'_1[y := e_7] \wedge Z_1 \leq N + \#(y, v'_1)Y$.

    Then $\exists Z = Z_1 + 1.\, (\pi_1\ (v_1, v_2))[y := e_6] \overset{Z}{\gg} v'_1[y := e_7] \wedge Z \leq N + 1 + \#(y,' v_1)Y$.

12. $\dfrac{v_2 \overset{N}{\gg} v'_2}{\pi_2\ (v_1, v_2) \overset{N+1}{\gg} v'_2}$

    By the induction hypothesis we obtain: $\exists Z_1.\, v_2[y := e_6] \overset{Z_1}{\gg} v'_2[y := e_7] \wedge Z_1 \leq N + \#(y, v'_2)Y$.

    Then $\exists Z = Z_1 + 1.\, (\pi_1\ (v_1, v_2))[y := e_6] \overset{Z}{\gg} v'_2[y := e_7] \wedge Z \leq N + 1 + \#(y,' v_2)Y$.

13. $\dfrac{e_1 \overset{M}{\gg} e_2}{f\ e_1 \overset{M}{\gg} f\ e_2}$:

    By the induction hypothesis $\exists Z_1.e_1[y := e_6] \overset{Z_1}{\gg} e_2[y := e_7] \wedge Z_1 \leq M + \#(y, e_2)Y$.

    Then $\exists Z = Z_1.(f\ e_1)[y := e_6] \overset{Z}{\gg} (f\ e_2)[y := e_7] \wedge Z \leq M + \#(y, f\ e_2)Y$.

14. $\dfrac{e_f \overset{N_f}{\gg} e'_f}{\lambda^{f\in F} f\ x_f.e_f \overset{N}{\gg} \lambda^{f\in F} f\ x_f.e'_f}$

By the induction hypothesis $\forall f \in F$.

$$\exists Z_f . e_f[y{:=}e_6] \overset{Z_f}{\gg} e'_f[y{:=}e_7] \wedge Z_f \leq N_f + \#(y, e'_f)Y$$

Then $\exists Z = \sum_{f \in F} Z_f . (\lambda^{f \in F}\ f\ x_f . e_f[y{:=}e_6]) \overset{Z}{\gg} e'_f[y{:=}e_7] \wedge Z \leq \sum N_f + \sum \#(y, e'_f)Y$.

15. $\dfrac{}{\#z \overset{0}{\gg} \#z}$ and $\exists Z = 0 . \#z[y{:=}e_6] \overset{0}{\gg} \#z[y{:=}e_7] \wedge Z \leq 0 + 0 \cdot Y$

16. $\dfrac{e_1 \overset{M}{\gg} e_2}{(\#z \Rightarrow e_1) \overset{M}{\gg} (\#z \Rightarrow e_2)}$ By the induction hypothesis, $\exists Z_1 . e_1[y{:=}e_6] \overset{Z_1}{\gg} e_2[y{:=}e_7]\ \wedge\ M + \#(y, e_2)Y$. Then,
$\exists Z = Z_1 . (\#z \Rightarrow e_1)[y{:=}e_6] \overset{Z}{\gg} (\#z \Rightarrow e_2)[y{:=}e_7]\ \wedge\ Z \leq M + \#(y, \#z \Rightarrow e_2)$.

17. $\dfrac{e_1 \overset{N}{\gg} e_2}{\lambda(\#z \Rightarrow x).e_1 \overset{N}{\gg} \lambda(\#z \Rightarrow x).e_2}$ By the induction hypothesis, we obtain $\exists Z_1 . e_1[y{:=}e_6] \overset{Z_1}{\gg} e_2[y{:=}e_7] \wedge Z_1 \leq N + \#(y, e_2)$.
Then $\exists Z = Z_1 . (\#z \Rightarrow e_1)[y{:=}e_6] \overset{Z}{\gg} (\#z \Rightarrow e_2)[y{:=}e_7] \wedge Z \leq N + \#(y, \#z \Rightarrow e_2)Y$

18. $\dfrac{}{\#z =_{\#} \#z \overset{1}{\gg} \mathsf{True}()}$, $\exists Z = 1 . (\#z =_{\#} z)[y{:=}e_6] \overset{Z}{\gg} \mathsf{True}()[y{:=}e_7] \wedge Z \leq 1 + \#(y, \mathsf{True}())$

19. $\dfrac{\#z \neq \#z'}{\#z =_{\#} \#z' \overset{1}{\gg} \mathsf{False}()}$, $\exists Z = 1 . (\#z =_{\#} z')[y{:=}e_6] \overset{Z}{\gg} \mathsf{False}()[y{:=}e_7] \wedge Z \leq 1 + \#(y, \mathsf{False}())$

20. $\dfrac{e_1 \overset{M}{\gg} e_3 \quad e_2 \overset{N}{\gg} e_4}{e_1 =_{\#} e_2 \overset{M+N}{\gg} e_3 =_{\#} e_4}$
By the induction hypothesis we obtain
(a) $\exists Z_1 . e_1[y{:=}e_6] \overset{Z_1}{\gg} e_3[y{:=}e_7] \wedge Z_1 \leq M + \#(y, e_3)Y$
(b) $\exists Z_2 . e_2[y{:=}e_6] \overset{Z_2}{\gg} e_4[y{:=}e_7] \wedge Z_2 \leq N + \#(y, e_4)Y$.
Then $\exists Z = Z_1 + Z_2 . (e_1 =_{\#} e_2)[y{:=}e_6] \overset{Z}{\gg} (e_3 =_{\#} e_4)[y{:=}e_7] \wedge Z \leq (M + N) + \#(y, (e_e =_{\#} e_4))Y$.

21. $\dfrac{e \overset{X}{\gg} e'}{\mathsf{isOVar}\ e \overset{X}{\gg} \mathsf{isOVar}\ e'}$ By the induction hypothesis, $\exists Z_1 . e[y{:=}e_6] \overset{Z_1}{\gg} e'[y{:=}e_6]\ \wedge\ Z_1 \leq X + \#(y, e')Y$. Then,
$\exists Z = Z_1 . (\mathsf{isOVar}\ e)[y{:=}e_6] \overset{Z}{\gg} (\mathsf{isOVar}\ e')[y{:=}e_7]\ \wedge\ Z \leq X + \#(y, \mathsf{isOVar}\ e')$.

22. $\dfrac{}{\mathsf{isOVar}\ \#z \overset{1}{\gg} \mathsf{True}()}$ and $\exists Z = 1 . (\mathsf{isOVar}\ \#z)[y{:=}e_6] \overset{Z}{\gg} \mathsf{True}()[y{:=}e_7]\ \wedge\ Z \leq 1 + \#(y, \mathsf{True}())Y$.

23. $\dfrac{v \neq \#z}{\mathsf{isOVar}\ v \overset{1}{\gg} \mathsf{False}()}$ Then, $\exists Z = 1 . (\mathsf{isOVar}\ v)[y{:=}e_6] \overset{Z}{\gg} \mathsf{False}()[\mathsf{y{:=}e_7}]\ \wedge\ Z \leq 1 + \#(y, \mathsf{False}())Y$.

3.4 Big-Step Semantics

Lemma 19 (Basic Property of Big-Step Semantics) *If $e \hookrightarrow e'$ then $e' \in \mathbb{V}$.*

Proof. Proof is straightforward by induction over the height of the derivation. □ [e.p.]

3.5 Classes

Lemma 20 (Basic Properties of Classes)

1. $\mathbb{V}, \mathbb{W}, \mathbb{S} \subseteq \mathbb{E}$
2. $\mathbb{V}, \mathbb{W}, \mathbb{S}$ *partition* $\mathbb{E}$.

Proof. Proof is by structural induction on e, and then by pattern matching on e. The following table summarizes this rather tedious proof:

$e \in \mathbb{E}$	$\in \mathbb{V}$	$\in \mathbb{W}$	$\in \mathbb{S}$
$()$	Yes	No	No
x	No	No	Yes
$\lambda x.e$	Yes	No	No
$(\lambda x.e)\ v$	No	Yes	No
$(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b)$	No	Yes	No
$(\lambda^{f \in F} f\ x_f.e_f)\ (f\ v)$	No	Yes	No
$w\ e$	No	Yes	No
$v\ w$	No	Yes	No
$(\mathbb{V} \setminus \lambda^e)\ v$	No	No	Yes
$s_1\ e$	No	No	Yes
$v\ s$	No	No	Yes
$(\lambda(\#z \Rightarrow x).e)\ (\mathbb{V} \setminus \#z \Rightarrow b)$	No	No	Yes
$(\lambda^{f \in F} f\ x_f.e_f)\ (\mathbb{V} \setminus (f\ v))$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
(v_1, v_2)	Yes	No	No
(w, e)	No	Yes	No
(v, w)	No	Yes	No
(s, e)	No	No	Yes
(v, s)	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
$\pi_1\ (v_1, v_2)$	No	Yes	No
$\pi_1\ w$	No	Yes	No
$\pi_1\ (\mathbb{V} \setminus (v_1, v_2))$	No	No	Yes
$\pi_1\ s$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			

$e \in \mathbb{E}$	$\in \mathbb{V}$	$\in \mathbb{W}$	$\in \mathbb{S}$
$\pi_2\ (v_1, v_2)$	No	Yes	No
$\pi_2\ w$	No	Yes	No
$\pi_2\ (\mathbb{V} \setminus (v_1, v_2))$	No	No	Yes
$\pi_2\ s$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
$f\ v$	Yes	No	No
$f\ w$	No	Yes	No
$f\ s$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
$\lambda^{f \in F}(f\ x_f).e_f$	Yes	No	No
$\#z$	Yes	No	No
$\#z \Rightarrow v$	Yes	No	No
$\#z \Rightarrow w$	No	Yes	No
$\#z \Rightarrow s$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
$\lambda(\#z \Rightarrow x).e$	Yes	No	No
$\#z =_\# \#z$	No	Yes	No
$w =_\# e$	No	Yes	No
$\#z =_\# w$	No	Yes	No
$(\mathbb{V} \setminus \#z) =_\# e$	No	No	Yes
$\#z =_\# (\mathbb{V} \setminus \#z)$	No	No	Yes
$s =_\# e$	No	No	Yes
$\#z =_\# s$	No	No	Yes
Exhaustive/Nonoverlapping: Yes			
isOVar v	No	Yes	No
isOVar w	No	Yes	No
isOVar s	No	No	Yes
Exhaustive/Nonoverlapping: Yes			

□[e.p.]

3.6 Parallel Reduction and Classes

There is a sense in which parallel reduction should respect the classes. The following lemma explicates these properties.

Lemma 21 (Parallel Reduction and Classes)

1. $\forall e \in \mathbb{E}, v \in \mathbb{V}.\, v \overset{M}{\gg} e \Longrightarrow e \in V$
2. $\forall e \in \mathbb{E}, s \in \mathbb{S}.\, s \overset{M}{\gg} e \Longrightarrow e \in S$
3. $\forall e \in \mathbb{E}, w \in \mathbb{W}.\, e \overset{M}{\gg} w \Longrightarrow e \in W.$

Proof (Lemma 21.1). By structural induction on $v \in \mathbb{V}$.

1. $() \in \mathbb{V}$ and $() \gg () \in \mathbb{V}$
2. $\dfrac{e \gg e'}{\lambda x.e \gg \lambda x.e'}$ and $\lambda x.e' \in \mathbb{V}$.

3. $\gg\Uparrow$ $\dfrac{\begin{array}{c} v_1 \gg e_1 \\ v_2 \gg e_2 \end{array}}{(v_1, v_2) \gg (e_1, e_2)} \begin{array}{c} \overset{IH}{\Longrightarrow} \\ \overset{IH}{\Longrightarrow} \end{array} \dfrac{\begin{array}{c} e_1 \in \mathbb{V} \\ e_2 \in \mathbb{V} \end{array}}{(e_1, e_2) \in \mathbb{V}} \in^{\mathbb{V}}\Downarrow$

4. $\gg\Uparrow$ $\dfrac{v \gg e}{f\ v \gg f\ e} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{V}}{f\ v' \in \mathbb{V}} \in^{\mathbb{V}}\Downarrow$
5. $\lambda^{i \in L}\ f_i\ x_i.e_i \gg \lambda^{i \in L}\ f_i\ x_i.e_i'$ and $\lambda^{i \in L}\ f_i\ x_i.e_i' \in \mathbb{V}$
6. $\#z \in \mathbb{V}$ and $\#z \gg \#z \in \mathbb{V}$
7. $\gg\Uparrow$ $\dfrac{v \gg v'}{\#z \Rightarrow v \gg \#z \Rightarrow v'} \overset{IH}{\Longrightarrow} \dfrac{v' \in \mathbb{V}}{\#z \Rightarrow v' \in \mathbb{V}} \in^{\mathbb{V}}\Downarrow$
8. $\lambda(\#z \Rightarrow x).e' \in \mathbb{V}$ and $\lambda(\#z \Rightarrow x).e \gg \lambda(\#z \Rightarrow x).e' \in \mathbb{V}$

□

Proof (Lemma 21.2).

$$\forall e \in \mathbb{E}, s \in S. s \overset{M}{\gg} e \Longrightarrow e \in S$$

By structural induction on $s \in \mathbb{S}$.

1. $x \in \mathbb{S}$ and $x \gg x$ and $x \in \mathbb{S}$.
2. $\gg\Uparrow$ $\dfrac{\begin{array}{c} e_1 \gg e_2 \\ s_1 \gg e_3 \end{array}}{s_1\ e_1 \gg e_3\ e_2} \begin{array}{c} \overset{\gg}{\Longrightarrow} \\ \overset{IH}{\Longrightarrow} \end{array} \dfrac{\begin{array}{c} e_2 \in \mathbb{E} \\ e_3 \in \mathbb{S} \end{array}}{e_3\ e_2 \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$

3. $\gg\Uparrow$ $\dfrac{\begin{array}{c} v \gg e_1 \\ s \gg e_2 \end{array}}{v\ s \gg e_1\ e_2} \begin{array}{c} \overset{21.1}{\Longrightarrow} \\ \overset{IH}{\Longrightarrow} \end{array} \dfrac{\begin{array}{c} e_1 \in \mathbb{V} \\ e_2 \in \mathbb{S} \end{array}}{e_1\ e_2 \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$

4. $\gg\Uparrow$ $\dfrac{\begin{array}{c} (\lambda(\#z \Rightarrow x).e) \gg (\lambda(\#z \Rightarrow x).e') \\ v \gg v' \quad v \in \mathbb{V} \setminus (\#z \Rightarrow b) \end{array}}{(\lambda(\#z \Rightarrow x).e)\ v \gg (\lambda(\#z \Rightarrow x).e)\ v'} \overset{\gg.21.1}{\Longrightarrow} \dfrac{v' \in \mathbb{V} \setminus (\#z \Rightarrow b)}{(\lambda(\#z \Rightarrow x).e)\ v' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
5. $\gg\Uparrow$ $\dfrac{\begin{array}{c} v \gg v' \\ s \gg s' \end{array}}{(v, s) \gg (v', s')} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{(v', s') \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
6. $\gg\Uparrow$ $\dfrac{\begin{array}{c} e \gg e' \\ s_1 \gg s_1' \end{array}}{(s_1, e) \gg (s_1', e')} \begin{array}{c} \overset{\gg}{\Longrightarrow} \\ \overset{IH}{\Longrightarrow} \end{array} \dfrac{\begin{array}{c} e' \in \mathbb{E} \\ s_1' \in \mathbb{S} \end{array}}{(s_1', e') \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$

7. $\gg\Uparrow$ $\dfrac{s \gg s'}{\pi_n\ s \overset{\pi}{\gg}_n\ s'} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{\pi_n\ s' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
8. $\gg\Uparrow$ $\dfrac{v \gg v' \quad v \in (\mathbb{V} \setminus (v, v))}{\pi_n\ v \gg \pi_n\ v'} \overset{\gg}{\Longrightarrow} \dfrac{v' \in (\mathbb{V} \setminus (v, v))}{\pi_n\ v' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
9. $\gg\Uparrow$ $\dfrac{s \gg s'}{f\ s \gg f\ s'} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{f\ s' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
10. $\gg\Uparrow$ $\dfrac{\begin{array}{c} \lambda^{f \in F}\ f\ x_f.e_f \gg \lambda^{f \in F}\ f\ x_f.e_f' \\ v \gg v' \quad v \in (\mathbb{V} \setminus f\ v) \end{array}}{(\lambda^{f \in F}\ f\ x_f.e_f)\ v \gg (\lambda^{f \in F}\ f\ x_f.e_f')\ v'} \overset{IH}{\Longrightarrow} \dfrac{v' \in (\mathbb{V} \setminus f\ v)}{(\lambda^{f \in F}\ f\ x_f.e_f')\ v' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
11. $\gg\Uparrow$ $\dfrac{s \gg s'}{\#z \Rightarrow s \gg \#z \Rightarrow s'} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{\#z \Rightarrow s' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$
12. $\gg\Uparrow$ $\dfrac{v \gg v' \quad v \in (\mathbb{V} \setminus \mathbb{B})}{\#z \Rightarrow v \gg \#z \Rightarrow v'} \overset{IH}{\Longrightarrow} \dfrac{v' \in (\mathbb{V} \setminus \mathbb{B})}{\#z \Rightarrow v' \in \mathbb{S}} \in^{\mathbb{S}}\Downarrow$

13. $\gg\Uparrow$ $\dfrac{\dfrac{e \gg e'}{v \gg v' \quad v' \in (\mathbb{V} \setminus \#z)} \overset{\gg}{\Longrightarrow} \dfrac{v' \in (\mathbb{V} \setminus \#z)}{v' = e' \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{v = e \gg v' = e'}$

14. $\gg\Uparrow$ $\dfrac{\dfrac{\#z \gg \#z}{v \gg v' \quad v \in (\mathbb{V} \setminus \#z)} \overset{\gg, \in\mathbb{V}}{\Longrightarrow} \dfrac{v' \in (\mathbb{V} \setminus \#z)}{\#z = v' \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{\#z = v \gg \#z = v'}$

15. $\gg\Uparrow$ $\dfrac{\dfrac{e \gg e'}{s \gg s'} \overset{\gg}{\Longrightarrow} e' \in \mathbb{E} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{s' = e' \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{s = e \gg s'_1 = s'_2}$

16. $\gg\Uparrow$ $\dfrac{\dfrac{s \gg s'}{\#z =_\# s \gg \#z =_\# s'} \overset{IH}{\Longrightarrow} \dfrac{s' \in \mathbb{S}}{\#z =_\# s' \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{}$

17. $\gg\Uparrow$ $\dfrac{\dfrac{v_2 \gg e_2 \quad v_2 \in \mathbb{V}}{`v_1 \gg e_1 \quad v_1 \in (\mathbb{V} \setminus \lambda^e)} \overset{\gg, 21.1}{\Longrightarrow} e_2 \in \mathbb{V} \overset{\gg, 21.1}{\Longrightarrow} \dfrac{e_1 \in (\mathbb{V} \setminus \lambda^e)}{e_1\ e_2 \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{v_1\ v_2 \gg e_1\ e_2}$

18. $\gg\Uparrow$ $\dfrac{\dfrac{s \gg e}{\mathsf{isOVar}\ s \gg \mathsf{isOVar}\ e} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{S}}{\mathsf{isOVar}\ e \in \mathbb{S}}\ {}_{\in\mathbb{S}}\Downarrow}{}$

□

Proof (Lemma 21.3).

$$\forall e \in \mathbb{E}, w \in W.\, e \overset{M}{\gg} w \Longrightarrow e \in W.$$

Property 3 follows directly from the previous two properties. □

3.7 Left Reduction

Lemma 22 (Left Reduction and Classes)

1. $\forall w \in W.\, (\exists e' \in \mathbb{E}.\, w \longmapsto e')$
2. $\forall e \in \mathbb{E}.\, (\exists e' \in \mathbb{E}.\, e \longmapsto e') \Longrightarrow e \in W$
3. $\forall v \in \mathbb{V}.\, \neg(\exists e' \in \mathbb{E}.\, v \longmapsto e')$
4. $\forall s \in S.\, \neg(\exists e' \in \mathbb{E}.\, s \longmapsto e')$.

Proof (Lemma 22). We only need to prove the first two, and the second two follow. The first one is by straightforward induction on the judgement $e \in \mathbb{W}$. The second is also by straightforward induction on the derivation $e \longmapsto e'$.

$$\forall w \in W.\, (\exists e' \in \mathbb{E}.\, w \longmapsto e')$$

By structural induction on w.

1. $(\lambda x.e)\ v \longmapsto e[x := v]$and $e[x := v] \in \mathbb{E}$
2. $(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \longmapsto e[x := \lambda x'.b[\#z := x']]$ and $e[x := \lambda x'.b[\#z := x']] \in \mathbb{E}$
3. $\longmapsto\Uparrow$ $\dfrac{w \longmapsto w'}{w\ e \longmapsto w'\ e}\ {}_{IH, \in\mathbb{E}}\Downarrow$
4. $\longmapsto\Uparrow$ $\dfrac{\dfrac{w \longmapsto w'}{v\ w \longmapsto v\ w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{v\ w' \in \mathbb{E}}\ {}_{\in\mathbb{E}}\Downarrow}{}$
5. $\longmapsto\Uparrow$ $\dfrac{\dfrac{w \longmapsto w'}{f\ w \longmapsto f\ w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{f\ w' \in \mathbb{E}}\ {}_{\in\mathbb{E}}\Downarrow}{}$
6. $\longmapsto\Uparrow$ $\dfrac{\dfrac{w \longmapsto w'}{\#z \Rightarrow w \longmapsto \#z \Rightarrow w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{\#z \Rightarrow w' \in \mathbb{E}}\ {}_{\in\mathbb{E}}\Downarrow}{}$

7. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{(w,e) \longmapsto (w',e)} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{(w',e) \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$
8. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{(v,w) \longmapsto (v,w')} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{(v,w') \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$
9. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{w = e \longmapsto w' = e} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{w' = e \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$
10. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{\#z = w \longmapsto \#z = w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{\#z = w' \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$
11. $\#z = \#z' \longmapsto \mathsf{True}()\text{and } \mathsf{True} \in \mathbb{E}$
12. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{\pi_n\ w \longmapsto \pi_n\ w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{\pi_n\ w' \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$
13. $\pi_1(v_1,v_2) \longmapsto v_1$ and $v_1 \in \mathbb{E}$
14. $\pi_2(v_1,v_2) \longmapsto v_2$ and $v_2 \in \mathbb{E}$
15. $\mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()$ and $\mathsf{True}() \in \mathbb{E}$
16. $\mathsf{isOVar}\ v \longmapsto \mathsf{False}()$ where $v \neq \#z$, and $\mathsf{False}() \in \mathbb{E}$.
17. $\hookrightarrow\Uparrow \dfrac{w \longmapsto w'}{\mathsf{isOVar}\ w \longmapsto \mathsf{isOVar}\ w'} \overset{IH}{\Longrightarrow} \dfrac{w' \in \mathbb{E}}{\mathsf{isOVar}\ w' \in \mathbb{E}}\ {}_{\in^{\mathbb{E}}\Downarrow}$

Part 2 is proven by induction on the height of derivations of $e \longmapsto e'$, and by case analysis on $e \longmapsto e'$.

$$\forall e \in \mathbb{E}.(\exists e' \in \mathbb{E}.\ e \longmapsto e') \Longrightarrow e \in \mathbb{W}$$

1. $\dfrac{}{(\lambda x.e)\ v \longmapsto e[x:=v]}$ and $(\lambda x.e)\ v \in \mathbb{W}$
2. $\hookrightarrow\Uparrow \dfrac{e_1 \longmapsto e_1'}{e_1\ e_2 \longmapsto e_1'\ e_2} \overset{IH}{\Longrightarrow} \dfrac{e_1 \in \mathbb{W}}{e_1\ e_2 \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
3. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{v\ e \longmapsto v\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{v\ e \in \mathbb{W}}$
4. $\hookrightarrow\Uparrow \dfrac{e_1 \longmapsto e_1'}{(e_1,e_2) \longmapsto (e_1',e_2)} \overset{IH}{\Longrightarrow} \dfrac{e_1 \in \mathbb{W}}{(e_1,e_2) \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
5. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{(v,e) \longmapsto (v,e')} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{(v,e) \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
6. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{\pi_1\ e \longmapsto \pi_1 e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{\pi_1\ e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
7. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{\pi_2\ e \longmapsto \pi_2 e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{\pi_2\ e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
8. $\dfrac{}{\pi_1\ (v_1,v_2) \longmapsto v_1}$ and $\pi_1\ (v_1,v_2) \in \mathbb{W}$
9. $\dfrac{}{\pi_2\ (v_1,v_2) \longmapsto v_2}$ and $\pi_2\ (v_1,v_2) \in \mathbb{W}$
10. $\dfrac{}{(\lambda^{f \in F \cup \{k\}}\ (f\ x_f).e_f)\ (k\ v) \longmapsto e_k[x_k:=v]}$, and $(\lambda^{f \in F \cup \{k\}}\ (f\ x_f).e_f)\ (k\ v) \in \mathbb{W}$
11. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{f\ e \longmapsto f\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{f\ e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
12. $\hookrightarrow\Uparrow \dfrac{e \longmapsto e'}{\#z \Rightarrow e \longmapsto \#z \Rightarrow e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{\#z \Rightarrow e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$
13. $\dfrac{}{(\lambda(\#z \Rightarrow x).e)\ (\#z' \Rightarrow b) \longmapsto e[x:=\lambda x'.b[\#z':=x']]}$ and $(\lambda(\#z \Rightarrow x).e)\ (\#z' \Rightarrow b) \in \mathbb{W}$
14. $\hookrightarrow\Uparrow \dfrac{e_1 \longmapsto e_1'}{e_1 =_{\#} e_2 \longmapsto e_1' =_{\#} e_2} \overset{IH}{\Longrightarrow} \dfrac{e_1 \in \mathbb{W}}{e_1 =_{\#} e_2 \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}\Downarrow}$

15. $\rightarrowtail\Uparrow$ $\dfrac{e \longmapsto e'}{\#z' =_{\#} e \longmapsto \#z' =_{\#} e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{\#z' =_{\#} e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}}\Downarrow$

16. $\dfrac{}{\#z =_{\#} z \longmapsto \mathsf{True}()}$ and $\#z =_{\#} \#z \in \mathbb{W}$

17. $\dfrac{\#z \neq \#z'}{\#z =_{\#} z' \longmapsto \mathsf{False}()}$ and $\#z =_{\#} \#z \in \mathbb{W}$

18. $\dfrac{}{\mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()}$ and $\mathsf{isOVar}\ \#z \in \mathbb{W}$.

19. $\dfrac{v \neq \#z}{\mathsf{isOVar}\ v \longmapsto \mathsf{False}()}$ and $\mathsf{isOVar}\ v \in \mathbb{W}$.

20. $\rightarrowtail\Uparrow$ $\dfrac{e \longmapsto e'}{\mathsf{isOVar}\ e \longmapsto \mathsf{isOVar}\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \in \mathbb{W}}{\mathsf{isOVar}\ e \in \mathbb{W}}\ {}_{\in^{\mathbb{W}}}\Downarrow$

□[e.p]

Remark 23 (Left Reduction Determinism) *From the above it easily follows that for any expression $e \in \mathbb{E}$, if $e \longmapsto e'$, then there is only one derivation by which $e \longmapsto e'$.*□

4 General Part of Confluence

The Church-Rosser theorem [1] for $\longrightarrow$ follows from Takahashi's property [9] (Theorem 27). The statement of Takahashi's property uses the notion of a *complete development.*

4.1 Parallel Reduction is Diamond

Lemma 24 (Parallel Reduction is Diamond) $\forall e_1, e, e_2 \in \mathbb{E}$.

$$e_1 \ll e \gg e_2 \Longrightarrow (\exists e' \in \mathbb{E}.\, e_1 \gg e' \ll e_2).$$

4.2 Takahashi's Property

Proof. Take $e' = !e$ and use Takahashi's property(Theorem 27). □

4.3 Main Confluence Result

Theorem 25 (Main Confluence Result) .

$$\forall e_1, e, e_2 \in \mathbb{E}. \quad e_1 \longleftarrow^* e \longrightarrow^* e_2 \Longrightarrow \exists e' \in \mathbb{E}. \quad e_1 \longrightarrow^* e' \longleftarrow^* e_2$$

Proof (Theorem 25). Follows directly from Lemma 24. □

5 Special Part of Confluence

Remark 26 *By a simple induction on e, we can see that $e \gg !e$.*

5.1 Takahashi's Property

Theorem 27 (Takahashi's Property) $\forall e_1, e_2 \in \mathbb{E}$.

$$e_1 \gg e_2 \Longrightarrow e_2 \gg !e_1.$$

Proof (Takahashi's property for $\gg$).

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \gg e_2 \Longrightarrow e_2 \gg !e_2$$

By induction on the height of the derivation $e_1 \gg e_2$.

1. $() \gg ()$ and $() \gg !()$.
2. For $x \gg x$ and $x \gg !x$
3. $\gg\Uparrow \dfrac{e \gg e'}{\lambda x.e \gg \lambda x.e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{(\lambda x.e') \gg !(\lambda x.e)}{}_{!,\gg}\Downarrow$
4. $\gg\Uparrow \dfrac{\begin{array}{c} v \gg v' \\ e \gg e' \end{array}}{(\lambda x.e)\ v \gg e'[x := v']} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} v' \gg !v \\ e' \gg !e \end{array}}{e'[x := v'] \gg !((\lambda x.e)\, v)}\ 14.6_{,!,\gg}\Downarrow$
5. $$\gg\Uparrow \frac{e \gg e'}{(\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b) \gg e'[x := \lambda x'.b[\#z := x']]} \overset{IH}{\Longrightarrow} \frac{e' \gg !e}{e'[x := \lambda x'.b[\#z := x']] \gg !((\lambda(\#z \Rightarrow x).e)\ (\#z \Rightarrow b))}{}_{\gg,!,14.6}\Downarrow$$
6. $$\gg\Uparrow \frac{\begin{array}{c} v \gg v' \\ e_k \gg e'_k \end{array}}{(\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f)\ (k\ v) \gg e'_k[x := v']} \overset{IH}{\Longrightarrow} \frac{\begin{array}{c} v' \gg !v \\ e'_k \gg !e_k \end{array}}{e'_k[x := v'] \gg !((\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f)\ (k\ v))}{}_{!,\gg,14.6}\Downarrow$$
7. $\gg\Uparrow \dfrac{\begin{array}{c} e_2 \gg e'_2 \\ e_1 \gg e'_1 \end{array}}{e_1\ e_2 \gg e'_1\ e'_2} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e'_2 \gg !e_1 \\ e'_1 \gg !e_1 \end{array}}{(e'_1\ e'_2) \gg !(e_1\ e_2)}{}_{\gg,!}\Downarrow$
8. $\gg\Uparrow \dfrac{\begin{array}{c} e_2 \gg e'_2 \\ e_1 \gg e'_1 \end{array}}{(e_1, e_2) \gg (e'_1, e'_2)} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e'_2 \gg !e_2 \\ e'_1 \gg !e_1 \end{array}}{(e'_1, e'_2) \gg !(e_1, e_2)}{}_{\gg,!}\Downarrow$
9. $\gg\Uparrow \dfrac{e \gg e'}{\pi_1\ e \gg \pi_1\ e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{(\pi_1 e') \gg !(\pi_1 e)}{}_{\gg,!}\Downarrow$
10. $\gg\Uparrow \dfrac{e \gg e'}{\pi_2\ e \gg \pi_2\ e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{(\pi_2 e') \gg !(\pi_2 e)}{}_{\gg,!}\Downarrow$
11. $\gg\Uparrow \dfrac{v_1 \gg v'_1}{\pi_1\ (v_1, v_2) \gg v'_1} \overset{IH}{\Longrightarrow} \dfrac{v'_1 \gg !v_1}{v'_1 \gg !(\pi_1\ (v_1, v_2))}{}_{\gg,!}\Downarrow$
12. $\gg\Uparrow \dfrac{v_2 \gg v'_2}{\pi_2\ (v_1, v_2) \gg v'_2} \overset{IH}{\Longrightarrow} \dfrac{v'_2 \gg !v_2}{v'_2 \gg !(\pi_2\ (v_1, v_2))}{}_{\gg,!}\Downarrow$
13. $\gg\Uparrow \dfrac{e \gg e'}{f\ e \gg f\ e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{f\ e' \gg !(f\ e)}{}_{\gg,!}\Downarrow$
14. $\gg\Uparrow \dfrac{e_f \gg e'_f}{\lambda^{f \in F}\ f\ x_f.e_f \gg \lambda^{f \in F}\ f\ x_f.e'_f} \overset{IH}{\Longrightarrow} \dfrac{e'_f \gg !e_f}{\lambda^{f \in F}\ f\ x_f.e'_f \gg !(\lambda^{f \in F}\ f\ x_f.e_f)}{}_{\gg,!}\Downarrow$
15. $\#z \gg \#z$ and $\#z \gg !\#z$
16. $\gg\Uparrow \dfrac{e \gg e'}{\#z \Rightarrow e \gg \#z \Rightarrow e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{\#z \Rightarrow e' \gg !(\#z \Rightarrow e)}{}_{\gg,!}\Downarrow$
17. $\gg\Uparrow \dfrac{e \gg e'}{\lambda(\#z \Rightarrow x).e \gg \lambda(\#z \Rightarrow x).e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg !e}{\lambda(\#z \Rightarrow x).e' \gg !(\lambda(\#z \Rightarrow x).e)}{}_{\gg,!}\Downarrow$
18. $\gg\Uparrow \dfrac{e_1 \gg e'_1 \quad e_2 \gg e'_2}{e_1 = e_2 \gg e'_1 = e'_2} \overset{IH}{\Longrightarrow} \dfrac{e'_1 \gg !e_1 \quad e'_2 \gg !e_2}{(e'_1 = e'_2) \gg !(e_1 = e_2)}{}_{\gg,!}\Downarrow$
19. $\#z = \#z \gg \mathsf{True}()$ and $\mathsf{True}() \gg !(\#z = \#z)$
20. $\#z \neq \#z' \quad \#z =_{\#} z\#z' \gg \mathsf{False}()$ and $\mathsf{False}() \gg !(\#z =_{\#} \#z')$

21. $\mathsf{isOVar}\ \#z \gg \mathsf{True}$ and $\mathsf{True}() \gg !(\mathsf{isOVar}\ \#z)$.

22. $\dfrac{v \neq \#z}{\mathsf{isOVar}\ v \gg \mathsf{False}()}$, and $\mathsf{False}() \gg !(\mathsf{isOVar}\ v)$.

23. $\gg\Uparrow \dfrac{e \gg e'}{\mathsf{isOVar}\ e \gg \mathsf{isOVar}\ e'} \overset{IH}{\Longrightarrow} \dfrac{e' \gg ! e}{\mathsf{isOVar}\ e' \gg !(\mathsf{isOVar}\ e)} \gg_{!}\Downarrow$

□[e.p.]

6 General Part of Computational Adequacy of Reduction Semantics

6.1 Main Soundness Theorem

Definition 28 (Observational Equivalence)

$$e_1 \approx e_2 \equiv \forall C \in \mathbb{C}.\, C[e_1]\Downarrow \Leftrightarrow C[e_2]\Downarrow$$

Definition 29 (Termination)

$$e_1\Downarrow \equiv \exists v \in \mathbb{V}.\, e_1 \hookrightarrow v$$

Theorem 30 (Soundness Theorem)

$$\forall e_1, e_2 \in \mathbb{E}.\ e_1 \longrightarrow e_2 \Longrightarrow e_1 \approx e_2$$

Proof (Theorem 30). By the definition of $\approx$, to prove our goal

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow e_1 \approx e_2$$

is to prove

$$C \in \mathbb{C}. \forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \wedge C[e_1], C[e_2] \in \mathbb{E} \Longrightarrow (C[e_1]\Downarrow \iff C[e_2]\Downarrow).$$

Noting that by the compatibility of $\longrightarrow$, we know that $C \in \mathbb{C}. \forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow C[e_1] \longrightarrow C[e_2]$, it is sufficient to prove a stronger statement:

$$C \in \mathbb{C}. \forall e_1, e_2 \in \mathbb{E}. C[e_1] \longrightarrow C[e_2] \wedge C[e_1], C[e_2] \in \mathbb{E} \Longrightarrow (C[e_1]\Downarrow \iff C[e_2]\Downarrow).$$

Noting further that $C \in \mathbb{C}. \forall a, b \in \mathbb{E}. a \equiv C[b] \in \mathbb{E} \Longrightarrow a \in \mathbb{E}$, it is sufficient to prove an even stronger statement:

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow (e_1\Downarrow \iff e_2\Downarrow).$$

This goal can be broken down into two parts:

S1

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow (e_1\Downarrow \Longrightarrow e_2\Downarrow),$$

and

S2

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow (e_2\Downarrow \Longrightarrow e_1\Downarrow).$$

Let us consider S1. By definition of termination, it says:

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}. e_1 \hookrightarrow v) \Longrightarrow (\exists v \in \mathbb{V}. e_2 \hookrightarrow v)).$$

We will show that big-step evaluation is included in reduction (Lemma 34). Thus, to prove S2 it is enough to prove:

$$\forall e_1, e_2 \in \mathbb{E}. e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}. e_1 \longrightarrow^* v) \Longrightarrow (\exists v \in \mathbb{V}. e_2 \hookrightarrow v)).$$

Confluence (Theorem 25) tell us that any two reduction paths are joinable, so we can weaken our goal as follows:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}, e_3 \in \mathbb{E}.\, e_1 \longrightarrow^* v \longrightarrow^* e_3 \wedge e_2 \longrightarrow^* e_3) \Longrightarrow (\exists v \in \mathbb{V}.\, e_2 \hookrightarrow v))$$

We will show (Lemmas 21 and Remark 16) that any reduction that starts from a value can only lead to a value (at the same level). Thus we can weaken further:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v, v_3 \in \mathbb{V}.\, e_1 \longrightarrow^* v \longrightarrow^* v_3 \wedge e_2 \longrightarrow^* v_3) \Longrightarrow (\exists v \in \mathbb{V}.\, e_2 \hookrightarrow v))$$

In other words, we already know that e_2 reduces to a value, and the question is really whether it *evaluates* to a value. Formally:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v_3 \in \mathbb{V}.\, e_2 \longrightarrow^* v_3) \Longrightarrow (\exists v \in \mathbb{V}.\, e_2 \hookrightarrow v)).$$

In fact, the original assumption is no longer necessary, and we will prove:

T1

$$\forall e_2 \in \mathbb{E}.\, ((\exists v_3 \in \mathbb{V}.\, e_2 \longrightarrow^* v_3) \Longrightarrow (\exists v \in \mathbb{V}.\, e_2 \hookrightarrow v)).$$

Now consider S2. By definition of termination, it says:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}.\, e_2 \hookrightarrow v) \Longrightarrow (\exists v \in \mathbb{V}.\, e_1 \hookrightarrow v)).$$

again, by the inclusion of evaluation in reduction, we can weaken:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}.\, e_2 \longrightarrow^* v) \Longrightarrow (\exists v \in \mathbb{V}.\, e_1 \hookrightarrow v)).$$

Given the first assumption in this statement we can also say:

$$\forall e_1, e_2 \in \mathbb{E}.\, e_1 \longrightarrow e_2 \Longrightarrow ((\exists v \in \mathbb{V}.\, e_1 \longrightarrow^* v) \Longrightarrow (\exists v \in \mathbb{V}.\, e_1 \hookrightarrow v)),$$

and we no longer need the assumption as it is sufficient to show:

T2

$$\forall e_1 \in \mathbb{E}.\, ((\exists v \in \mathbb{V}.\, e_1 \longrightarrow^* v) \Longrightarrow (\exists v \in \mathbb{V}.\, e_1 \hookrightarrow v)).$$

But note that T1 and T2 are identical goals. They state:

T

$$\forall e \in \mathbb{E}.\, ((\exists v \in \mathbb{V}.\, e \longrightarrow^* v) \Longrightarrow (\exists v \in \mathbb{V}.\, e \hookrightarrow v)).$$

This statement is a direct consequence of Lemma 31. □

It is easy to show that $e \hookrightarrow v \Longrightarrow e \longrightarrow^* v$, as it follows directly from Lemma 34.

6.2 Reduction is in Evaluation

Lemma 31 (Reduction is in Evaluation) $\forall e \in \mathbb{E}, v_1 \in \mathbb{V}.$

$$e \longrightarrow^* v_1 \Longrightarrow (\exists v_3 \in \mathbb{V}.\, e \hookrightarrow v_3 \longrightarrow^* v_1).$$

Proof. We arrive at this result by an adaptation of Plotkin's proof for a similar result for the CBV and CBN lambda calculi [3]. The main steps in the development are:

1. We strengthen our goal to become:

$$e \longrightarrow^* v_1 \Longrightarrow (\exists v_3 \in \mathbb{V}.\, e \hookrightarrow v_3 \longrightarrow^* v_1).$$

2. We define a *left reduction function* $\longmapsto$ (Section 2.11) such that (Lemma 36): $\forall e \in \mathbb{E}, v \in \mathbb{V}$.

$$e \longmapsto^* v \iff e \hookrightarrow v$$

and $\forall e_1, e_2 \in \mathbb{E}. e_1 \longmapsto e_2 \implies e_1 \longrightarrow e_2$ (Lemma 35). Thus, big-step evaluation (or simply evaluation) is exactly a chain of left reductions that ends in a value.
3. Our goal is restated as:

$$e \longrightarrow^* v_1 \implies (\exists v_3 \in \mathbb{V}. e \longmapsto^* v_3 \longrightarrow^* v_1).$$

4. For technical reasons, the proofs are simpler if we use a parallel reduction relation $\gg$ (Section 2.8) similar to the one introduced in the last section. Our goal is once again restated as:

$$e \gg^* v_1 \implies (\exists v_3 \in \mathbb{V}. e \longmapsto^* v_3 \gg^* v_1).$$

5. The left reduction function induces a very fine classification $(\mathbb{V}, \mathbb{W}, \mathbb{S})$ on terms. In particular, any term $e \in \mathbb{E}$ must be exactly one of the following three (Lemma 20):
   (a) a *value* $e \in \mathbb{V}$,
   (b) a *workable* $e \in \mathbb{W}$, or
   (c) a *stuck* $e \in \mathbb{S}$,
   where membership in each of these three sets is defined inductively over the structure of the term. We write v, w and s to refer to a member of one of the three sets above, respectively. Left reduction at level n is a total function exactly on the members of the set W^n (Lemma 22). Thus, left reduction is strictly undefined on non-workables, that is, it is undefined on values and on stuck terms. Furthermore, if the result of any parallel reduction is a value, the source must have been either a value or a workable (Lemma 21). We will refer to this property of parallel reduction as *monotonicity*.
6. Using the above classification, we break our goal into two cases, depending on whether the starting point is a value or a workable:
   G1 $\forall v_1, v \in \mathbb{V}$.

$$v \gg^* v_1 \implies (\exists v_3 \in \mathbb{V}. v = v_3 \gg^* v_1),$$

   G2 $\forall w \in W, v \in \mathbb{V}$.

$$w \gg^* v_1 \implies (\exists v_3 \in \mathbb{V}. w \longmapsto^+ v_3 \gg^* v_1).$$

   It is obvious that G1 is true. Thus, G2 becomes the current goal.
7. By the monotonicity of parallel reduction, it is clear that all the intermediate terms in the reduction chain $w \gg^* v_1$ are either workables or values. Furthermore, workables and values do not interleave, and there is exactly one transition from workables to values in the chain. Thus, this chain can be visualised as follows:

$$w_1 \gg w_2 \gg \ldots w_{k-1} \gg w_k \gg v \gg^* v_1.$$

   We prove that the transition $w_k \gg v$ can be replaced by an evaluation (Lemma 37):
   R1 $\forall w \in W, v \in \mathbb{V}$.

$$w \gg v \implies (\exists v_2 \in \mathbb{V}. w \longmapsto^+ v_2 \gg v).$$

   With this lemma, we know that we can replace the chain above by one where the evaluation involved in going from the last workable to the first value is explicit:

$$w_1 \gg w_2 \gg \ldots w_{k-1} \gg w_k \longmapsto^+ v_2 \gg^* v_1.$$

   What is left is then to "push back" this information about the last workable in the chain to the very first workable in the chain. This is achieved by a straightforward iteration (by induction over the number of k of workables in the chain) of a result that we prove (Lemma 32):
   R2 $\forall w_1, w_2 \in W, v_1 \in \mathbb{V}$.

$$w_1 \gg w_2 \longmapsto^+ v_1 \implies (\exists v_2 \in \mathbb{V}. w_1 \longmapsto^+ v_2 \gg v_1).$$

With this result, we are able to move the predicate $\_ \longmapsto^+ v_3 \gg^* v$ all the way back to the first workable in the chain. This step can be visualised as follows. With one application of R2 we have the chain:

$$w_1 \gg w_2 \gg \ldots w_{k-1} \longmapsto^+ v_3 \gg^* v_1,$$

and with $k-2$ applications of R2 we have:

$$w_1 \longmapsto^+ v_{k+1} \gg^* v_1,$$

thus completing the proof.

□

6.3 Push Back

Lemma 32 (Push Back) $\forall X \in \mathbb{N}, w_1, w_2 \in W, v_2 \in \mathbb{V}.$

$$w_1 \overset{X}{\gg} w_2 \longmapsto^+ v_1 \Longrightarrow (\exists v_2 \in \mathbb{V}.\, w_1 \longmapsto^+ v_2 \gg v_1).$$

Proof. The assumption corresponds to a chain of reductions:

$$w_1 \gg w_2 \longmapsto w_3 \longmapsto \ldots w_{k-1} \longmapsto w_k \longmapsto v_1.$$

Applying Permutation to $w_1 \gg w_2 \longmapsto w_3$ gives us $(\exists e_{2'} \in \mathbb{E}.\, w_1 \longmapsto^+ e_{2'} \gg w_3)$. By the monotonicity of parallel reduction, we know that only a workable can reduce to a workable, that is, $(\exists w_{2'} \in W^n.\, w_1 \longmapsto^+ w_{2'} \gg w_3)$. Now we have the chain:

$$w_1 \longmapsto^+ w_{2'} \gg w_3 \longmapsto \ldots w_{k-1} \longmapsto w_k \longmapsto v_1.$$

Repeating this step $k-2$ times we have:

$$w_1 \longmapsto^+ w_{2'} \longmapsto^+ w_{3'} \longmapsto^+ \ldots w_{k-1'} \gg w_k \longmapsto v_1.$$

Applying Permutation to $w_{k-1'} \gg w_k \longmapsto v_1$ give us $(\exists e_{k'} \in \mathbb{E}.\, w_{k-1'} \longmapsto^+ e_{k'} \gg v_1)$. By the monotonicity of parallel reduction, we know that $e_{k'}$ can only be a value or a workable. If it is a value then we have the chain:

$$w_1 \longmapsto^+ w_{2'} \longmapsto^+ w_{3'} \longmapsto^+ \ldots w_{k-1'} \longmapsto^+ v_{k'} \gg v_1$$

and we are done. If it is a workable, then applying Transition to $w_{k'} \gg v_1$ gives us $(\exists v_2 \in V.\, w_{k'} \longmapsto^+ v_2 \gg v_1)$. This means that we now have the chain:

$$w_1 \longmapsto^+ w_{2'} \longmapsto^+ w_{3'} \longmapsto^+ \ldots w_{k-1'} \longmapsto^+ w_{k'} \longmapsto^+ v_2 \gg v_1$$

and we are done. □

7 Special Part of Computational Adequacy of Reduction Semantics

Remark 33 (Non-termination and the Finiteness of Trees) *The reader should be reminded here that all the structures that we ever construct in this development are finite trees. Thus, non-terminating computations are* not *modelled by infinite derivations, but rather, by the absence of a "conclusive" derivation. In particular, a big-step derivation is simply absent for a "non-terminating" computation, and thus, the big-step semantics identifies stuck and non-terminating computations. Similarly, a small-step derivation is always defined on a non-terminating computation, but every finite sequence of small-step can be extended by another step. Thus, no finite sequence of small-steps leads to a value (or a stuck for that matter). Note, however, that the small-step semantics allows us to distinguish between a stuck (which cannot be advanced by small-step reduction) and a workable (which can be advanced an arbitrarily large number of times by small-step reduction).*

7.1 Evaluation is in Reduction

Lemma 34 (Evaluation is in Reduction) $\forall e \in \mathbb{E}, v \in \mathbb{V}$.

$$e \hookrightarrow v \Longrightarrow e \longrightarrow^* v.$$

Proof. By a straightforward induction on the height of the judgement $e \hookrightarrow v$.

1. $() \hookrightarrow ()$ and $() \longrightarrow^* ()$.
2. $\lambda x.e \hookrightarrow \lambda x.e$ and $\lambda x.e \longrightarrow^* \lambda x.e$
3. $\#z \hookrightarrow \#z$. obviously $\#z \longrightarrow^* \#z$
4. $\lambda(\#z \Rightarrow x).e \hookrightarrow \lambda(\#z \Rightarrow x).e$ and $\lambda(\#z \Rightarrow x).e \longrightarrow^* \lambda(\#z \Rightarrow x).e$.
5. $\lambda^{i \in L}\, f\ x_i.e_i \hookrightarrow \lambda^{i \in L}\, f\ x_i.e_i$ and $\lambda^{i \in L}\, f\ x_i.e_i \longrightarrow^* \lambda^{i \in L}\, f\ x_i.e_i$
6. $$\dfrac{\begin{array}{c} e_1 \hookrightarrow \lambda x.e \\ e_2 \hookrightarrow e_3 \\ e[x := e_3] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4}$$
   By induction hypothesis,$e_1 \longrightarrow^* \lambda x.e$, $e_2 \longrightarrow^* e_3$ and $e[x := e_3] \longrightarrow^* e_4$. Then, by compatibility of $\longrightarrow^*$: $e_1\ e_2 \longrightarrow^* (\lambda x.e)\ e_2 \longrightarrow^* (\lambda x.e)\ e_3 \longrightarrow_{\beta_1} e[x := e_3] \longrightarrow^* e_4$.
7. $$\dfrac{\dfrac{\begin{array}{c} e_1 \hookrightarrow \lambda(\#\_ \Rightarrow x).e \\ e_2 \hookrightarrow \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4}\hookrightarrow\Uparrow}{\begin{array}{c} e_1 \longrightarrow^* \lambda(\#\_ \Rightarrow x).e \\ e_2 \longrightarrow^* \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \longrightarrow^* e_4 \end{array}}{\scriptstyle IH\Downarrow\ IH\Downarrow\ IH\Downarrow}$$
   $$\dfrac{\begin{array}{c} e_1 \longrightarrow^* \lambda(\#\_ \Rightarrow x).e \\ e_2 \longrightarrow^* \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \longrightarrow^* e_4 \end{array}}{e_1\ e_2 \longrightarrow^* (\lambda(\#\_ \Rightarrow x).e)\ e_2 \longrightarrow^* (\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b_3) \longrightarrow_{\beta_2} e[x := \lambda x.b_3[\#z := x']] \longrightarrow^* e_4}\text{comp.}\ \longrightarrow^*\Downarrow$$
8. $$\dfrac{\dfrac{\begin{array}{c} e_1 \hookrightarrow \lambda^{1 \in L \cup \{k\}}\ f_i\ x_i.e_i \\ e_2 \hookrightarrow f_k\ e_4 \quad k \leq n \\ e_k[x := e_4] \hookrightarrow e_5 \end{array}}{e_1\ e_2 \hookrightarrow e_5}\hookrightarrow\Uparrow}{\begin{array}{c} e_1 \longrightarrow^* \lambda^{i \in L \cup \{k\}}\ f_i\ x_i.e_i \\ e_2 \longrightarrow^* f_k\ e_4 \quad k \leq n \\ e_k[x := e_4] \longrightarrow^* e_5 \end{array}}{\scriptstyle IH\Downarrow\ IH\Downarrow\ IH\Downarrow}$$
   $$\dfrac{\begin{array}{c} e_1 \longrightarrow^* \lambda^{i \in L \cup \{k\}}\ f_i\ x_i.e_i \\ e_2 \longrightarrow^* f_k\ e_4 \quad k \leq n \\ e_k[x := e_4] \longrightarrow^* e_5 \end{array}}{e_1\ e_2 \longrightarrow^* (\lambda^{i \in F \cup \{k\}}\ f_i\ x_i.e_i)\ e_2 \longrightarrow^* (\lambda^{i \in F \cup \{k\}}\ f_i\ x_i.e_i)\ (f_k\ e_4) \longrightarrow_{\beta_3} e_k[x := e_4] \longrightarrow^* e_5}\text{comp.}\ \longrightarrow^*\Downarrow$$
9. $$\hookrightarrow\Uparrow \dfrac{\begin{array}{c} e_1 \hookrightarrow v_1 \\ e_2 \hookrightarrow v_2 \end{array}}{(e_1, e_2) \hookrightarrow (v_1, v_2)} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e_1 \longrightarrow^* v_1 \\ e_2 \longrightarrow^* v_2 \end{array}}{(e_1, e_2) \longrightarrow^* (v_1, e_2) \longrightarrow^* (v_1, v_2)}\text{comp.}\ \longrightarrow^*\Downarrow$$
10. $$\hookrightarrow\Uparrow \dfrac{e \hookrightarrow (v_1, v_2)}{\pi_1\ e \hookrightarrow v_1} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* (v_1, v_2)}{\pi_1\ e \longrightarrow^* \pi_1\ (v_1, v_2) \longrightarrow_{\pi_1} v_1}\text{comp.}\ \longrightarrow^*\Downarrow$$
11. $$\hookrightarrow\Uparrow \dfrac{e \hookrightarrow (v_1, v_2)}{\pi_2\ e \hookrightarrow v_2} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* (v_1, v_2)}{\pi_2\ e \longrightarrow^* \pi_2\ (v_1, v_2) \longrightarrow_{\pi_2} v_2}\text{comp.}\ \longrightarrow^*\Downarrow$$
12. $$\hookrightarrow\Uparrow \dfrac{e_1 \hookrightarrow e_2}{f_k\ e_1 \hookrightarrow f_k\ e_2} \overset{IH}{\Longrightarrow} \dfrac{e_1 \longrightarrow^* e_2}{f_k\ e_1 \longrightarrow^* f_k\ e_2}\text{comp.}\ \longrightarrow^*\Downarrow$$
13. $$\hookrightarrow\Uparrow \dfrac{e_1 \hookrightarrow v}{\#z \Rightarrow e_1 \hookrightarrow \#z \Rightarrow v} \overset{IH}{\Longrightarrow} \dfrac{e_1 \longrightarrow^* v}{\#z \Rightarrow e_1 \longrightarrow^* \#z \Rightarrow v}\ \text{comp.}\ \longrightarrow^*\Downarrow$$
14. $$\hookrightarrow\Uparrow \dfrac{\begin{array}{c} e_1 \hookrightarrow \#z \\ e_2 \hookrightarrow \#z \end{array}}{e_1 =_\# e_2 \hookrightarrow \mathsf{True}()} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e_1 \longrightarrow^* \#z \\ e_2 \longrightarrow^* \#z \end{array}}{e_1 =_\# e_2 \longrightarrow^* \#z = e_2 \longrightarrow^* \#z = \#z \longrightarrow_\# \mathsf{True}()}\text{comp.}\ \longrightarrow^*\Downarrow$$

15. $\hookrightarrow\Uparrow$ $\dfrac{\begin{array}{c} e_1 \hookrightarrow \#z \\ e_2 \hookrightarrow \#z' \quad \#z \neq \#z' \end{array}}{e_1 = e_2 \hookrightarrow \mathsf{False}()} \overset{IH}{\Longrightarrow} \dfrac{\begin{array}{c} e_1 \longrightarrow^* \#z \\ e_2 \longrightarrow^* \#z' \end{array}}{e_1 =_\# e_2 \longrightarrow^* \#z = e_2 \longrightarrow^* \#z = \#z' \longrightarrow_\# \mathsf{False}()}\mathrm{comp.}_{\dashrightarrow^*}\Downarrow$

16. $\hookrightarrow\Uparrow$ $\dfrac{e \hookrightarrow \#z}{\mathsf{isOVar}\ e \hookrightarrow \mathsf{True}()} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* \#z}{\mathsf{isOVar}\ e \longrightarrow^* \mathsf{isOVar}\ \#z \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{True}()}\ \mathrm{comp.}_{\dashrightarrow^*}\Downarrow$

17. $\hookrightarrow\Uparrow$ $\dfrac{e \hookrightarrow v \quad v \neq \#z}{\mathsf{isOVar}\ e \hookrightarrow \mathsf{False}()} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow^* v \quad v \neq \#z}{\mathsf{isOVar}\ e \longrightarrow^* \mathsf{isOVar}\ v \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{False}()}\ \mathrm{comp.}_{\dashrightarrow^*}\Downarrow$

What is harder to show is the "converse", that is, that $e \longrightarrow^* v \Longrightarrow (\exists v' \in V. e \hookrightarrow v')$. It is a consequence of the stronger result of Lemma 31.

In the rest of this section, we present the definitions and lemmas mentioned above.

7.2 Left Reduction is in Reduction

Lemma 35 (Left Reduction is in Reduction) $\forall e_1, e_2 \in \mathbb{E}$.

$$e_1 \longmapsto e_2 \Longrightarrow e_1 \longrightarrow e_2.$$

Proof (Lemma 35). Proof is straightforward, by induction on the height of the first judgement.

1. If $(\lambda x.e)\ v \longmapsto e[x := v]$, and $(\lambda x.e)\ v \longrightarrow_{\beta_1} e[x := v]$.
2. $\dashrightarrow\Uparrow$ $\dfrac{e_1 \longmapsto e_1'}{e_1\ e_2 \longmapsto e_1'\ e_2} \overset{IH}{\Longrightarrow} \dfrac{e_1 \longrightarrow e_1'}{e_1\ e_2 \longrightarrow e_1'\ e2}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
3. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{v\ e \longmapsto v\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{v\ e \longrightarrow v\ e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
4. $\dashrightarrow\Uparrow$ $\dfrac{e_1 \longmapsto e_1'}{(e_1, e_2) \longmapsto (e_1', e_2)} \overset{IH}{\Longrightarrow} \dfrac{e_1 \longrightarrow e_1'}{(e_1, e_2) \longrightarrow (e_1', e_2)}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
5. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{(v, e) \longmapsto (v, e')} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{(v, e) \longrightarrow (v, e')}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
6. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{\pi_n\ e \longmapsto \pi_n\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{\pi_n\ e \longrightarrow \pi_n\ e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
7. $\pi_1\ (v_1, v_2) \longmapsto v_1$, and $\pi_1\ (v_1, v_2) \longrightarrow_\pi v_1$
8. $\pi_2\ (v_1, v_2) \longmapsto v_2$, and $\pi_2\ (v_1, v_2) \longrightarrow_\pi v_1$
9. $(\lambda^{i \in L \cup \{k\}}\ f_i\ x_i.e_i)\ (f_k\ v) \longmapsto e_k[x_k := v]$ and $(\lambda^{i \in L \cup \{k\}}\ f_i\ x_i.e_i)\ (f_k\ v) \longrightarrow_{\beta_3} e_k[x_k := v]$.
10. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{f\ e \longmapsto f\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{f\ e \longrightarrow f\ e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
11. $(\lambda(\#z \Rightarrow x).e)\ (\#z' \Rightarrow b) \longmapsto e[x := \lambda x'.b[\#z' := x']]$, and $(\lambda(\#z \Rightarrow x).e)\ (\#z' \Rightarrow b) \longrightarrow_{\beta_2} e[x := \lambda x'.b[\#z' := x']]$.
12. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{\#z \Rightarrow e \longmapsto \#z \Rightarrow e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{\#z \Rightarrow e \longrightarrow \#z \Rightarrow e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
13. $\#z =_\# z \longmapsto \mathsf{True}()$, and $\#z = \#z \longrightarrow_\# \mathsf{True}()$
14. $\dfrac{\#z \neq \#z'}{\#z =_\# z' \longmapsto \mathsf{False}()}$ and $\dfrac{\#z \neq \#z'}{\#z = \#z' \longrightarrow_\# \mathsf{False}()}$.
15. $\dashrightarrow\Uparrow$ $\dfrac{e_1 \longmapsto e_1'}{e_1 =_\# e_2 \longmapsto e_1' =_\# e_2} \overset{IH}{\Longrightarrow} \dfrac{e_1 \longrightarrow e_1'}{e_1 = e_2 \longrightarrow e_1' = e_2}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
16. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{\#z' =_\# e \longmapsto \#z' =_\# e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{\#z' = e \longrightarrow \#z' = e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$
17. $\mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()$ and $\mathsf{isOVar}\ \#z \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{True}()$
18. If $v \neq \#z$, $\mathsf{isOVar}\ v \longmapsto \mathsf{True}()$ and $\mathsf{isOVar}\ v \longrightarrow_{\delta_{\mathsf{isOVar}}} \mathsf{True}()$
19. $\dashrightarrow\Uparrow$ $\dfrac{e \longmapsto e'}{\mathsf{isOVar}\ e \longmapsto \mathsf{isOVar}\ e'} \overset{IH}{\Longrightarrow} \dfrac{e \longrightarrow e'}{\mathsf{isOVar}\ e \longrightarrow \mathsf{isOVar}\ e'}\ \mathrm{comp.}_{\dashrightarrow}\Downarrow$

□

7.3 Left Reduction is Evaluation

Lemma 36 (Left Reduction and Big-step Semantics) $\forall e \in \mathbb{E}, v \in \mathbb{V}$.

$$e \longmapsto^* v \iff e \hookrightarrow v.$$

Proof. **The forward direction:** $\forall e \in \mathbb{E}, v \in \mathbb{V}. e \longmapsto^* v \implies e \hookrightarrow v$

By induction on the length k of derivation $e \longmapsto^k$, and then by induction on the size of e. It proceeds by case analysis on e.

- $k = 0$
  1. For all values v, $v \longmapsto^* v$, since $\longmapsto^*$ is reflexive. For all other expressions $e \longmapsto^* e$, e is not a value, and the property holds by contradiction.
- $k = n + 1$.
  1. $e_1\ e_2 \longmapsto^i (\lambda x.e)\ e_2 \longmapsto^j (\lambda x.e)\ v' \longmapsto^1 e[x := v'] \longmapsto^k v$ where $i + j + k = n$. Then, by induction hypothesis

$$\frac{\begin{array}{c} e_1 \hookrightarrow (\lambda x.e) \\ e_2 \hookrightarrow v' \\ e[x := v'] \hookrightarrow v \end{array}}{e_1\ e_2 \hookrightarrow v} \hookrightarrow \Downarrow$$

  2. $e_1\ e_2 \longmapsto^i (\lambda(\#\_ \Rightarrow x).e)\ e_2 \longmapsto^j (\lambda x.e)\ (\#z \Rightarrow b') \longmapsto^1 e[x := \lambda x'.b'[\#z := x']] \longmapsto^k v$ where $i + j + k = n$. Then, by induction hypothesis

$$\frac{\begin{array}{c} e_1 \hookrightarrow \lambda(\#\_ \Rightarrow x).e \\ e_2 \hookrightarrow \#z \Rightarrow b' \\ e[x := \lambda x'.b'[\#z := x']] \hookrightarrow v \end{array}}{e_1\ e_2 \hookrightarrow v} \hookrightarrow \Downarrow$$

  3. $e_1\ e_2 \longmapsto^i (\lambda^{l \in F \cup \{o\}}\ f_l\ x_l.e_l)\ e_2 \longmapsto^j (\lambda^{l \in F \cup \{o\}}\ f_l\ x_l.e_l)\ (f_o\ v') \longmapsto^1 e_o[x := v'] \longmapsto^k v$ where $i + j + k = n$. Then, by induction hypothesis

$$\frac{\begin{array}{c} e_1 \hookrightarrow (\lambda^{l \in F \cup \{o\}}\ f_l\ x_l.e_l) \\ e_2 \hookrightarrow f_o\ v' \\ e_l[x := v'] \hookrightarrow v \end{array}}{e_1\ e_2 \hookrightarrow v} \hookrightarrow \Downarrow$$

  4. $(e_1, e_2) \longmapsto^i (v_1, e_2) \longmapsto^j (v_1, v_2)$, where $i + j = n + 1$. Then, by the induction hypothesis

$$\frac{e_1 \hookrightarrow v_1 \quad e_2 \hookrightarrow v_2}{(e_1, e_2) \hookrightarrow (v_1, v_2)} \hookrightarrow \Downarrow$$

  5. $\pi_1\ e \longmapsto^n \pi_1\ (v_1, v_2) \longmapsto^1 v_1$. For each step in the n first reductions, clearly the same rule of the left reduction applies, namely $\dfrac{e \longmapsto e'}{\pi_1\ e \longmapsto \pi_1\ e'}$. Thus it easily follows that $e \longmapsto^n (v_1, v_2)$. Applying the induction hypothesis to this result, we obtain $e \hookrightarrow (v_1, v_2)$. By definition of $\hookrightarrow$, from this follows $\pi_1\ e \hookrightarrow v_1$.
  6. $\pi_2\ e \longmapsto^n \pi_2\ (v_1, v_2) \longmapsto^1 v_2$. For each step in the n first reductions, clearly the same rule of the left reduction applies, namely $\dfrac{e \longmapsto e'}{\pi_2\ e \longmapsto \pi_2\ e'}$. Thus it easily follows that $e \longmapsto^n (v_1, v_2)$. Applying the induction hypothesis to this result, we obtain $e \hookrightarrow (v_1, v_2)$. By definition of $\hookrightarrow$, from this follows $\pi_2\ e \hookrightarrow v_2$.
  7. $$\frac{e_1 \longmapsto^{n+1} v}{f\ e_1 \longmapsto^{n+1} f\ v} \overset{IH}{\Longrightarrow} \frac{e_1 \hookrightarrow v}{f\ e_1 \hookrightarrow f\ v}$$
  8. $$\frac{e_1 \longmapsto^{n+1} v}{\#z \Rightarrow e_1 \longmapsto^{n+1} \#z \Rightarrow v} \overset{IH}{\Longrightarrow} \frac{e_1 \hookrightarrow v}{\#z \Rightarrow e_1 \hookrightarrow \#z \Rightarrow v} \hookrightarrow \Downarrow$$

9. $e_1 = e_2 \longmapsto^i \#z = e_2 \longmapsto^j \#z = \#z \longmapsto^1 \mathsf{True}$, where $i + j = n$. Then, by the induction hypothesis

$$\frac{\begin{array}{c} e_1 \hookrightarrow \#z \\ e_2 \hookrightarrow \#z \end{array}}{e_1 = e_2 \hookrightarrow \mathsf{True}()} \hookrightarrow\Downarrow$$

10. Similarly,
$e_1 = e_2 \longmapsto^i \#z = e_2 \longmapsto^j \#z = \#z' \longmapsto^1 \mathsf{False}$, where $i + j = n$. Then, by the induction hypothesis

$$\frac{\begin{array}{c} e_1 \hookrightarrow \#z \\ e_2 \hookrightarrow \#z' \end{array}}{e_1 = e_2 \hookrightarrow \mathsf{False}()} \hookrightarrow\Downarrow$$

11. $\mathsf{isOVar}\ e \longmapsto^i \mathsf{isOVar}\ \#z \longmapsto^1 \mathsf{True}()$, then $e \longmapsto^i \#z$. By the induction hypothesis $e \hookrightarrow \#z$. Then, $\mathsf{isOVar}\ e \hookrightarrow \mathsf{True}()$.
12. $\mathsf{isOVar}\ e \longmapsto^i \mathsf{isOVar}\ v \longmapsto^1 \mathsf{False}()$, then $e \longmapsto^i v$. By the induction hypothesis $e \hookrightarrow v$. Then, $\mathsf{isOVar}\ e \hookrightarrow \mathsf{False}()$.

The backward direction: $\forall e \in \mathbb{E}.\ \forall v \in \mathbb{V}. e \hookrightarrow v \Longrightarrow e \longmapsto^* v$. By induction on the height of the derivation of $e \hookrightarrow v$, and then by the size of e.

1. $\dfrac{}{() \hookrightarrow ()}$ and $() \longmapsto^0 ()$.
2. $\dfrac{}{\lambda x.e \hookrightarrow \lambda x.e}$ and $\lambda x.e \longmapsto^0 \lambda x.e$.
3. $\dfrac{\begin{array}{l} e_1 \hookrightarrow \lambda x.e \\ e_2 \hookrightarrow e_3 \\ e[x := e_3] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4}$ By induction hypothesis: $\begin{array}{l} e_1 \longmapsto^i (\lambda x.e) \\ e_2 \longmapsto^j v \\ e[x := e_3] \longmapsto^k e_4 \end{array}$. Then, $e_1\ e_2 \longmapsto^i (\lambda x.e)\ e_2 \longmapsto^j (\lambda x.e)\ v \longmapsto^1 e[x := v] \longmapsto^k e_4$.
4. $\dfrac{\begin{array}{l} e_1 \hookrightarrow \lambda^{f \in L \cup \{k\}} (f\ x_f).e_f \\ e_2 \hookrightarrow f_k\ e_4 \\ e_k[x := e_4] \hookrightarrow e_5 \end{array}}{e_1\ e_2 \hookrightarrow e_5}$. By induction hypothesis: $\begin{array}{l} e_1 \longmapsto^i \lambda^{i \in L \cup \{k\}} (f\ x_f).e_f \\ e_2 \longmapsto^j k\ e_4 \\ e_k[x := e_4] \longmapsto^l e_5 \end{array}$ Then, $e_1\ e_2 \longmapsto^i (\lambda^{i \in L \cup \{k\}} (f_i\ x_i).e_i)\ e_2 \longmapsto^j (\lambda^{i \in L \cup \{k\}} (f_i\ x_i).e_i)\ (k\ e_4) \longmapsto^1 e_k[x := e_4] \longmapsto^l e_5$.
5. $\dfrac{\begin{array}{l} e_1 \hookrightarrow \lambda(\#\_ \Rightarrow x).e \\ e_2 \hookrightarrow \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \hookrightarrow e_4 \end{array}}{e_1\ e_2 \hookrightarrow e_4}$.
By the induction hypothesis $\begin{array}{l} e_1 \longmapsto^i \lambda(\#\_ \Rightarrow x).e \\ e_2 \longmapsto^j \#z \Rightarrow b_3 \\ e[x := \lambda x'.(b_3[\#z := x'])] \longmapsto^k e_4 \end{array}$.
Then, $e_1\ e_2 \longmapsto^i (\lambda(\#\_ \Rightarrow x).e)\ e_2 \longmapsto^j (\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b_3) \longmapsto^1 e[x := \lambda x'.b[\#z := x']] \longmapsto^k e_4$.
6. $\dfrac{e_1 \hookrightarrow e_3 \quad e_2 \hookrightarrow e_4}{(e_1, e_2) \hookrightarrow (e_3, e_4)}$ By the induction hypothesis $\begin{array}{l} e_1 \longmapsto^i e_3 \\ e_2 \longmapsto^j e_4 \end{array}$ Then, $(e_1, e_2) \longmapsto^i (e_3, e_2) \longmapsto^j (e_3, e_4)$.
7. $\dfrac{e \hookrightarrow (e_3, e_4)}{\pi_1\ e \hookrightarrow e_3}$. By the induction hypothesis:$e \longmapsto^n (e_3, e_4)$. Then, $\pi_1\ e \longmapsto^n \pi_1\ (e_3, e_4) \longmapsto^1 e_3$.
8. $\dfrac{e \hookrightarrow (e_3, e_4)}{\pi_2\ e \hookrightarrow e_4}$. By the induction hypothesis: $e \longmapsto^n (e_3, e_4)$. Then, $\pi_2\ e \longmapsto^n \pi_2\ (e_3, e_4) \longmapsto^1 e_4$.
9. $\dfrac{e_1 \hookrightarrow e_2}{f_k\ e_1 \hookrightarrow f_k\ e_2}$ By the induction hypothesis $e_1 \longmapsto^{n+1} e_2$. Then, $f\ e_1 \longmapsto^{n+1} f\ e_2$.
10. $\dfrac{}{\lambda^{i \in L} f_i\ x_i.e_i \hookrightarrow \lambda^{i \in L} f_i\ x_i.e_i}$ and $\lambda^{i \in L} f_i\ x_i.e_i \longmapsto^0 \lambda^{i \in L} f_i\ x_i.e_i$.
11. $\dfrac{e \hookrightarrow v}{\#z \Rightarrow e \hookrightarrow \#z \Rightarrow v}$ By the induction hypothesis $e \longmapsto^{n+1} v$. Then, $\#z \Rightarrow e \longmapsto^{n+1} \#z \Rightarrow v$.
12. $\#z \hookrightarrow \#z$, and $\#z \longmapsto^0 \#z$.

13. $\lambda(z \Rightarrow x).e \hookrightarrow \lambda(z \Rightarrow x).e$, and $\lambda(z \Rightarrow x).e \longmapsto^0 \lambda(z \Rightarrow x).e$.
14. $\dfrac{e_1 \hookrightarrow \#z \quad e_2 \hookrightarrow \#z}{e_1 =_\# e_2 \hookrightarrow \mathsf{True}()}$. By the induction hypothesis: $\begin{matrix} e_1 \longmapsto^i \#z \\ e_2 \longmapsto^j \#z \end{matrix}$. Then, $e_1 =_\# e_2 \longmapsto^i \#z =_\# e_2 \longmapsto^j \#z =_\# \#z \longmapsto^1 \mathsf{True}()$.
15. $\dfrac{e_1 \hookrightarrow \#z \quad e_2 \hookrightarrow \#z' \quad \#z' \neq \#z}{e_1 =_\# e_2 \hookrightarrow \mathsf{True}()}$. By the induction hypothesis: $\begin{matrix} e_1 \longmapsto^i \#z \\ e_2 \longmapsto^j \#z' \end{matrix}$. Then, $e_1 =_\# e_2 \longmapsto^i \#z =_\# e_2 \longmapsto^j \#z =_\# \#z' \longmapsto^1 \mathsf{False}()$.
16. $Rulee \hookrightarrow \#z\mathsf{isOVar}\ e \hookrightarrow \mathsf{True}()$ By the induction hypothesis: $e \longmapsto^* \#z$. Then, $\mathsf{isOVar}\ e \longmapsto^* \mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()$.
17. $\dfrac{e \hookrightarrow v \quad v \neq \#z}{\mathsf{isOVar}\ e \hookrightarrow \mathsf{False}()}$ By the induction hypothesis: $e \longmapsto^* v$ (and $v \neq \#z$). Then, $\mathsf{isOVar}\ e \longmapsto^* \mathsf{isOVar}\ v \longmapsto \mathsf{False}()$.

7.4 Transition Lemma

Lemma 37 (Transition) $\forall X \in \mathbb{N}.\, \forall w \in W, v \in \mathbb{V}.$

$$w \overset{X}{\gg} v \Longrightarrow \exists v_2 \in \mathbb{V}, Y \in \mathbb{N}.\, w \longmapsto^+ v_2 \gg v \ \wedge\ Y < X.$$

Proof (Lemma 37(Transition)). Proof is by induction on the complexity X, and then on the structure of w.

1. For the workable $(\lambda x.e)\ v$, $\dfrac{e \overset{M}{\gg} e' \quad v \overset{N}{\gg} v'}{(\lambda x.e)\ v \overset{M+\#(x,e')N+1}{\gg} e'[x := v']}$
   Since, $e \overset{M}{\gg} e'$ and $v \overset{N}{\gg} v'$, then by lemma 18, $\exists Z.\, e[x := v] \overset{Z}{\gg} e'[x := v'] \wedge Z \leq M + \#(x, e')N$. Now, there are two possibilities.
   (a) First, if e[x:=v] is a workable, then since $Z < M + \#(x, e')N + 1$, the induction hypothesis applies to $e[x := v]$, and therefore: $\exists v_1.\exists Y_1.e[x := v] \longmapsto^+ v_1 \overset{Y}{\gg} e'[x := v'] \wedge Y_1 < Z$ Thus, we obtain our goal as follows: $\exists v_2 = v_1.\, \exists Y = Y_1.$

   $$(\lambda x.e)\ v \longmapsto^1 e[x := v] \longmapsto^+ v_2 \overset{Y}{\gg} e'[x := v'] \wedge Y < M + \#(x, e')N + 1$$

   .
   (b) Otherwise, if $e[x := v]$ is a value, then $\exists v_2 = e[x := v].\, \exists Y = Z.$

   $$(\lambda x.e)\ v \longmapsto^1 e[x := v] \overset{Y}{\gg} e'[x := v'] \wedge Y < M + \#(x, e')N + 1$$

2. For the workable $(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b)$ $\dfrac{e \overset{M}{\gg} e'}{(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \overset{M+1}{\gg} e'[x := \lambda x'.b[\#z := x']]}$. Since $e \overset{M}{\gg} e'$ and $\lambda x'.b[\#z := x'] \overset{0}{\gg} \lambda x'.b[\#z := x']$, by lemma 18 $\exists Z.\, e[x := \lambda x'.b[\#z := x']] \overset{Z}{\gg} e'[x := \lambda x'.b[\#z := x']] \wedge Z \leq M$. Again, there are two possibilities
   (a) $e[x := \lambda x'.b[\#z := x']]$ is a workable. Then, since $Z < M + 1$, the induction hypothesis applies and we obtain $\exists v_1.\, \exists Y_1.\, e[x := \lambda x'.b[\#z := x']] \longmapsto^+ v_1 \overset{Y_1}{\gg} e'[x := \lambda x'.b[\#z := x']] \wedge Y_1 < Z$. Then, it easily follows $\exists v_2 = v_1.\exists Y = Y_1.$,

   $$(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \longmapsto^1 e[x := \lambda x'.b[\#z := x']] \longmapsto^+ v_2 \overset{Y}{\gg} e'[x := \lambda x'.b[\#z := x']] \wedge Y < M + 1$$

   (b) Otherwise, $e[x := \lambda x'.b[\#z := x']]$ is a value. Then, $\exists v_2 = e[x := \lambda x'.b[\#z := x']].\, \exists Y = Z.$

   $$(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \longmapsto^1 e[x := \lambda x'.b[\#z := x']] \overset{Y}{\gg} e'[x := \lambda x'.b[\#z := x']] \wedge Y < M + 1$$

   .

3. The workable $w\ e$ can never parallel-reduce to a value in one parallel reduction step, so the property vacuously nolds.
4. Similarly, the workable $v\ w$ can never parallel-reduce to a a value in one parallel reduction step, so the property vacuously holds.
5. For the workable $f\ w$, $\dfrac{w \overset{X}{\gg} v}{f\ w \overset{X}{\gg} v}$.

   By the induction hypothesis $\exists v_1.\, \exists Y_1.\, w \longmapsto^+ v_2 \overset{Y_1}{\gg} v \wedge Y_1 < X$. Then, it easily follows that $\exists v_2 = f\ v_1.\, \exists Y = Y_1$.

$$f\ w \longmapsto^+ f\ v_2 \overset{Y}{\gg} f\ v \ \wedge\ Y < X$$

   .
6. For the workable $\#z \Rightarrow w$, $\dfrac{w \overset{M}{\gg} v}{\#z \Rightarrow w \overset{M}{\gg} \#z \Rightarrow v}$.

   Then, by the induction hypothesis, $\exists v_1.\, \exists Y_1.\, w \longmapsto^+ v_1 \overset{Y_1}{\gg} v \ \wedge\ Y_1 \leq M$.
   Then, it easily follows that $\exists v_2 = \#z \Rightarrow v_1.\, \exists Y = Y_1.\, (\#z \Rightarrow w) \longmapsto^+ v_2 \overset{Y}{\gg} (\#z \Rightarrow v) \ \wedge\ Y < M$.
7. For the workable (w, e), $\dfrac{w \overset{M}{\gg} v_1 \quad e \overset{N}{\gg} v_2}{(w, e) \overset{M+N}{\gg} (v_1, v_2)}$.

   Then, by the induction hypothesis $\exists v_3.\, \exists Y_1.\, w \longmapsto^+ v_3 \overset{Y_1}{\gg} v_1 \ \wedge\ Y_1 < M$.
   From this it easily follows: $\exists v_4 = (v_3, e).\, \exists Y = Y_1 + N$.

$$(w, e) \longmapsto^+ (v_3, e) \overset{Y}{\gg} (v_1, v_2) \ \wedge\ Y < M + N$$

8. For the workable (v, w), $\dfrac{v \overset{M}{\gg} v_1 \quad w \overset{N}{\gg} v_2}{(v, w) \overset{M+N}{\gg} (v_1, v_2)}$.

   Then, by the induction hypothesis: $\exists v_3.\, \exists Y_1.\, w \longmapsto^+ v_3 \overset{Y_1}{\gg} v_2 \ \wedge\ Y < N$.
   From this it easily follows: $\exists v_3 = (v, v_3).\, \exists Y = M + Y_1$.

$$(v, w) \longmapsto^+ (v, v_3) \overset{Y}{\gg} (v_1, v_2) \ \wedge\ Y < M + N$$

9. For the workable $w = e$, it is easy to show that it can never be parallel-reduced in one steps to a value, so the property holds vacuously.
10. Similarly for the workable $\#z = w$ it is easy to show that it can never be parallel-reduced in one step to a value, so the property holds vacuously.
11. For the workable $\#z = \#z'$, $\#z = \#z' \overset{1}{\gg} \mathsf{True}()$ if $\#z = \#z'$. Then, obviously, $\exists v_2 = \mathsf{True}().\, \exists Y = 0.\, \#z = \#z' \longmapsto^1 \mathsf{True}() \overset{0}{\gg} \mathsf{True}() \wedge 0 < 1$.
12. For the workable $\#z = \#z'$, $\#z = \#z' \overset{1}{\gg} \mathsf{False}()$ if $\#z \neq \#z'$. Then, obviously, $\exists v_2 = \mathsf{False}().\, \exists Y = 0.\, \#z = \#z' \longmapsto^1 \mathsf{False}() \overset{0}{\gg} \mathsf{False}() \wedge 0 < 1$.
13. For the workable $\pi_1\ w$, it is easy to show that they could never reduced in one step to values by a single parallel reduction step, so the property vacuously holds.
14. For the workable $\pi_2\ w$ it is easy to show that they could never reduced in one step to values by a single parallel reduction step, so the property vacuously holds.
15. For the workable $\pi_1\ (v_1, v_2)$, $\dfrac{v_1 \overset{M}{\gg} v_1'}{\pi_1\ (v_1, v_2) \overset{M+1}{\gg} v_1'}$. Then, $\exists v_2 = v_1.\, \exists Y = M.\, \pi_1\ (v_1, v_2) \longmapsto^1 v_1 \overset{M}{\gg} v_1' \wedge Y < M+1$.
16. For the workable $\pi_2\ (v_1, v_2)$, $\dfrac{v_2 \overset{M}{\gg} v_2'}{\pi_2\ (v_1, v_2) \overset{M+1}{\gg} v_2'}$. Then, $\exists v_2 = v_2.\, \exists Y = M.\, \pi_2\ (v_1, v_2) \longmapsto^1 v_2 \overset{M}{\gg} v_2' \wedge Y < M+1$.
17. For the workable $\mathsf{isOVar}\ \#z$, $\mathsf{isOVar}\ \#z \overset{1}{\gg} \mathsf{True}()$. Then, $\exists v_2 = \mathsf{True}().\, \exists Y = 0.\, \mathsf{isOVar}\ \#z \longmapsto^+ \mathsf{True}() \overset{0}{\gg} \mathsf{True}() \ \wedge\ Y < 1$.

18. For the workable $\mathsf{isOVar}\ v$, where $v \neq \#z$, $\mathsf{isOVar}\ v \overset{1}{\gg} \mathsf{False}()$. Then, $\exists v_2 = \mathsf{False}().\exists Y = 0.\mathsf{isOVar}\ v \longmapsto^+ \mathsf{False}() \overset{0}{\gg} \mathsf{False} \wedge Y < 1$.
19. For the workable $\mathsf{isOVar}\ w$ it is easy to show that it could never be reduced to a value in a single parallel-reduction step, so the property vacuously holds.

□[e.p.]

7.5 Permutation Lemma

Lemma 38 (Permutation) $\forall X \in \mathbb{N}.\forall w_1, w_2 \in W, e_1 \in \mathbb{E}.$

$$w_1 \overset{X}{\gg} w_2 \longmapsto e_1 \Longrightarrow (\exists e_2 \in \mathbb{E}.\, w_1 \longmapsto^+ e_2 \gg e_1).$$

Proof (Lemma 38). $\forall X \in \mathbb{N}.\forall w_1, w_2 \in W, e_1 \in \mathbb{E}.$

$$w_1 \overset{X}{\gg} w_2 \longmapsto e_1 \Longrightarrow (\exists e_2 \in \mathbb{E}.\, w_1 \longmapsto^+ e_2 \gg e_1).$$

Proof is by induction on the complexity X of derivation $w_1 \overset{X}{\gg} w_1$, and then my the size of w_1. Proof proceeds by case analysis over derivation of $w_1 \overset{X}{\gg} w_2$.

1. For the workable $(\lambda x.e)\ v$, $\dfrac{e \overset{M}{\gg} e' \quad v \overset{N}{\gg} v'}{(\lambda x.e)\ v \overset{M+\#(x,e')N+1}{\gg} e'[x := v']}$, and $e'[x := v'] \longmapsto e_1$.

   By Lemma 18, $e[x := v] \overset{M+\#(x,e')N}{\gg} e'[x := v']$. Since $M + \#(x,e')N < M + \#(x,e')N + 1$, and by monotonicity properties $e'[x := v']$ is a workable, then the induction hypothesis can be applied to $e[x := v]$ to obtain: $\exists e_2.e[x := v] \longmapsto^+ e_2 \gg e_1$. Then, it easily follows: $\exists e_3 = e_2.(\lambda x.e)\ v \longmapsto^1 e[x := v] \longmapsto^+ e_3 \gg e_1$.
2. For the workable $(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b)$, $\dfrac{e \overset{M}{\gg} e'}{(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \overset{M+1}{\gg} e'[x := \lambda x'.b[\#z := x']] \longmapsto e_1}$.

   By Lemma 18, $\exists Z.e[x := \lambda x'.b[\#z := x']] \overset{Z}{\gg} e'[x := \lambda x'.b[\#z := x']] \wedge Z \leq M$. Then, by the induction hypothesis: $\exists e_2.e[x := \lambda x'.b[\#z := x']] \longmapsto^+ e_2 \gg e_1$. Then, it immediately follows $\exists e_3 = e_2.(\lambda(\#\_ \Rightarrow x).e)\ (\#z \Rightarrow b) \longmapsto^1 e[x := \lambda x'.b[\#z := x']] \longmapsto^+ e_3 \gg e_1$.
3. For the workable $w_1\ e \overset{M+N}{\gg} w_1'\ e' \longmapsto e_1$, $\dfrac{w_1 \overset{M}{\gg} w_1' \quad e \overset{N}{\gg} e'}{w_1\ e \overset{M+N}{\gg} w_1';e'}$.

   There are five possibilities, and they all must be examined.
   (a) $w_1\ e_1 \overset{X}{\gg} w_1'\ e_1' \longmapsto e_2\ e_1'$ By definition of $\gg$, $w_1 \overset{M}{\gg} w_1'$, and $w_1' \longmapsto e_2$ Then, by the induction hypothesis: $\exists e_3.\, w_1 \longmapsto^+ e_3 \gg e_2$. Then, $\exists e_4 = e_3\ e_1.\, w_1\ e_1 \longmapsto^+ e_4 \gg e_2\ e_1'$.
   (b) $w\ e \overset{X}{\gg} v\ w' \longmapsto v\ w''$ By definition of $\gg$, $e \gg w'$, and by definition of $\longmapsto$, also $w' \longmapsto w''$. Then, by monotonicity of $\gg$, e must be a workable, and the induction hypothesis can be applied to it: $\exists e_3.\, e \longmapsto^+ e_3 \gg w''$. Since $w \gg v$, then by transition lemma: $\exists v_2.\, w \longmapsto^+ v_2 \gg v$. Then, $\exists e_4 = v_2\ e_3.\, w\ e \longmapsto^+ v_2\ e \longmapsto^+ v_2\ e_3 \gg v\ w''$.
   (c) $w\ e \overset{X}{\gg} (\lambda x.e_1)\ v \longmapsto e_1[x := v]$ There are two possibilities:
      i. $e \in \mathbb{V}$. Then, applying transition to w gives: $\exists v_2 = \lambda x.e_4.w \longmapsto^+ (\lambda x.e_4) \gg (\lambda x.e_1)$. Then, $\exists e_3 = (\lambda x.e_4)\ e.\, w\ e \longmapsto^+ (\lambda x.e_4)\ e \gg e_1[x := e]$.
      ii. $e \in \mathbb{W}$. Then, transition can be applied to both w and e, to obtain $\exists v_1 = \lambda x.e_4.\, w \longmapsto^+ \lambda x.e_4 \gg \lambda x.e_1$ and $\exists v_2.e \longmapsto^+ v_2 \gg v$. Then $\exists e_5 = v_1\ v_2.\, w\ e \longmapsto^+ v_1\ e \longmapsto^+ v_1\ v_2 \gg e[x := v]$.
   (d) $w\ e \overset{X}{\gg} (\lambda(\#z \Rightarrow x).e_1)\ (\#z \Rightarrow b) \longmapsto e_1[x := \lambda x'.b[\#z := x']]$ Similar to previous case.
   (e) $w\ e \overset{X}{\gg} (\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f)\ (k\ v) \longmapsto e[x := v]$ Similar to previous case.
4. For the workable $v\ w$, there are four cases:
   (a) $v\ w \overset{M+N}{\gg} v'\ w' \longmapsto v'\ w''$ Then, by definition of $\overset{X}{\gg}$, $v \overset{M}{\gg} v'$ and $w \overset{N}{\gg} w'$. By the induction hypothesis we have: $\exists e_2.v \longmapsto^+ e_2 \gg w''$. Then, $\exists e_3 = v\ e_2.\, v\ w \longmapsto^+ v\ e_2 \gg v'\ w''$.

(b) $(\lambda x.e_1)\ w \overset{M+N}{\gg} (\lambda x.e_1')\ v \longmapsto e_1'[x:=v]$ Since $w \gg v$, we can apply transition lemma. Thus, $\exists v_2.w \longmapsto^+ v_2 \gg v$.
Now, $\exists e_2 = (\lambda x.e_1)\ v_2.(\lambda x.e_1)\ w \longmapsto^+ (\lambda x.e_1)\ v_2 \longmapsto^1 e_1[x:=v_2]$. Then, by the subsituttion lemma: $e_1[x:=v_2] \gg e_1'[x:=v]$, since $e_1 \gg e_1'$ and $v_2 \gg v$.

(c) $(\lambda(\#z \Rightarrow x).e_1)\ (\#z \Rightarrow w) \overset{M+N}{\gg} (\lambda(\#z \Rightarrow x).e_1')\ (\#z \Rightarrow b) \longmapsto e_1'[x:=\lambda x.b[\#z:=x']]$ Similar to previous case.

(d) $(\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f)\ (k\ w) \gg (\lambda^{f \in F \cup \{k\}}\ f\ x_f.e_f')\ (k\ v) \longmapsto e_k'[x:=v]$ Similar to previous case.

5. $f\ w \overset{X}{\gg} f\ w' \longmapsto f\ e$ Then, by the definition of $\gg$, $w \overset{X}{\gg} w' \longmapsto e$.
By the induction hypothesis, $\exists e_1.\ w \longmapsto^+ e_1 \gg e$. Then, $\exists e_2 = f\ e_1.f\ w \longmapsto^+ f\ e_1 \gg f\ e$.
6. $\#z \Rightarrow w \overset{X}{\gg} \#z \Rightarrow w' \longmapsto \#z \Rightarrow e$. Then, by definition of $\gg$, $w \gg w' \longmapsto^+ e$. By the induction hypothesis $\exists e_1.w \longmapsto^+ e_1 \gg e$. Then, $\exists e_2 = \#z \Rightarrow e_1.\#z \Rightarrow w \longmapsto^+ \#z \Rightarrow e_1 \gg \#z \Rightarrow e$.
7. For the workable (w, e) there are two possibilities:
   (a) $(w,e) \overset{M+N}{\gg} (w',e') \longmapsto (w'',e')$ Then, by the induction hypothesis: $\exists e_1.\ w \longmapsto^+ e_1 \gg w''$.
   Then, $\exists e_2 = (e_1, e).(w,e) \longmapsto^+ (e_1,e) \gg (w'',e')$.
   (b) $(w,e) \overset{M+N}{\gg} (v,w') \longmapsto (v,w'')$. Then, e must be a workable, and $e \gg w' \longmapsto w''$. Thus, by the induction hypothesis $\exists e_1.e \longmapsto^+ e_1 \gg w''$. Furthermore, by the transition lemma: $\exists v_1.w \longmapsto^+ v_1 \gg v$.
   Then $\exists e_2 = (e_1, v_1).(w,e) \longmapsto^+ (v_1,e) \longmapsto^+ (v_1,e_1) \gg (v,w'')$.
8. For the workable (v,w), where $(v,w) \gg (v',w') \longmapsto (v',w'')$. By the induction hypothesis $\exists e_1.w \longmapsto^+ e_1 \gg w''$. Then $\exists e_2 = (v,e_1).(v,w) \longmapsto^+ (v,e_1) \gg (v',w'')$.
9. For the workable $w = e$, there are four possibilites:
   (a) $w = e \gg w' = e' \longmapsto w'' = e'$ Then $w \gg w' \longmapsto w''$. By the induction hypothesis: $\exists e_1.\ w \longmapsto^+ e_1 \gg w''$. Then $\exists e_2 = (e_1 = e).w = e \longmapsto^+ e_1 = e \gg w'' = e'$.
   (b) $w = e \gg \#z = w' \longmapsto \#z = e'$ By transition lemma $\exists v_1.w \longmapsto^+ v_1 \gg \#z$. Since $e \gg w'$, by transitivity properties of $\gg$, e must also be a workable, and, as $w' \longmapsto e'$, the induction hypothesis applies: $\exists e_1.e \longmapsto^+ e_1 \gg e'$. Then $\exists e_2 = (v_1 = e_1).w = e \longmapsto^+ v_1 = e \longmapsto^+ v_1 = e_1 \gg \#z = e'$.
   (c) $w = e \gg \#z = \#z \longmapsto \mathsf{True}()$ By transition lemma and properties of parallel reduction, $w \longmapsto^+ \#z$ and $e \longmapsto^+ \#z$. Then $w = e \longmapsto^+ \#z = e \longmapsto^+ \#z = \#z \gg \mathsf{True}()$.
   (d) $w = e \gg \#z = \#z' \longmapsto \mathsf{False}()$ By transition lemma and properties of parallel reduction, $w \longmapsto^+ \#z$ and $e \longmapsto^+ \#z'$. Then $w = e \longmapsto^+ \#z = e \longmapsto^+ \#z = \#z' \gg \mathsf{False}()$.
10. For the workable $\#z = w$,there are three possibilities:
    (a) $\#z = w \longmapsto \#z = w' \longmapsto \#z = w''$. By the induction hypothesis, $\exists e_1.\ w \longmapsto^+ e_1 \gg w''$. Then $\exists e_2 = (\#z = e_1).\#z = w \longmapsto^+ \#z = e_1 \gg \#z = w''$.
    (b) $\#z = w \longmapsto \#z = \#z \longmapsto \mathsf{True}()$ By the transition lemman $\exists v.w \longmapsto^+ v \gg \#z$.. But v must be $\#z$, if it parallel reduces to $\#z$. So, $\exists e_2 = (\#z = \#z).\#z = w \longmapsto^+ \#z = \#z \overset{1}{\gg} \mathsf{True}()$.
    (c) $\#z = w \longmapsto \#z = \#z' \longmapsto \mathsf{False}()$ By the transition lemman $\exists v.w \longmapsto^+ v \gg \#z'$.. But v must be $\#z'$, if it parallel reduces to $\#z'$. So, $\exists e_2 = (\#z = \#z').\#z = w \longmapsto^+ \#z = \#z' \overset{1}{\gg} \mathsf{False}()$.
11. For the workable $\#z = \#z'$, there are two possibilities:
    (a) $\#z = \#z' \overset{0}{\gg} \#z = \#z' \longmapsto \mathsf{True}()$. Obviously $\exists e_1 = \mathsf{True}().\#z = \#z' \longmapsto^+ \mathsf{True}() \gg \mathsf{True}()$.
    (b) $\#z = \#z' \overset{0}{\gg} \#z = \#z' \longmapsto \mathsf{False}()$. Obviously $\exists e_1 = \mathsf{False}().\#z = \#z' \longmapsto^+ \mathsf{False}() \gg \mathsf{False}()$.
12. For the workable, $\pi_1\ w$ there are two possibilities:
    (a) $\pi_1\ w \gg \pi_1\ w' \longmapsto \pi_1\ w''$. Then, by the induction hypothesis, $\exists e_1.\ w \longmapsto^+ e_1 \gg w''$. From this it easily follows $\exists e_2 = \pi_1\ e_1.\pi_1\ w \longmapsto^+ \pi_1\ e_1 \gg \pi_1\ w''$.
    (b) $\pi_1\ w \gg \pi_1\ (v_1,v_2) \longmapsto^+ v_1$. Then, by transition lemma, $\exists v_3.\ w \longmapsto^+ (v_3,v_4) \gg (v_1,v_2)$. Then, $\exists e_2 = \pi_1(v_3,v_4).\pi_1\ w \longmapsto^+ \pi_1\ (v_3,v_4) \gg v_1$
13. The case for π_2 is symetrical to the case above.
14. For the workable $\pi_1(v_1,v_2) \gg \pi_2(v_1',v_2') \longmapsto v_1'$. Then $v_1 \gg v_1'$.
$\exists e_2 = v_1.\pi_1\ (v_1,v_2) \longmapsto^1 v_1 \gg v_1'$.
15. For the workable $\pi_2(v_1,v_2) \gg \pi_2(v_1',v_2') \longmapsto v_2'$. Then $v_2 \gg v_2'$.
$\exists e_2 = v_2.\pi_2\ (v_1,v_2) \longmapsto^1 v_2 \gg v_2'$.
16. For the workable $\mathsf{isOVar}\ \#z$, $\mathsf{isOVar}\ \#z \overset{0}{\gg} \mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()$. Then $\exists e_2 = \mathsf{True}().\mathsf{isOVar}\ \#z \longmapsto^+ e_2 \overset{0}{\gg} \mathsf{True}()$.

17. For the workable $\mathsf{isOVar}\ v$,where $v \neq \#z$, $\mathsf{isOVar}\ v \overset{X}{\gg} \mathsf{isOVar}\ v' \longmapsto \mathsf{False}()$. Then $\exists e_2 = \mathsf{False}().\mathsf{isOVar}\ v \longmapsto^+ e_2 \overset{0}{\gg} \mathsf{False}()$.
18. For the workable $\mathsf{isOVar}\ w$, there are three possibilities:
    (a) $\mathsf{isOVar}\ w \gg \mathsf{isOVar}\ \#z \longmapsto \mathsf{True}()$. By definition of $\gg$, $w \gg \#z$. By transition $w \longmapsto^+ \#z$. Then $\exists e_2 = \mathsf{True}.\mathsf{isOVar}\ w \longmapsto^+ \mathsf{isOVar}\ \#z \longmapsto e_2 \gg \mathsf{True}()$.
    (b) $\mathsf{isOVar}\ w \gg \mathsf{isOVar}\ v \longmapsto \mathsf{False}()$, where $v \neq \#z$. By definition of $\gg$, $w \gg v$. By transition $\exists v'.\, w \longmapsto^+ v' \gg v$. Then $\exists e_2 = \mathsf{False}().\mathsf{isOVar}\ w \longmapsto^+ \mathsf{isOVar}\ v' \longmapsto e_2 \gg \mathsf{False}()$.
    (c) $\mathsf{isOVar}\ w \gg \mathsf{isOVar}\ w' \longmapsto \mathsf{isOVar}\ w''$. By definitions of $\gg$ and $\longmapsto$, $w \gg w' \longmapsto w''$. By the induction hypothesis, $\exists e.\, w \longmapsto^+ e \gg w''$. Then, $\exists e_2 = \mathsf{isOVar}\ e.\mathsf{isOVar}\ w \longmapsto^+ \mathsf{isOVar}\ e \gg \mathsf{isOVar}\ w''$.

□[e.p.]

References

1. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, revised edition, 1984.
2. Oregon Graduate Institute Technical Reports. P.O. Box 91000, Portland, OR 97291-1000,USA. Available online from ftp://cse.ogi.edu/pub/tech-reports/README.html. Last viewed August 1999.
3. G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1:125–159, 1975.
4. Walid Taha. *Multi-Stage Programming: Its Theory and Applications*. PhD thesis, Oregon Graduate Institute of Science and Technology, July 1999. Revised October 99. Available from author (taha@cs.chalmers.se).
5. Walid Taha. A sound reduction semantics for untyped CBN mutli-stage computation. Or, the theory of MetaML is non-trivial. In *2000 SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Maniplation (PEPM'00)*, January 2000.
6. Walid Taha and Tim Sheard. Multi-stage programming with explicit annotations. In *Proceedings of the ACM-SIGPLAN Symposium on Partial Evaluation and semantic based program manipulations PEPM'97, Amsterdam*, pages 203–217. ACM, 1997. An extended and revised version appears in [8].
7. Walid Taha and Tim Sheard. MetaML and multi-stage programming with explicit annotations. Technical Report CSE-99-007, Department of Computer Science, Oregon Graduate Institute, January 1999. Extended version of [6]. Available from [2].
8. Walid Taha and Tim Sheard. MetaML: Multi-stage programming with explicit annotations. *Theoretical Computer Science*, 248(1-2), 2000. In Press. Revised version of [7].
9. Masako Takahashi. Parallel reductions in λ-calculus. *Information and Computation*, 118(1):120–127, April 1995.

8 Acknowledgements

The third author would like to thank David Sands and Thiery Coquand for useful discussions that have influenced this work possitively.