Spinal Atomic Lambda-Calculus

Tom Gundersen
Red Hat, Inc.
teg@jklm.no

Willem Heijltjes
University of Bath, England, UK
http://www.cs.bath.ac.uk/~wbh22/
w.b.heijltjes@bath.ac.uk

Michel Parigot
Laboratoire PPS, UMR 7126, CNRS & Université Paris 7 (France)
michel.parigot@gmail.com

David Rhys Sherratt
Friedrich-Schiller University Jena, Germany
david.rhys.sherratt@uni-jena.de

Abstract

We present the spinal atomic lambda-calculus, a lambda-calculus with explicit sharing and atomic duplication that achieves spinal full laziness: duplicating only the direct path between a binder and bound variables is enough for beta reduction to proceed. We show how this calculus is the result of a Curry-Howard style interpretation of the deep inference proof formalism. We prove this calculus has natural properties with respect to the lambda-calculus: confluence and preservation of strongly normalising terms.

2012 ACM Subject Classification Mathematics of computing → Lambda calculus

Keywords and phrases Lambda-Calculus, Sharing, Duplication, Laziness, Scope

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

In the lambda-calculus, a main source of efficiency is *sharing*: multiple use of a single subterm, commonly expressed through graph reduction [] or explicit substitution [1]. This work, and the *atomic lambda-calculus* [15] on which it builds, is an investigation into sharing as it occurs naturally in intuitionistic *deep-inference* proof theory [23, 14].

The atomic lambda-calculus arose as a Curry–Howard interpretation of a deep-inference proof system, in particular of the *distribution* rule below left, a variant of the characteristic *medial* rule [9]. In the term calculus, the corresponding *distributor* construct enables duplication to proceed *atomically*, on individual constructors, in the style of sharing graphs []. As a consequence, the natural reduction strategy in the atomic lambda-calculus is *fully lazy* [?, ?]: it duplicates only the minimal part of a term, the *skeleton*, that can be obtained by lifting out subterms as explicit substitutions.[1]

$$\text{Distribution:}\quad \frac{A \to (B \wedge C)}{(A \to B) \wedge (A \to C)}\,d \qquad\qquad \text{Switch:}\quad \frac{(A \to B) \wedge C}{A \to (B \wedge C)}\,s$$

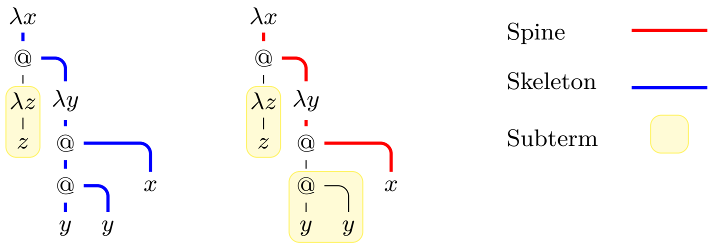
In this work, we investigate the computational interpretation of another characteristic deep-inference proof rule: the *switch* rule above right []. Our result is the *spinal atomic*

[1] While duplication is atomic *locally*, a duplicated abstraction does not form a redex until also its bound variables have been duplicated; hence duplication becomes fully lazy *globally*.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:43
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

lambda-calculus, a term calculus in which the natural strategy is a refined form of full laziness, *spine duplication*. This strategy duplicates only the *spine* of an abstraction: the paths to its bound variables in the syntax tree of the term.

We illustrate these notions below, for the example term $\lambda x.(\lambda z.z)(\lambda y.(yy)x)$. The *scope* of the abstraction λx is the entire subterm, $(\lambda z.z)(\lambda y.(yy)x)$ (which may or may not be taken to include λx itself). The *skeleton*, indicated in blue below, is the term $\lambda x.w(\lambda y.(yy)x)$ where the subterm $\lambda z.z$ is lifted out as an (explicit) substitution $[\lambda z.z/w]$. The *spine* of a term, indicated in red in the second image, cannot naturally be expressed with explicit substitution, though one can get an impression with *capturing* substitutions: it would be $\lambda x.w(\lambda y.vx)$, with the subterm yy extracted by a capturing substitution $[yy/x]$.



We identify four natural duplication regimes from the literature. For a shared term $\lambda x.N$ to become available as the function of a redex:

Laziness duplicates its *scope* [];
Full laziness duplicates its *skeleton* [**?**, **?**];
Spinal full laziness duplicates its *spine* [**?**, **?**];
Optimal reduction duplicates just the abstraction λx and its bound variables x [**?**, **?**].

We investigate the computational meaning of the following *switch* rule of deep-inference proof theory [23, 14]:

$$\frac{(A \to B) \wedge C}{A \to (B \wedge C)}\, s$$

On its own, it corresponds to an *end-of-scope* marker in λ-calculus. This is a special annotation of a subterm, to indicate that a given variable does not occur free, so that a substitution on that variable can be aborted early. In the above rule, A corresponds to the binding variable of an abstraction and C to the subterm of said abstraction where it doesn't occur, while B represents those subterms where it does occur.

The main thrust of our work is to incorporate this rule, and its computational interpretation as a term construct, into the *atomic* λ*-calculus* [15]. This calculus results from an investigation of the following *medial* rule:

$$\frac{(A \vee B) \to (C \wedge D)}{(A \to C) \wedge (B \to D)}\, m$$

The medial rule enables duplication to proceed *atomically*: on individual constructors (abstraction and application) rather than entire subterms. The atomic λ-calculus implements *full laziness*, a standard notion of sharing where only the *skeleton* of a term needs to be duplicated. Given a term t which needs to be duplicated, full laziness allows to share all maximal subterms $u_1, \ldots, u_k$ of t that do not contain occurrences of a variable bound in t outside u_i. The constructors in t not in any u_i are then part of the skeleton.

Our investigation is then focused on the interaction of switch and medial. Based on this we develop the *spinal atomic* λ*-calculus*, a natural evolution of the atomic λ-calculus.

The new calculus improves on full laziness with *spinal full laziness*, which duplicates the *spine* rather than the skeleton. The spine of an abstraction are the direct paths from the binder to bound variables [3]. The graph below provides an example of this for the term $\lambda x.(\lambda z.z)\lambda y.(yy)x$, where the spine of λx is the very thick red line and the largest subterms that could be identied by an end-of-scope operator in the term calculus (or the switch rule in the proof theory) are enclosed by yellow boxes.
In Section 2 introduces a simple sharing calculus that we expand on in Section 3, where we introduce the syntax and semantics of the spinal atomic λ-calculus, and its typing system. In Section 4 we further study the reduction rules that allow for spinal duplication, and prove natural properties of these rules such as termination and confluence. In Section 5 we extend these results to include beta reduction, and show preservation of strong normalisation with respect to the λ-calculus. We conclude in Section 6.

1.1 Related Work

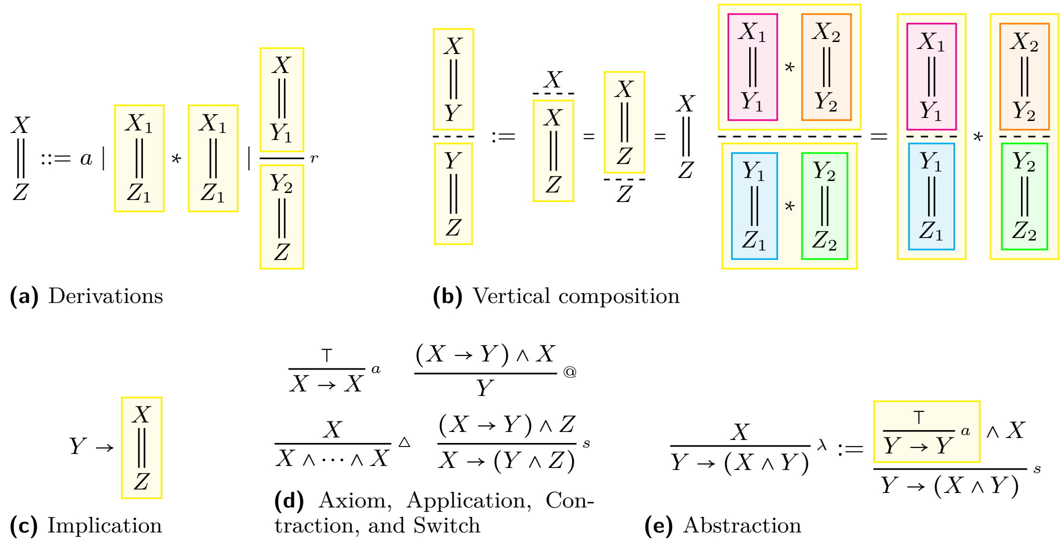
Spine duplication has been implemented by Blanc et al. in [7], by making use of labels in a dag-implementation. The main purpose of their work is to study sharing in Wadsworth's *weak λ-calculus* [25] (further studied in [10]). Balabonski [3] showed that spine duplication allows for an optimal reduction in the sense of Lévy [21] for *weak reduction* i.e. where a β-reduction $(\lambda x.t)s$ occurring in a subterm u can only reduce if all free variables in the redex are also free in the term u. Blelloch and Greiner [8] showed that the weak call-by-value reduction strategy can be implemented in polynomial time with respect to the size of the initial term and the number of β steps in said term. Given the restriction that u is a closed term, this is then the same as *closed reduction* [11, 12]. Our work generalizes spine duplication to the λ-calculus. It uses environments to implement sharing, and does not make use of labels, while maintaining a close intuition to dag-implementations.

End-of-scope markers in the λ-calculus have been seen throughout literature. *Berkling's lambda bar* [5] has shown to remove the need for variable names while maintaining correctness; improving efficiency by removing the need for alpha conversion [6]. This result was generalized by *Adbmal* (invert of "Lambda") [17]. It was shown by using multiple variables, scopes can be sequenced rather than nested which correspond closely to the boxes in MELL proof nets of linear logic [20]. This approach was studied further in [24] as graph reduction that satisfies optimality [21]. Although these approaches could identify the skeleton of a term, none however identify the spine of a term, which meant the scopes explicitly displayed may be larger than necessary from the perspective of performing substitution. This problem was solved by *director strings*, introduced by Kennaway and Sleep in [18] for combinator reduction and then generalized for any strategy by Fernández et al. in [13]. Director strings are an annotation on terms detailing the location of variable occurrences. An apt annotation on the body of an abstraction will consequently identify the spine of it. Despite being implementations that use director strings [22, 13, 12], an implementation with sharing techniques allowing for duplicating solely the spine could not be found.

2 Typing a λ-calculus in open deduction

A *derivation* from a *premise* formula X to a *conclusion* formula Z is constructed inductively as in Figure 1a, with from left to right: a propositional atom a, where $X = Z = a$; *horizontal composition* with a connective $*$, where $X = X_1 * X_2$ and $Z = Z_1 * Z_2$; and *rule composition*, where r is an inference rule from Y_1 to Y_2. The boxes serve as parentheses (since derivations extend in two dimensions) and may be omitted. Derivations are considered up to associativity of rule composition. One may consider formulas as derivations that omit rule composition; and the binary $*$ may be generalised to 0-ary, unary, and n-ary operators. *Vertical composition* of a derivation from X to Y and one from Y to Z, depicted by a dashed line, is a defined operation, given in Figure 1b.

A system for intuitionistic logic is given by the binary connectives $\rightarrow$, $\wedge$, and nullary connective $\top$, where we restrict implication to a form in Figure 1c, and the inference rules in Figure 1d. We work modulo associativity, symmetry, and unitality of conjunction, justifying the n-ary contraction, and may omit $\top$ from the axiom rule. A 0-ary contraction, with conclusion $\top$, is a *weakening*. Figure 1e: the abstraction rule (λ) is derived from axiom and switch.



(a) Derivations **(b)** Vertical composition **(c)** Implication **(d)** Axiom, Application, Contraction, and Switch **(e)** Abstraction

Figure 1 Intuitionistic Proof System in Open Deduction

2.1 The Sharing Calculus

Our starting point is the *sharing calculus* (Λ^S), a calculus with an explicit sharing construct, similar to explicit substitution [1].

▶ **Definition 1.** *The* pre-terms r, s, t *and* sharings $[\Gamma]$ *of the* Λ^S *are defined by:*

$$s,t \quad ::= \quad x \quad | \quad \lambda x.t \quad | \quad s\,t \quad | \quad u[\Gamma] \qquad [\Gamma] \quad ::= \quad [x_1,\dots,x_n \leftarrow s]$$

with from left to right: a variable*; an* abstraction*, where x occurs free in t and becomes bound; an* application*, where t and s use distinct variable names; and a* closure*; in $u[\vec{x} \leftarrow s]$ the variables in the vector $\vec{x} = x_1,\dots,x_n$ all occur in t and become bound, and t and s use distinct variable names.* Terms *are pre-terms modulo* permutation *equivalence $(\sim)$:*

$$t[\vec{x} \leftarrow s][\vec{y} \leftarrow r] \sim t[\vec{y} \leftarrow r][\vec{x} \leftarrow s] \qquad (\{\vec{y}\} \cap (s)_{fv} = \{\})$$

A term is in sharing normal form *if all sharings occur as $[\vec{x} \leftarrow x]$ either at the top level or directly under a binding abstraction, as $\lambda x.t[\vec{x} \leftarrow x]$.*

Note that variables are *linear*: variables occur at most once, and bound variables must occur. A vector $\vec{x}$ has length $|\vec{x}|$ and consist of the variables $x_1,\dots,x_{|\vec{x}|}$. An *environment* is a sequence of sharings $\overline{[\Gamma]} = [\Gamma_1]\dots[\Gamma_n]$. Substitution is written $\{x/t\}$, and $\{t_1/x_1\}\dots\{t_n/x_n\}$ may be abbreviated to $\{t_i/x_i\}_{i\in[n]}$.

▶ **Definition 2.** *The* interpretation *of a term t to the λ-term $[\![\,t\,]\!]$ given as follows*

$$[\![\,x\,]\!] = x \quad [\![\,\lambda x.t\,]\!] = \lambda x.[\![\,t\,]\!] \quad [\![\,s\,t\,]\!] = [\![\,s\,]\!]\,[\![\,t\,]\!] \quad [\![\,t[\vec{x} \leftarrow s]\,]\!] = [\![\,t\,]\!]\{[\![\,s\,]\!]/x_i\}_{i\in[n]}$$

The translation $(\!|\,N\,|\!)$ *of a λ-term N is the unique sharing-normal term t such that $N = [\![\,t\,]\!]$.*

A term t will be typed by a derivation with restricted types, as shown below, where the *context type* $\Gamma = A_1 \wedge \dots \wedge A_n$ will have an A_i for each free variable x_i of t. We connect free variables to their premises by writing A^x and $\Gamma^{\vec{x}}$. The Λ^S is then typed as in Figure 2.

Basic Types: $A, B, C \;\; := \;\; a \mid A \rightarrow B$ Context Types: $\Gamma, \Delta, \Omega \;\; := \;\; A \mid \top \mid \Gamma \wedge \Delta$

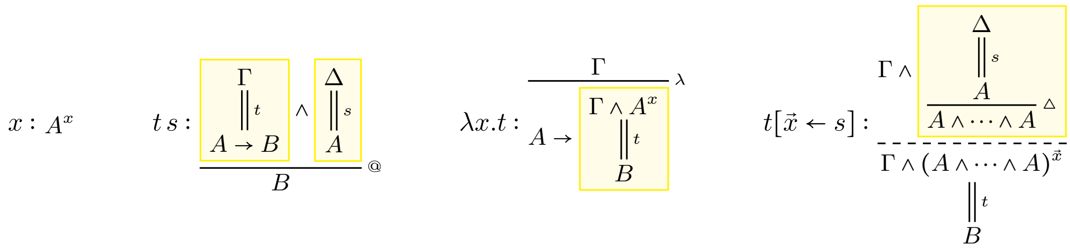


Figure 2 Typing System for Λ^S

3 The Spinal Atomic λ-Calculus

We now formally introduce the syntax of the spinal atomic λ-calculus (Λ^S_a), by extending the definition of the sharing calculus in Definition 1 with a *distributor* construct that allows for atomic duplication of terms.

▶ **Definition 3** (Pre-Terms)**.** *The* pre-terms r, s, t, closures $[\Gamma]$, *and* environments $\overline{[\Gamma]}$ *of the* Λ^S_a *are defined by:*

$$t \;\;::=\;\; x \;\mid\; s\,t \;\mid\; x\langle\,\vec{y}\,\rangle.t \;\mid\; t[\Gamma]$$

$$[\Gamma] \;\;::=\;\; [\vec{x} \leftarrow t] \;\mid\; [e_1\langle\,\vec{x}_1\,\rangle\dots e_n\langle\,\vec{x}_n\,\rangle \,|\, d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}] \qquad \overline{[\Gamma]} \;\;::=\;\; [\Gamma] \;\mid\; \overline{[\Gamma]}[\Gamma]$$

First note that we denote abstractions such that $\lambda x.t \equiv x\langle\,x\,\rangle.t$. We introduce a new notion of abstraction called *phantom-abstraction*, which can be intuitively be thought of as a partially duplicated abstraction. An abstraction $x\langle\,x\,\rangle.t$ and a phantom-abstraction $x\langle\,\vec{y}\,\rangle.t$ are two instances of the same construct. We call the variables inside the brackets the *cover* of the abstraction. If the cover is the same as the preceeding variable, then it is an abstraction, otherwise it is a phantom-abstraction and we call the preceeding variable a *phantom-variable*.

The distributor $u[e_1\langle\,\vec{x}_1\,\rangle\dots e_n\langle\,\vec{x}_n\,\rangle \,|\, d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}]$ captures the phantom-variables $e_1, \dots, e_n$ in u and the covers associated with those phantom-variables are captured by the environment $\overline{[\Gamma]}$. We sometimes write the distributor as $u[\overrightarrow{e\langle\,x\,\rangle} \,|\, d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}]$ when we are not concerned about the binding of phantom-variables. Terms are then pre-terms with sensible and correct bindings. To define terms, we first define *free* and *bound* variables and phantom variables.

▶ **Definition 4** (Free and Bound Variables). *The free variables $(-)_{fv}$ and bound variables $(-)_{bv}$ of a pre-term t is defined as follows*

$$
\begin{aligned}
(x)_{fv} &= \{x\} & (x)_{bv} &= \{\} \\
(s\,t)_{fv} &= (s)_{fv} \cup (t)_{fv} & (s\,t)_{bv} &= (s)_{bv} \cup (t)_{bv} \\
(x\langle\, x\,\rangle.t)_{fv} &= (t)_{fv} - \{x\} & (x\langle\, x\,\rangle.t)_{bv} &= (t)_{bv} \cup \{x\} \\
(c\langle\, \vec{x}\,\rangle.t)_{fv} &= (t)_{fv} & (c\langle\, \vec{x}\,\rangle.t)_{bv} &= (t)_{bv} \\
(u[\vec{x} \leftarrow t])_{fv} &= (u)_{fv} \cup (t)_{fv} - \{\vec{x}\} & (u[\vec{x} \leftarrow t])_{bv} &= (u)_{bv} \cup (t)_{bv} \cup \{\vec{x}\} \\
(u[\overrightarrow{e\langle\, x\,\rangle} \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}])_{fv} &= (u\overline{[\Gamma]})_{fv} - \{c\} & (u[\overrightarrow{e\langle\, x\,\rangle} \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}])_{bv} &= (u\overline{[\Gamma]})_{bv} \\
(u[\overrightarrow{e\langle\, x\,\rangle} \,|\, c\langle\, \vec{y}\,\rangle\, \overline{[\Gamma]}])_{fv} &= (u\overline{[\Gamma]})_{fv} \cup \{c\} & (u[\overrightarrow{e\langle\, x\,\rangle} \,|\, c\langle\, \vec{y}\,\rangle\, \overline{[\Gamma]}])_{bv} &= (u\overline{[\Gamma]})_{bv}
\end{aligned}
$$

▶ **Definition 5** (Free and Bound Phantom-Variables). *The free phantom-variables $(-)_{fp}$ and bound phantom-variables $(-)_{bp}$ of the pre-term t is defined as follows*

$$
\begin{aligned}
(x)_{fp} &= \{\} & (x)_{bp} &= \{\} \\
(s\,t)_{fp} &= (s)_{fp} \cup (t)_{fp} & (s\,t)_{bp} &= (s)_{bp} \cup (t)_{bp} \\
(x\langle\, x\,\rangle.t)_{fp} &= (t)_{fp} & (x\langle\, x\,\rangle.t)_{bp} &= (t)_{bp} \\
(c\langle\, \vec{x}\,\rangle.t)_{fp} &= (t)_{fp} \cup \{c\} & (c\langle\, \vec{x}\,\rangle.t)_{bp} &= (t)_{bp} \\
(u[\vec{x} \leftarrow t])_{fp} &= (u)_{fp} \cup (t)_{fp} & (u[\vec{x} \leftarrow t])_{bp} &= (u)_{bp} \cup (t)_{bp}
\end{aligned}
$$

$$
\begin{aligned}
(u[e_1\langle\, \vec{x_1}\,\rangle \dots e_n\langle\, \vec{x_n}\,\rangle \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}])_{fp} &= (u\overline{[\Gamma]})_{fp} - \{e_1, \dots, e_n\} \\
(u[e_1\langle\, \vec{x_1}\,\rangle \dots e_n\langle\, \vec{x_n}\,\rangle \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}])_{bp} &= (u\overline{[\Gamma]})_{bp} \cup \{e_1, \dots, e_n\} \\
(u[e_1\langle\, \vec{x_1}\,\rangle \dots e_n\langle\, \vec{x_n}\,\rangle \,|\, c\langle\, \vec{y}\,\rangle\, \overline{[\Gamma]}])_{fp} &= (u\overline{[\Gamma]})_{fp} \cup \{c\} - \{e_1, \dots, e_n\} \\
(u[e_1\langle\, \vec{x_1}\,\rangle \dots e_n\langle\, \vec{x_n}\,\rangle \,|\, c\langle\, \vec{y}\,\rangle\, \overline{[\Gamma]}])_{bp} &= (u\overline{[\Gamma]})_{bp} \cup \{e_1, \dots, e_n\}
\end{aligned}
$$

Variables are bound by abstractions (not phantoms) and sharings. Phantom-variables are bound by distributors. With these definitions, we can formally define the terms of Λ_a^S.

▶ **Definition 6** (Terms). *A* term $t \in \Lambda_a^S$ *is a pre-term with the following constraints*

1. *Each variable may occur at most once.*
2. *In an abstraction $x\langle\, x\,\rangle.t$, $x \in (t)_{fv}$.*
3. *In a phantom-abstraction $c\langle\, x_1, \dots, x_n\,\rangle.t$, $\{x_1, \dots, x_n\} \subset (t)_{fv}$.*
4. *In a sharing $u[x_1, \dots, x_n \leftarrow t]$, $\{x_1, \dots, x_n\} \subset (u)_{fv}$.*
5. *In a distributor $u[e_1\langle\, w_1^1, \dots, w_{k_1}^1\,\rangle \dots e_n\langle\, w_1^n, \dots, w_{k_n}^n\,\rangle \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}]$*
   a. *For all $1 \le i \le n$ and $1 \le m \le k_n$, $w_m^i(u)_{fv}$ and becomes bound by $\overline{[\Gamma]}$.*
   b. *$\{e_1\langle\, w_1^1, \dots, w_{k_1}^1\,\rangle, \dots, e_n\langle\, w_1^n, \dots, w_{k_n}^n\,\rangle\} \subset (u)_{fc}$, and $\{e_1, \dots, e_n\} \subset (u)_{fp}$, and each e_i becomes bound.*
   c. *The variable c occurs somewhere in the environments $\overline{[\Gamma]}$.*
6. *In a distributor $u[e_1\langle\, w_1^1, \dots, w_{k_1}^1\,\rangle \dots e_n\langle\, w_1^n, \dots, w_{k_n}^n\,\rangle \,|\, c\langle\, y_1, \dots, y_m\,\rangle\, \overline{[\Gamma]}]$*
   a. *Both $5(a)$ and $5(b)$ hold.*
   b. *For all $1 \le i \le m$, y_i occurs in the environments $\overline{[\Gamma]}$.*

We also work modulo permutation with respect to the variables in the cover of phantom-abstractions. Let $\vec{x}$ be a list of variables and let $\vec{x_P}$ be a permutation of that list, then the following terms are considered equal.

$$u[\vec{x} \leftarrow t] \sim u[\vec{x_P} \leftarrow t] \qquad\qquad c\langle\, \vec{x}\,\rangle.t \sim c\langle\, \vec{x_P}\,\rangle.t$$

Terms are typed with the typing system for Λ^S extended with the *distribution* inference rule.

$$\frac{A \to (B_1 \wedge \cdots \wedge B_n)}{(A \to B_1) \wedge \cdots \wedge (A \to B_n)}\, d$$

This rule is the result of computationally interpreting the medial rule as done in [15]. We obtain this variant of the medial rule due to the restriction for implications and to avoid introducing disjunction to the typing system. The terms of Λ^S_a are then typed as in both Figure 2 and Figure 3. Note environments are typed by the derivations of all its closures composed horizontally with the conjunction connective.

$$c\langle\, \vec{x}\,\rangle.t : \dfrac{(A \to \Gamma) \wedge \Delta}{A^c \to \boxed{\begin{array}{c}\Gamma^{\vec{x}} \wedge \Delta \\ \Vert\, t \\ C\end{array}}}\, s \qquad\qquad u[\overrightarrow{e\langle\, x\,\rangle} \,|\, c\langle\, \vec{z}\,\rangle\, \overline{[\Gamma]}] : \dfrac{\boxed{\dfrac{\dfrac{(C \to \Gamma) \wedge \Delta}{C^c \to \boxed{\begin{array}{c}\Gamma^{\vec{z}} \wedge \Delta \\ \Vert\, \overline{[\Gamma]} \\ \Sigma_1 \wedge \cdots \wedge \Sigma_n\end{array}}}\, s}{(C^{e_1} \to \Sigma_1^{\vec{x_1}}) \wedge \cdots \wedge (C^{e_n} \to \Sigma_n^{\vec{x_n}})}\, d} \wedge \Omega}{\begin{array}{c}(C \to \Sigma_1) \wedge \cdots \wedge (C \to \Sigma_n) \wedge \Omega \\ \Vert\, u \\ E\end{array}}$$

Figure 3 Typing derivations for phantom-abstractions and distributors

3.1 Compilation and Readback

We now define the translations between Λ^S_a and the original λ-calculus. First we define the interpretation $\Lambda \to \Lambda^S_a$ (*compilation*). Intuitively, it replaces each abstraction $\lambda x.-$ with the term $x\langle\, x\,\rangle. - [x_1, \ldots, x_n \leftarrow x]$ where $x_1, \ldots, x_n$ replace the occurrences of x. Actual substitutions are denoted as $\{t/x\}$. Let $|\,M\,|_x$ denote the number of occurrences of x in M, and if $|\,M\,|_x = n$ let $M\frac{n}{x}$ denote M with the occurrences of x by fresh, distinct variables $x^1, \ldots, x^n$. First, the translation of a *closed* term M is $(\!|\, M\,|\!)'$, defined below

▶ **Definition 7** (Compilation). *The interpretation for closed lambda terms, $(\!|\,\Lambda\,|\!)' : \Lambda \to \Lambda^S_a$ is defined below*

$$\begin{aligned}
(\!|\, x\,|\!)' &= x \\
(\!|\, M\, N\,|\!)' &= (\!|\, M\,|\!)'\, (\!|\, N\,|\!)' \\
(\!|\, \lambda x.M\,|\!)' &= \begin{cases} x\langle\, x\,\rangle.(\!|\, M\,|\!)' & \textit{if } |\,M\,|_x = 1 \\ x\langle\, x\,\rangle.(\!|\, M\frac{n}{x}\,|\!)'[x^1, \ldots, x^n \leftarrow x] & \textit{if } |\,M\,|_x = n \neq 1 \end{cases}
\end{aligned}$$

For an arbitrary term M, if $x_1, \ldots, x_k$ are the free variables of M such that $|\,M\,|_{x_i} = n_i > 1$, the translation $(\!|\, M\,|\!)$ is

$$(\!|\, M\frac{n_1}{x_1} \ldots \frac{n_k}{x_k}\,|\!)'[x_1^1, \ldots, x_1^{n_1} \leftarrow x_1] \ldots [x_k^1, \ldots, x_k^{n_k} \leftarrow x_k]$$

The readback into the λ-calculus is slightly more complicated, specifically due to the bindings induced by the distributor. Interpreting a distributor $[\![\, u[e_1\langle\, \vec{x_1}\,\rangle \ldots e_n\langle\, \vec{x_n}\,\rangle \,|\, c\langle\, c\,\rangle\, \overline{[\Gamma]}]\,]\!]$ construct as a λ-term requires (1) converting the phantom-abstractions it binds in u into abstractions (2) collapsing the environment (3) maintaining the bindings between the converted abstractions and the intended variables located in the environment.

▶ **Definition 8.** *Given a total function σ with domain D and codomain C, we* overwrite *the function with case $x \mapsto V$ where $x \in D$ and $V \in C$ such that*

$$\sigma[x \mapsto V](z) = \begin{cases} V & z = x \\ \sigma(z) & \textit{otherwise} \end{cases}$$

When using the map σ as part of the translation, the intuition is that for all bound variables x in the term we are translatings, it should be that $\sigma(x) = x$. The map $\gamma : V \to V$ is defined similarly, and the purpose is to keep track of the binding of phantom-variables.

▶ **Definition 9.** *The interpretation $\llbracket - \mid - \mid - \rrbracket : \Lambda_a^S \times (V \to \Lambda) \times (V \to V) \to \Lambda$ is defined as*

$$\llbracket x \mid \sigma \mid \gamma \rrbracket = \sigma(x)$$

$$\llbracket s\, t \mid \sigma \mid \gamma \rrbracket = \llbracket s \mid \sigma \mid \gamma \rrbracket\, \llbracket t \mid \sigma \mid \gamma \rrbracket$$

$$\llbracket c\langle\, c\,\rangle.t \mid \sigma \mid \gamma \rrbracket = \lambda c.\llbracket t \mid \sigma[c \mapsto c] \mid \gamma \rrbracket$$

$$\llbracket c\langle\, x_1, \ldots, x_n\,\rangle.t \mid \sigma \mid \gamma \rrbracket = \lambda c.\llbracket t \mid \sigma[x_i \mapsto \sigma(x_i)\{c/\gamma(c)\}]_{i\in[n]} \mid \gamma \rrbracket$$

$$\llbracket u[x_1, \ldots, x_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i\in[n]} \mid \gamma \rrbracket$$

$$\llbracket u[e_1\langle\, \vec{w}_1\,\rangle, \ldots, e_n\langle\, \vec{w}_n\,\rangle \mid c\langle\, c\,\rangle\, \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket = \llbracket u\overline{[\Gamma]} \mid \sigma \mid \gamma[e_i \mapsto c]_{i\in[n]} \rrbracket$$

$$\llbracket u[e_1\langle\, \vec{w}_1\,\rangle, \ldots, e_n\langle\, \vec{w}_n\,\rangle \mid c\langle\, x_1, \ldots, x_m\,\rangle\, \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket = \llbracket u\overline{[\Gamma]} \mid \sigma' \mid \gamma[e_i \mapsto c]_{i\in[n]} \rrbracket$$
$$\textit{where } \sigma' = \sigma[x_i \mapsto \sigma(x_i)\{c/\gamma(c)\}]_{i\in[n]}$$

The following Proposition justifies working modulo permutation equivalence.

▶ **Proposition 10.** *For $s, t \in \Lambda_a^S$, if $s \sim t$ then $\llbracket s \rrbracket = \llbracket t \rrbracket$*

The following Lemma not only proves we have good translations, but is also important for proving confluence of Λ_a^S (Theorem 34).

▶ **Lemma 11.** *For a closed $t \in \Lambda_a^S$, in sharing normal form, and a closed $N \in \Lambda$.*

$$\llbracket (\!| N |\!)' \rrbracket = N \qquad (\!| \llbracket t \rrbracket |\!)' = t \qquad \exists_{M\in\Lambda}.t = (\!| M |\!)'$$

3.2 Rewrite Rules

Both the spinal atomic λ-calculus and the atomic λ-calculus of [15] follow atomic reduction steps, i.e. they apply on individual constructors. The biggest difference is that our calculus is capable of duplicating not only the skeleton but also the spine. The rewrite rules in our calculus make use of 3 operations, *substitution*, *book-keeping*, and *exorcism*.

The operation *substitution* $t\{s/x\}$ propagates through the term t, and replaces the free occurences of the variable x with the term s. Moreover, if x occurs in the cover of a phantom-variable $e\langle\, \vec{y} \cdot x\,\rangle$, then substitution replaces the x in the cover with $(s)_{fv}$, $e\langle\, \vec{y} \cdot (s)_{fv}\,\rangle$.

Although substitution performs some book-keeping on phantom-abstractions, we define an explicit notion of book-keeping $\{\vec{y}/e\}_b$ that updates the variables stored in a free cover i.e. for a term t, $e\langle\, \vec{x}\,\rangle \in (t)_{fc}$ then $e\langle\, \vec{y}\,\rangle \in (t\{\vec{y}/e\}_b)_{fc}$.

The last operation we introduce is called *exorcism* $\{c\langle\, \vec{x}\,\rangle\}_e$. We perform exorcisms on phantom-abstractions to convert them to abstractions. Intuitively, this will be performed on phantom-abstractions with phantom-variables bound to a distributor when said distributor is eliminated. It converts phantom-abstractions to abstractions by introducing a sharing of the phantom-variable that captures the variables in the cover, i.e. $c\langle\, \vec{x}\,\rangle.t\{c\langle\, \vec{x}\,\rangle\}_e = c\langle\, c\,\rangle.t[\vec{x} \leftarrow c]$.

▶ **Proposition 12.** *Given $M \in \Lambda$ such that for all $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$, the translation $[\![\, u \,|\, \sigma \,|\, \gamma \,]\!]$ commutes with substitution $\{M/x\}$ in the following way*

$$[\![\, u\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma[x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]] \,|\, \gamma \,]\!]$$

▶ **Proposition 13.** *Book-keeping commutes with the translation in the following way*
if $c\langle\, y_1, \ldots, y_m \,\rangle. \in (u)_{fc}$ such that $\{x_1, \ldots, x_n\} \subset \{y_1, \ldots, y_m\}$
and for those $z \in \{y_1, \ldots, y_m\}/\{x_1, \ldots, x_n\}$, $\gamma(c) \notin (\sigma(z))_{fv}$
or if simply $\{x_1, \ldots, x_n\} \cap (u)_{fv} = \{\}$

$$[\![\, u\{x_1, \ldots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma \,|\, \gamma \,]\!]$$

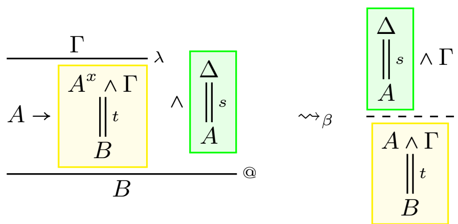
▶ **Proposition 14.** *Exorcisms commute with the translation in the following way*
if $c\langle\, x_1, \ldots, x_n \,\rangle. \in (u)_{fc}$ or $\{x_1, \ldots, x_n\} \cap (u)_{fv} = \{\}$

$$[\![\, u\{c\langle\, x_1, \ldots, x_n \,\rangle\}_e \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma[x_i \mapsto c]_{i \in [n]} \,|\, \gamma \,]\!]$$

Using these operations, we define the rewrite rules that allow for spinal duplication. Firstly we have beta reduction ($\rightsquigarrow_\beta$), which strictly requires an abstraction (not a phantom).

$$(x\langle\, x \,\rangle.t)\, s \rightsquigarrow_\beta t\{s/x\} \qquad (\beta)$$

However, its effect is very different: here β-reduction is a linear operation, since the bound variable x occurs exactly once in the body t. Any duplication of the term t in the atomic lambda-calculus proceeds via the sharing reductions. The proof rewrite step that corrersponds to β-reduction is shown below.



The first set of sharing reduction rules move closures towards the outside of a term. Most of these rewrite rules only change the typing derivations in the way that subderivations are composed, with the exception of moving a closure out of scope of a distributor.

$$s[\Gamma]\, t \rightsquigarrow_L (s\, t)[\Gamma] \qquad (l_1)$$

$$s\, t[\Gamma] \rightsquigarrow_L (s\, t)[\Gamma] \qquad (l_2)$$

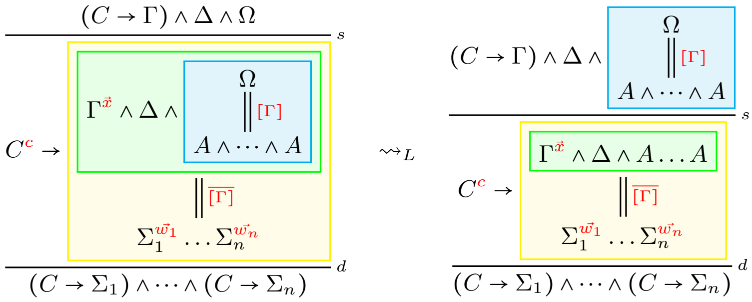
$$d\langle\, \vec{x} \,\rangle.t[\Gamma] \rightsquigarrow_L (d\langle\, \vec{x} \,\rangle.t)[\Gamma] \text{ if } \{\vec{x}\} \cap (t)_{fv} = \{\vec{x}\} \qquad (l_3)$$

$$u[\vec{x} \leftarrow t[\Gamma]] \rightsquigarrow_L u[\vec{x} \leftarrow t][\Gamma] \qquad (l_4)$$

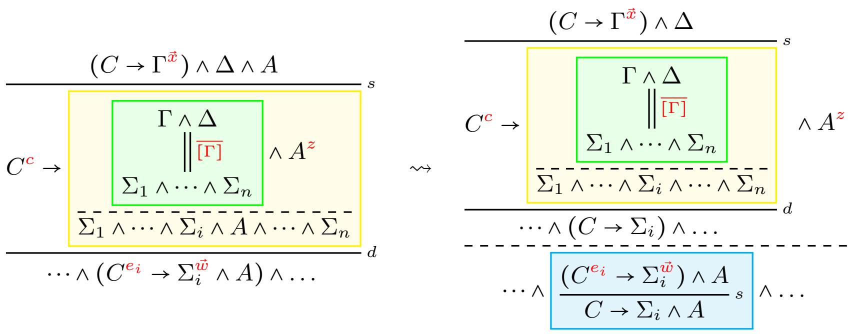
For the case of lifting a closure outside a distributor, we use a notation $\|\, [\Gamma] \,\|$ to identify the variables captured by a closure, i.e.$\|\, [\vec{x} \leftarrow t] \,\| = \{\vec{x}\}$ and $\|\, [e_1\langle\, \vec{x_1} \,\rangle, \ldots, e_n\langle\, \vec{x_x} \,\rangle \,|\, c\langle\, c \,\rangle\, \overline{[\Gamma]}] \,\| = \{\vec{x_1}, \ldots, \vec{x_n}\}$. Then let $\{\vec{z}\} = \|\, [\Gamma] \,\|$ in the following rewrite rule, that can only occur if $\{\vec{x}\} \cap ([\Gamma])_{fv} = \{\}$.

$$\begin{aligned} &u[e_1\langle\, \vec{w_1} \,\rangle \ldots e_n\langle\, \vec{w_n} \,\rangle \,|\, c\langle\, \vec{x} \,\rangle\, \overline{[\Gamma]}[\Gamma]] \\ &\quad \rightsquigarrow_L u\{(\vec{w_i}/\vec{z})/e_i\}_{b\, i \in [n]}[e_1\langle\, \vec{w_1}/\vec{z} \,\rangle \ldots e_n\langle\, \vec{w_n}/\vec{z} \,\rangle \,|\, c\langle\, \vec{x} \,\rangle\, \overline{[\Gamma]}][\Gamma] \end{aligned} \qquad (l_5)$$

The proof rewrite rule corresponding with the rewrite rule l_5 can be broken down into two parts. The first part is readjusting how the derivations compose as shown below.



The second part of the rewrite rule justifies the need for the book-keeping operation. In the rewrite below, let A be the type of a variable z where $z \in \vec{z}$. After lifting, we want to remove the variable from the cover as to ensure correctness since the variables in the cover denote the variables captured by the environment. Book-keeping allows us to remove these variables simultaneously.



The lifting rules (l_i) are justified by the need to lift closures out of the distributor, as opposed to duplicating them. The second set of rewrite rules, consecutive sharings are compounded and unary sharings are applied as substitutions. For simplicity, in the equivalent proof rewrite step we only show the binary case for each rule.

$$u[w_1, \ldots, w_m \leftarrow y_i][y_1, \ldots, y_n \leftarrow t] \rightsquigarrow_C u[y_1, \ldots, y_{i-1}, w_1, \ldots, w_m, y_{i+1}, \ldots, y_n \leftarrow t] \qquad (c_1)$$

$$u[x \leftarrow t] \rightsquigarrow_C u\{t/x\} \qquad (c_2)$$

$$\cfrac{A}{A \wedge \boxed{\cfrac{A}{A \wedge A}\,\triangle}}\,\triangle \quad \rightsquigarrow_C \quad \cfrac{A}{A \wedge A \wedge A}\,\triangle \qquad\qquad \cfrac{A}{A}\,\triangle \quad \rightsquigarrow_C \quad A$$

The atomic steps for duplicating are given in the third and final set of rewrite rules. The first being the atomic duplication step of an application, which is the same rule used in [15]. The binary case proof rewrite steps for each rule are also provided.

$$u[x_1 \ldots x_n \leftarrow s\,t] \rightsquigarrow_D u\{z_1\,y_1/x_1\} \ldots \{z_n\,y_n/x_n\}[z_1 \ldots z_n \leftarrow s][y_1 \ldots y_n \leftarrow t] \qquad (d_1)$$

$$\cfrac{\cfrac{(A \to B) \wedge A}{B}\,@}{B \wedge B}\,\triangle \quad \rightsquigarrow_D \quad \begin{array}{c} \cfrac{(A \to B)}{(A \to B) \wedge (A \to B)}\,\triangle \wedge \cfrac{B}{B \wedge B}\,\triangle \\ \hdashline \cfrac{(A \to B) \wedge A}{B}\,@ \wedge \cfrac{(A \to B) \wedge A}{B}\,@ \end{array}$$

$$\begin{array}{l} u[x_1, \ldots, x_n \leftarrow c\langle\, \vec{y}\,\rangle.t] \rightsquigarrow_D \\ \quad u\{e_i\langle\, w_i\,\rangle.w_i/x_i\}_{i \in [n]}[e_1\langle\, w_1\,\rangle \ldots e_n\langle\, w_n\,\rangle \,|\, c\langle\, \vec{y}\,\rangle\,[w_1, \ldots, w_n \leftarrow t]] \end{array} \qquad (d_2)$$

$$\dfrac{\dfrac{(A\to B)\wedge\Gamma}{A\to\boxed{\begin{array}{c}B\wedge\Gamma\\ \|\\ C\end{array}}}\,s}{(A\to C)\wedge(A\to C)}\,\triangle \quad\rightsquigarrow_D\quad \dfrac{\dfrac{(A\to B)\wedge\Gamma}{A\to\boxed{\begin{array}{c}B\wedge\Gamma\\ \|\\ \dfrac{C}{C\wedge C}\,\triangle\end{array}}}\,s}{(A\to C)\wedge(A\to C)}\,d$$

$$u[e_1\langle\,\vec{w}_1\,\rangle\ldots e_n\langle\,\vec{w}_n\,\rangle\,|\,c\langle\,c\,\rangle\,[\vec{w}_1,\ldots,\vec{w}_n\leftarrow c]]\rightsquigarrow_D u\{e_1\langle\,\vec{w}_1\,\rangle\}_e\ldots\{e_n\langle\,\vec{w}_n\,\rangle\}_e \qquad (d_3)$$

$$\dfrac{\dfrac{}{A\to\dfrac{A}{A\wedge A}\,\triangle}\,a}{(A\to A)\wedge(A\to A)}\,d \quad\rightsquigarrow_D\quad \dfrac{}{A\to A}\,a\wedge\dfrac{}{A\to A}\,a$$

As an example, observe $u[z_1,z_2\leftarrow\lambda x.(\lambda z.z)\,\lambda y.(y\,y)\,x]$ (note $\lambda x.t\equiv x\langle\,x\,\rangle.t$). By (d_2) we obtain $u'[e_1\langle\,z_1\,\rangle,e_2\langle\,z_2\,\rangle\,|\,x\langle\,x\,\rangle\,[z_1,z_2\leftarrow(\lambda z.z)\,\lambda y.(y\,y)\,x]]$ where $u'=u\{e_i\langle\,z_i\,\rangle.z_i/z_i\}_{i\in[2]}$. Then by reductions $(d_1,\ l_5)$, we obtain the distributor $u''[e_1\langle\,z_1\,\rangle,e_2\langle\,z_2\,\rangle\,|\,x\langle\,x\,\rangle\,[z_1,z_2\leftarrow\lambda y.(y\,y)\,x]]$ where $u''=u\{e_i\langle\,z_i\,\rangle.a_i\,z_i/z_i\}_{i\in[2]}$. Then by $(d_2,\ d_1,\ l_5,\ l_5)$ we obtain the distributor $u'''[e_1\langle\,z_1\,\rangle,e_2\langle\,z_2\,\rangle\,|\,x\langle\,x\,\rangle\,[z_1,z_2\leftarrow x]]$ which can be eliminated by (d_3). A full example can be found in the Appendix.

Each rewrite rule preserves the conclusion of the derivation, and thus the following proposition is easy to observe.

▶ **Proposition 15.** *If $s\rightsquigarrow_{(L,C,D,\beta)}t$ and $s:C$, then $t:C$*

The readback translation collapses the shared terms. The lifting, duplication, and compound rules are used solely for the duplication of terms. Therefore it is expected that the following Lemma be true (proven in Appendix by induction). It is also important for proving confluence of Λ_a^S (Theorem 34).

▶ **Lemma 16.** *If $s\rightsquigarrow_{(L,D,C)}t$ then $[\![\,s\,|\,\sigma\,|\,\gamma\,]\!]=[\![\,t\,|\,\sigma\,|\,\gamma\,]\!]$*

4 Strong Normalisation of Sharing Reductions

In order to show our calculus is strongly normalising, we first show that the sharing reduction rules are strongly normalising. To do this, we make use of an 'intermediate calculus' called the weakening calculus. Following the approaches of [15], we indite a measure on terms based on its connection with the weakening calculus. We show that this measure strictly decreases as sharing reduction progresses. Additionally, similar ideas and results can be found elsewhere, i.e. with memory in [19], the λ-I calculus in [4], the λ-void calculus [2], and the weakening $\lambda\mu$-calculus [16].

▶ **Definition 17.** *The $\mathcal{W}$-terms and the weakening calculus ($\Lambda_{\mathcal{W}}$) are*

$$T,U,V\quad::=\quad x\quad|\quad\lambda x.T^*\quad|\quad U\,V\quad|\quad T[\leftarrow U]\quad|\quad\bullet \qquad (*)\text{ where } x\in(T)_{fv}$$

The terms are variable, abstraction, application, weakening, and a bullet. In the weakening $T[\leftarrow U]$, the subterm U is *weakened*. The interpretation of atomic terms to weakening terms $[\![\,-\,|\,-\,|\,-\,]\!]_{\mathcal{W}}$ can be seen as an extension of the translation into the λ-calculus (Definition 9)

▶ **Definition 18.** *The interpretation $[\![\,-\,|\,-\,|\,-\,]\!]_{\mathcal{W}}:\Lambda_a^S\times(V\to\Lambda_{\mathcal{W}})\times(V\to V)\to\Lambda_{\mathcal{W}}$ with maps $\sigma:V\to\Lambda_{\mathcal{W}}$ and $\gamma:V\to V$ is defined as an extension of the translation in (Definition*

9) with the following additional special cases.

$$[\![u[\leftarrow t] \mid \sigma \mid \gamma]\!]_{\mathcal{W}} = [\![u \mid \sigma \mid \gamma]\!]_{\mathcal{W}} [\leftarrow [\![t \mid \sigma \mid \gamma]\!]_{\mathcal{W}}]$$

$$[\![u[\,|\, c\langle\, c \,\rangle\, \overline{[\Gamma]}] \mid \sigma \mid \gamma]\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \mid \sigma[c \mapsto \bullet] \mid \gamma]\!]_{\mathcal{W}}$$

$$[\![u[\,|\, c\langle\, x_1, \ldots, x_n \,\rangle\, \overline{[\Gamma]}] \mid \sigma \mid \gamma]\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \mid \sigma' \mid \gamma]\!]_{\mathcal{W}}$$

$$\textit{where } \sigma'(z) = \begin{cases} \sigma(z)\{\bullet/\gamma(c)\} & z \in \{x_1, \ldots, x_n\} \\ \sigma(z) & \textit{otherwise} \end{cases}$$

We also have translations of the weakening calculus to and from the lambda calculus. Both of these translations were provided in [15]. The interpretation $\lfloor - \rfloor$ from weakening terms to λ-terms discards all weakenings. The interpretation $(\!|-|\!)^{\mathcal{W}} : \Lambda \to \Lambda_{\mathcal{W}}$ is defined below.

▶ **Definition 19.** *The interpretation $M \in \Lambda$, $(\!|-|\!)^{\mathcal{W}} : \Lambda \to \Lambda_{\mathcal{W}}$ is defined by*

$$\begin{aligned} (\!| x |\!)^{\mathcal{W}} &= x \\ (\!| M\, N |\!)^{\mathcal{W}} &= (\!| M |\!)^{\mathcal{W}}\, (\!| N |\!)^{\mathcal{W}} \\ (\!| \lambda x.N |\!)^{\mathcal{W}} &= \begin{cases} \lambda x.(\!| N |\!)^{\mathcal{W}} & \textit{if } x \in (N)_{fv} \\ \lambda x.(\!| N |\!)^{\mathcal{W}}[\leftarrow x] & \textit{otherwise} \end{cases} \end{aligned}$$

The following equalities can be observed, where $\sigma^{\Lambda}(z) = \lfloor \sigma^{\mathcal{W}}(z) \rfloor$.

▶ **Proposition 20.** *For $N \in \Lambda$ and $t \in \Lambda_a^S$ the following properties hold*

$$\lfloor [\![t \mid \sigma^{\mathcal{W}} \mid \gamma]\!]_{\mathcal{W}} \rfloor = [\![t \mid \sigma^{\Lambda} \mid \gamma]\!] \qquad [\![(\!| N |\!)]\!]^{\mathcal{W}} = (\!| N |\!)^{\mathcal{W}} \qquad \lfloor (\!| N |\!)^{\mathcal{W}} \rfloor = N$$

▶ **Definition 21.** *In the weakening calculus, β-reduction is defined as follows, where $\overline{[\Gamma]}$ are weakening constructs.*

$$((\lambda x.T)\overline{[\Gamma]})\, U \to_{\beta} T\{U/x\}\overline{[\Gamma]} \tag{w_β}$$

Here we can take advantage that preservation of strong normalisation has been proven for this weakening calculus already in [15], providing the proof for Proposition 22.

▶ **Proposition 22.** *If $N \in \Lambda$ is strongly normalising, then so is $(\!| N |\!)^{\mathcal{W}}$*

When translating from the spinal atomic λ-calculus to the weakening calculus, weakenings are maintained whilst sharings are interpreted through duplication via substitution. Thus the reduction rules in the weakening calculus cover the spinal reductions for nullary distributors and weakenings.

▶ **Definition 23.** *The weakening reductions ($\to_{\mathcal{W}}$) proceeds as follows.*

$$\begin{aligned} \lambda x.T[\leftarrow U] &\to_{\mathcal{W}} (\lambda x.T)[\leftarrow U] \quad \textit{if } x \notin (U)_{fv} && (w_1) \\ U[\leftarrow T]\, V &\to_{\mathcal{W}} (U\, V)[\leftarrow T] && (w_2) \\ U\, V[\leftarrow T] &\to_{\mathcal{W}} (U\, V)[\leftarrow T] && (w_3) \\ T[\leftarrow U[\leftarrow V]] &\to_{\mathcal{W}} T[\leftarrow U][\leftarrow V] && (w_4) \\ T[\leftarrow \lambda x.U] &\to_{\mathcal{W}} T[\leftarrow U\{\bullet/x\}] && (w_5) \\ T[\leftarrow U\, V] &\to_{\mathcal{W}} T[\leftarrow U][\leftarrow V] && (w_6) \\ T[\leftarrow \bullet] &\to_{\mathcal{W}} T && (w_7) \\ T[\leftarrow U] &\to_{\mathcal{W}} T \quad \textit{if } U \textit{ is a subterm of } T && (w_8) \end{aligned}$$

It is easy to see that these rules correspond to special cases of the sharing reduction rules for Λ_a^S. Lifting a closure relates (w_1) and (l_3), (w_2) and (l_1), (w_3) and (l_2), (w_4) and (l_4), (w_5) and (d_2), and duplicating a term relates (w_6) and (d_1), and (w_7) and (d_3). It is not so obvious to see what the case (w_8) corresponds to. If U is a subterm of T, then in the corresponding Λ_a^S-term this term would be shared and one of the copies would be in a weakening. Thus this reduction relates to the case (c_1), where we remove the weakening. We demonstrate by considering $t[\leftarrow y][\vec{x}\cdot y\cdot\vec{z}\leftarrow u] \rightsquigarrow_C t[\vec{x}\cdot\vec{z}\leftarrow u]$. On the left hand side, the corresponding weakening-term (obtained by $(\!|-|\!)^{\mathcal{W}}$) would have the weakening $[\leftarrow U]$ where $U = (\!|u|\!)^{\mathcal{W}}$. This is because U is substituted into $[\leftarrow y]$, but on the right hand side this would be gone. This situation can only occur if there are other copies of U substituted into the term. This corresponds to if only the corresponding (c_1) reduction rule can occur. This resemblace is confirmed by the following Lemmas.

▶ **Lemma 24.** *If $t \rightsquigarrow_\beta u$ then $[\![t]\!]^{\mathcal{W}} \to_\beta^+ [\![u]\!]^{\mathcal{W}}$*

▶ **Lemma 25.** *If $t \rightsquigarrow_{(C,D,L)} u$ and for any $x \in (t)_{bv} \cup (t)_{fp}$ and for all z, $x \notin (\sigma(z))_{fv}$.*

$$[\![t\,|\,\sigma\,|\,\gamma]\!]_{\mathcal{W}} \to_{\mathcal{W}}^* [\![u\,|\,\sigma\,|\,\gamma]\!]_{\mathcal{W}}$$

We now define the components that we use for our measure on spinal atomic λ-terms that we will use to prove strong normalisation of sharing reductions. The *height* of a term is intuitively a multiset of integers that record the distance of each sharing. The distance is measured by the number of constructors from the sharing node to the root of the term in its graphical notation. The height is defined on terms as $\mathcal{H}^i(-)$, where i is an integer. We say $\mathcal{H}(t)$ for $\mathcal{H}^1(t)$. We use $\uplus$ to denote the disjoint union of two multisets. We denote $\mathcal{H}^i([\Gamma_1]) \uplus \cdots \uplus \mathcal{H}^i([\Gamma_n])$ as $\mathcal{H}^i(\overline{[\Gamma]})$ for the environment $\overline{[\Gamma]} = [\Gamma_1], \ldots, [\Gamma_n]$.

▶ **Definition 26** (Sharing Height)**.** *The sharing height $\mathcal{H}^i(t)$ of a term t is given by*

$$\begin{aligned}
\mathcal{H}^i(x) &= \{\} \\
\mathcal{H}^i(s\,t) &= \mathcal{H}^{i+1}(s) \uplus \mathcal{H}^{i+1}(t) \\
\mathcal{H}^i(c\langle\,\vec{x}\,\rangle.t) &= \mathcal{H}^{i+1}(t) \\
\mathcal{H}^i(t[\Gamma]) &= \mathcal{H}^i(t) \uplus \mathcal{H}^i([\Gamma]) \uplus \{i^1\} \\
\mathcal{H}^i([x_1,\ldots,x_n \leftarrow t]) &= \mathcal{H}^{i+1}(t) \\
\mathcal{H}^i([\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}]) &= \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \{(i+1)^n\} \text{ where } n \text{ is the number of closures in } \overline{[\Gamma]}
\end{aligned}$$

This measure then strictly decreases for the rewrite rules l_1, l_2, l_3, l_4 and l_5.

▶ **Lemma 27.** *If $t \rightsquigarrow_L u$ then $\mathcal{H}^i(t) > \mathcal{H}^i(u)$*

The other measure we consider is the *weight* of a term. Intuitively this quantifies the remaining duplications, which are performed with $\rightsquigarrow_D$ reductions. Calculating the weight of a term requires an auxiliary function from variables to integers. This function is defined by assigning integer weights to the variables of a term. This auxiliary function is defined on terms $\mathcal{V}^i(-)$, where i is an integer. To measure variables independently of binders is vital. It allows to measure distributors, which duplicate λ's but not the bound variable. Also, only bound variables for abstractions are measured since variables bound by sharings are substituted in the interpretation.

▶ **Definition 28** (Variable Weights). *The function $\mathcal{V}^i(t)$ returns a function that assigns integer weights to the free variables of t. It is defined by the following*

$$\begin{aligned}
\mathcal{V}^i(x) &= \{x \mapsto i\} \\
\mathcal{V}^i(s\,t) &= \mathcal{V}^i(s) \uplus \mathcal{V}^i(t) \\
\mathcal{V}^i(c\langle c\rangle.t) &= \mathcal{V}^i(t)/\{c\} \\
\mathcal{V}^i(c\langle \vec{x}\rangle.t) &= \mathcal{V}^i(t) \uplus \{c \mapsto i\} \\
\mathcal{V}^i(t[\leftarrow s]) &= \mathcal{V}^i(t) \uplus \mathcal{V}^1(s) \\
\mathcal{V}^i(t[x_1,\dots,x_n \leftarrow s]) &= \mathcal{V}^i(t)/\{x_1,\dots,x_n\} \uplus \mathcal{V}^{f(x_1)+\dots+f(x_n)}(s) \text{ where } f = \mathcal{V}^i(t) \\
\mathcal{V}^i(t[e_1\langle \vec{w_1}\rangle \dots e_n\langle \vec{w_n}\rangle \,|\, c\langle c\rangle\,\overline{[\Gamma]}]) &= \mathcal{V}^i(t\overline{[\Gamma]})/\{c, e_1,\dots,e_n\} \\
\mathcal{V}^i(t[e_1\langle \vec{w_1}\rangle \dots e_n\langle \vec{w_n}\rangle \,|\, c\langle \vec{x}\rangle\,\overline{[\Gamma]}]) &= \mathcal{V}^i(t\overline{[\Gamma]})/\{e_1,\dots,e_n\} \uplus \{c \mapsto i\}
\end{aligned}$$

The weight of a term can then be defined via the use of this auxiliary function. The auxiliary function is used when calculating the weight of a sharing, where the sharing weight of the variables bound by the sharing play a significant role in calculating the weight of the shared term. In the case of a weakening, we assign an initial weight of 1 to indicate that the constructor is not duplicated by appears at least once in the weakening calculus. Again we say $\mathcal{W}(t) = \mathcal{W}^1(t)$.

▶ **Definition 29** (Sharing Weight). *The sharing weight $\mathcal{W}^i(t)$ of a term t is a multiset of integers computed by the function defined below*

$$\begin{aligned}
\mathcal{W}^i(x) &= \{\} \\
\mathcal{W}^i(s\,t) &= \mathcal{W}^i(s) \uplus \mathcal{W}^i(t) \uplus \{i\} \\
\mathcal{W}^i(c\langle c\rangle.t) &= \mathcal{W}^i(t) \uplus \{i\} \uplus \{\mathcal{V}^i(t)(c)\} \\
\mathcal{W}^i(c\langle \vec{x}\rangle.t) &= \mathcal{W}^i(t) \uplus \{i\} \\
\mathcal{W}^i(t[\leftarrow s]) &= \mathcal{W}^i(t) \uplus \mathcal{W}^1(s) \\
\mathcal{W}^i(t[x_1,\dots,x_n \leftarrow s]) &= \mathcal{W}^i(t) \uplus \mathcal{W}^{f(x_1)+\dots+f(x_n)}(s) \text{ where } f = \mathcal{V}^i(t) \\
\mathcal{W}^i(t[e_1\langle \vec{w_1}\rangle \dots e_n\langle \vec{w_n}\rangle \,|\, c\langle c\rangle\,\overline{[\Gamma]}]) &= \mathcal{W}^i(t\overline{[\Gamma]}) \uplus \{\mathcal{V}^i(t\overline{[\Gamma]})(c)\} \\
\mathcal{W}^i(t[e_1\langle \vec{w_1}\rangle \dots e_n\langle \vec{w_n}\rangle \,|\, c\langle \vec{x}\rangle\,\overline{[\Gamma]}]) &= \mathcal{W}^i(t\overline{[\Gamma]})
\end{aligned}$$

We show that this measure then strictly decreases on the rewrite rules d_1, d_2, d_3 and is unaffected by all the other sharing reduction rules.

▶ **Lemma 30.** *If $t \rightsquigarrow_D u$ then $\mathcal{W}^i(t) > \mathcal{W}^i(u)$*

▶ **Lemma 31.** *If $t \rightsquigarrow_{(L,C)} u$ then $\mathcal{W}^i(t) = \mathcal{W}^i(u)$*

The last measure we consider is the number of closures in the term, where is can be easily observed that the rewrite rules c_1 and c_2 strictly decrease this measure, and that the $\rightsquigarrow_L$ rules do not alter the number of closures. We then use this along with height and weight to define a *sharing measure* on terms.

▶ **Definition 32.** *The* sharing measure *of a Λ_a^S-term t is a triple $(\mathcal{W}(t), C, \mathcal{H}(t))$ where C is the number of closures in t. We can compare two different sharing measures by considering the lexicographical preferences according to weight > number of closures > height.*

▶ **Theorem 33.** *Sharing reduction $\rightsquigarrow_{(D,L,C)}$ is strongly normalising*

Proof. From Lemma 30, Lemma 31, and Lemma 27, it follows that the sharing measure of a term is strictly decreasing under $\rightsquigarrow_{(D,L,C)}$, proving the statement. ◀

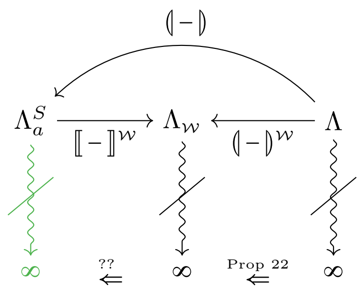
Now that we have proven the sharing reductions are strongly normalising, we can prove that they are confluent for closed terms.

▶ **Theorem 34.** *The sharing reduction relation $\rightsquigarrow_{(D,L,C)}$ is confluent*

Proof. Lemma 16 tells us that the preservation is preserved under reduction i.e. for $s \rightsquigarrow_{(D,L,C)} t$, $[\![s]\!] = [\![t]\!]$. Therefore given $t \rightsquigarrow^*_{(D,L,C)} s_1$ and $t \rightsquigarrow^*_{(D,L,C)} s_2$, $[\![t]\!] = [\![s_1]\!] = [\![s_2]\!]$. Since we know that sharing reductions are strongly normalising, we know there exists terms u_1 and u_2 in sharing normal form such that $s_1 \rightsquigarrow^*_{(D,L,C)} u_1$ and $s_2 \rightsquigarrow^*_{(D,L,C)} u_2$. Lemma 11 tells us that terms in closed terms in sharing normal form are in correspondence with their denotations i.e. $(\!| [\![t]\!] |\!)' = t$. Since by Lemma 16 we know $[\![u_1]\!] = [\![s_1]\!] = [\![s_2]\!] = [\![u_2]\!]$, and by Lemma 11 $(\!| [\![u_1]\!] |\!)' = u_1$ and $(\!| [\![u_2]\!] |\!)' = u_2$, we can conclude $u_1 = u_2$. Hence, we prove confluence. ◀

5 Preservation of Strong Normalisation

Here we show how Λ^S_a preserves strong normalisation with respect to the λ-calculus. Recall that by Proposition 20 that for all $N \in \Lambda$, $[\![(\!| N |\!)]\!]^{\mathcal{W}} = (\!| N |\!)^{\mathcal{W}}$, and that Proposition 22 states if a term $N \in \Lambda$ is strongly normalising then so is $(\!| N |\!)^{\mathcal{W}}$. Observe that the statement 'if term M has an infinite reduction sequence then term N has an infinite reduction sequence' is equivalent to 'if term N is strongly normalising then term M is strongly normalising' by contraposition. Therefore, given a strongly normalising term $N \in \Lambda$, we know that its corresponding weakening term is also strongly normalising. Furthermore, since $[\![(\!| N |\!)]\!]^{\mathcal{W}} = (\!| N |\!)^{\mathcal{W}}$, we know that $[\![(\!| N |\!)]\!]^{\mathcal{W}}$ is also strongly normalising.



We prove that the spinal atomic λ-calculus preserves strong normalisation with the following.

▶ **Lemma 35.** *For $t \in \Lambda^S_a$ has an infinite reduction path, then $[\![t]\!]^{\mathcal{W}}$ also has an infinite reduction path.*

Proof. Due to Theorem 34, we know that the infinite reduction path contains an infinite β-reduction. This means in the reduction sequence, between each β-reduction, there are finite many $\rightsquigarrow_{(D,L,C)}$ reduction steps. Lemma 25 says each $\rightsquigarrow_{(D,L,C)}$ step in Λ^S_a corresponds to zero or more weakening reductions ($\rightsquigarrow^*_{\mathcal{W}}$). Lemma 24 says that each beta reduction in Λ^S_a corresponds to one or more β-steps in $\Lambda_{\mathcal{W}}$. Therefore, it is inevitable that $[\![t]\!]^{\mathcal{W}}$ also has an infinite reduction path. ◀

▶ **Theorem 36.** *If $N \in \Lambda$ is strongly normalising, then so is $(\!| N |\!)$.*

Proof. For a given $N \in \Lambda$ that is strongly normalising, we know by Lemma 22 that $(\!| N |\!)^{\mathcal{W}}$ is strongly normalising. Then $[\![(\!| N |\!)]\!]^{\mathcal{W}}$ is strongly normalising, since Proposition 20 states that $(\!| N |\!)^{\mathcal{W}} = [\![(\!| N |\!)]\!]^{\mathcal{W}}$. Then by Lemma 35, which states that if $[\![t]\!]^{\mathcal{W}}$ is strongly normalising, then t is strongly normalising, proves that $(\!| N |\!)$ is strongly normalising. ◀

6 Conclusion and Further Remarks

We have studied the computational interpretation of the switch rule and discovered its correspondence with scope in the λ-calculus. We have studied the interaction between the switch and the medial rule, the two characteristic inference rules of deep inference. We interpret a calculus based on this interaction, which not only has the ability to duplicate terms atomically but can also duplicate solely the spine of an abstraction such that beta reduction can be applied on the duplicates. We show that this resulting calculus has natural properties with respect to the λ-calculus.

In the future we would like to have a full Curry-Howard correspondence rather than just an interpretation, i.e. where each inference rule in the typing system corresponds with a construct in the term calculus. This would mean introducing an explicit end-of-scope operator (such as done in [5, 17, 13]) to correspond with the switch rule. Additionally, we aim to translate the result of Blanc, Lévy, and Maranget [7] into our calculus. There they provide an algorithm proven by Balabonski in [3] to implement optimal reduction for Wadsworth's *weak λ-calculus* [25] (further studied in [10]). By showing their result in our formalism, we would develop a logical framework that follows an optimal reduction strategy.

References

1. Martin Abadi, Luca Cardelli, Pierre-Loius Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991. `doi:10.1017/S0956796800000186`.
2. Beniamino Accattoli and Delia Kesner. The permutative λ-calculus. In Nikolaj Bjørner and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 23–36, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
3. Thibaut Balabonski. A unified approach to fully lazy sharing. In *Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '12, pages 469–480, New York, NY, USA, 2012. ACM. URL: `http://doi.acm.org/10.1145/2103656.2103713`, `doi:10.1145/2103656.2103713`.
4. Henk P Barendregt. The lambda calculus: Its syntax and semantics, revised ed., vol. 103 of studies in logic and the foundations of mathematics, 1984.
5. Klaus. J. Berkling. *A Symmetric Complement to the Lambda Calculus*. Bonn Interner Bericht ISF. Gesellschaft für Mathematik und Datenverarbeitung mbH, 1976. URL: `https://books.google.de/books?id=T5FLQwAACAAJ`.
6. Klaus J. Berkling and Elfriede Fehr. A consistent extension of the lambda-calculus as a base for functional programming languages. *Information and Control*, 55(1):89 – 101, 1982. URL: `http://www.sciencedirect.com/science/article/pii/S0019995882904582`, `doi:https://doi.org/10.1016/S0019-9958(82)90458-2`.
7. Tomasz Blanc, Jean-Jacques Lévy, and Luc Maranget. Sharing in the weak lambda-calculus revisited. In Erik Barendsen, Herman Geuvers, Venanzio Capretta, and Milad Niqui, editors, *Reflections on Type Theory, Lambda Calculus, and the Mind, Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday*, pages 41–50. Nijmegen Radboud Universiteit Nijmegen, 2007.
8. Guy Blelloch and John Greiner. Parallelism in sequential functional languages. In *Proceedings of the Seventh International Conference on Functional Programming Languages and Computer*

Architecture, FPCA '95, pages 226–237, New York, NY, USA, 1995. ACM. URL: http://doi.acm.org/10.1145/224164.224210, doi:10.1145/224164.224210.

9 Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001. URL: http://cs.bath.ac.uk/ag/kai/lcl-lpar.pdf, doi:10.1007/3-540-45653-8_24.

10 Naim Cagman and J.Roger Hindley. Combinatory weak reduction in lambda calculus. *Theoretical Computer Science*, 198(1):239 – 247, 1998. URL: http://www.sciencedirect.com/science/article/pii/S0304397597002508, doi:https://doi.org/10.1016/S0304-3975(97)00250-8.

11 Maribel Fernández and Ian Mackie. Closed reductions in the λ-calculus. In Jörg Flum and Mario Rodriguez-Artalejo, editors, *Computer Science Logic*, pages 220–234, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

12 Maribel Fernández, Ian Mackie, and François-Régis Sinot. Closed reduction: explicit substitutions without α-conversion. *Mathematical Structures in Computer Science*, 15(2):343–381, 2005. doi:10.1017/S0960129504004633.

13 Maribel Fernández, Ian Mackie, and François-Régis Sinot. Lambda-calculus with director strings. *Applicable Algebra in Engineering, Communication and Computing*, 15(6):393–437, Apr 2005. URL: https://doi.org/10.1007/s00200-005-0169-9, doi:10.1007/s00200-005-0169-9.

14 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1:1–64, 2007. URL: http://cs.bath.ac.uk/ag/p/SystIntStr.pdf, doi:10.1145/1182613.1182614.

15 Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic lambda calculus: A typed lambda-calculus with explicit sharing. In Orna Kupferman, editor, *28th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 311–320. IEEE, 2013. URL: http://opus.bath.ac.uk/34527/1/AL.pdf, doi:10.1109/LICS.2013.37.

16 Fanny He. *The Atomic Lambda-Mu Calculus*. PhD thesis, University of Bath, 2018. URL: https://fh341.github.io/pdf/HE-Thesis.pdf.

17 Dimitri Hendriks and Vincent van Oostrom. Adbmal. In Franz Baader, editor, *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach, FL, USA, July 28 - August 2, 2003, Proceedings*, volume 2741 of *Lecture Notes in Computer Science*, pages 136–150, 2003. URL: https://doi.org/10.1007/978-3-540-45085-6_11, doi:10.1007/978-3-540-45085-6_11.

18 Richard Kennaway and Ronan Sleep. Director strings as combinators. *ACM Trans. Program. Lang. Syst.*, 10(4):602–626, October 1988. URL: http://doi.acm.org/10.1145/48022.48026, doi:10.1145/48022.48026.

19 Jan Willem Klop. *Combinatory Reduction Systems*. PhD thesis, Utrecht University, 1980.

20 Yves Lafont. From proof-nets to interaction nets. In *Advances in Linear Logic*, pages 225–247. Cambridge University Press, 1994.

21 Jean-Jacques Lévy. Optimal reductions in the lambda calculus. *To HB Curry: Essays on Combinatory Logic, Lambda Coalculus and Formalism*, pages 159–191, 1980.

22 François-Régis Sinot, Maribel Fernández, and Ian Mackie. Efficient reductions with director strings. In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications*, pages 46–60, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

23 Alwen Tiu. A system of interaction and structure II: The need for deep inference. *Logical Methods in Computer Science*, 2(2):4:1–24, 2006. URL: https://arxiv.org/pdf/cs/0512036.pdf, doi:10.2168/LMCS-2(2:4)2006.

24 Vincent van Oostrom, Kees-Jan van de Looij, and Marijn Zwitserlood. Lambdascope: another optimal implementation of the lambda-calculus. In *Workshop on Algebra and Logic on Programming Systems (ALPS)*, 2004.

25 Christopher P. Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus.* PhD thesis, University of Oxford, 1971.

A The Spinal Atomic λ-Calculus

A.1 Compilation and Readback

In this section we provide the proof for Proposition 11: For $s, t \in \Lambda_a^S$, if $s \sim t$ then $[\![s]\!] = [\![t]\!]$.

Proof. Let us consider the cases.

$t[\Gamma_1][\Gamma_2] \sim t[\Gamma_2][\Gamma_1]$

Consider $[\![t[\Gamma_1][\Gamma_2] \,|\, \sigma \,|\, \gamma]\!] = [\![t[\Gamma_1] \,|\, \sigma' \,|\, \gamma']\!] = [\![t \,|\, \sigma'' \,|\, \gamma'']\!]$. Since due to conditions any variable $x \in \| [\Gamma_2] \|$ cannot occur in $[\Gamma_1]$, for all subterms s located in $[\Gamma_1]$, $[\![s \,|\, \sigma' \,|\, \gamma']\!] = [\![s \,|\, \sigma \,|\, \gamma]\!]$. Therefore $[\![t \,|\, \sigma'' \,|\, \gamma'']\!] = [\![t[\Gamma_2] \,|\, \sigma''' \,|\, \gamma''']\!] = [\![t[\Gamma_2][\Gamma_1] \,|\, \sigma \,|\, \gamma]\!]$.

The remaining cases discuss permutations of variables in sharings and phantom-abstractions. In both these cases, we overwrite σ for the cases of the variables in said sharing or phantom-abstractions. The order in which they appear do not influence the translation since we do this for all variables regardless. ◀

We also provide the proof for Lemma 11: For a closed $t \in \Lambda_a^S$, where t has no distributor constructs and only variables are shared, and a closed $N \in \Lambda$. the following

$$[\![(\!| N |\!)']\!] = N \qquad\qquad (\!| [\![t]\!] |\!)' = t \qquad\qquad \exists_{M \in \Lambda}. t = (\!| M |\!)'$$

Proof. We prove $[\![(\!| N |\!)']\!] = N$ by induction on N

Base Case: Variable
$[\![(\!| x |\!)']\!] = [\![x]\!] = x$

Inductive Case: Application
$[\![(\!| M\, N |\!)']\!] = [\![(\!| M |\!)']\!] \, [\![(\!| N |\!)']\!] = M\, N$

Inductive Case: Abstraction
$[\![(\!| \lambda x.M |\!)']\!]$
Case: $|M|_x = 1$
$= \lambda x.[\![(\!| M |\!)']\!] = \lambda x.M$

Case: $|M|_x = n$
$= \lambda x.[\![(\!| M\frac{n}{x} |\!)'[x_1, \ldots, x_n \leftarrow x]]\!] = \lambda x.[\![(\!| M\frac{n}{x} |\!)' \,|\, \sigma \,|\, I]\!] = \lambda x.[\![(\!| M\frac{n}{x} |\!)']\!]\{x/x_i\}_{1 \le i \le n}$
$\overset{\text{I.H.}}{=} \lambda x.M\frac{n}{x}\{x/x_i\}_{1 \le i \le n} = \lambda x.M$

We prove $(\!| [\![t]\!] |\!)' = t$ by induction on t

Base Case: Variable
$(\!| [\![x]\!] |\!)' = (\!| x |\!)' = x$

Inductive Case: Application
$(\!| [\![s\, t]\!] |\!)' = (\!| [\![s]\!] |\!)' (\!| [\![t]\!] |\!)' \overset{\text{I.H.}}{=} s\, t$

Inductive Case: Abstraction

Case: $\llparenthesis \llbracket x\langle x \rangle.t \rrbracket \rrparenthesis' = x\langle x \rangle.\llparenthesis \llbracket t \rrbracket \rrparenthesis' \overset{\text{I.H.}}{=} x\langle x \rangle.t$

Case: $\llparenthesis \llbracket x\langle x \rangle.t[x_1, \dots, x_n \leftarrow x] \rrbracket \rrparenthesis' = \llparenthesis \lambda x.\llbracket t \,|\, \sigma \,|\, I \rrbracket \rrparenthesis'$
$= \llparenthesis \lambda x.\llbracket t \rrbracket \{x/x_i\}_{1 \le i \le n} \rrparenthesis' = x\langle x \rangle.\llparenthesis \llbracket t \rrbracket \rrparenthesis'[x_1, \dots, x_n \leftarrow x]$
$\overset{\text{I.H.}}{=} x\langle x \rangle.t[x_1, \dots, x_n \leftarrow x]$

The proof for $\exists_{M \in \Lambda}.t = \llparenthesis M \rrparenthesis'$ is the same as in [15]. ◀

A.2 Rewrite Rules

Here we will give more concrete definitions of substitution, book-keeping and exorcisms respectively.

▶ **Definition 37** (Substitution). *The operation* substitution *is defined as*

$$
\begin{aligned}
x\{s/x\} &= s \\
y\{s/x\} &= y \\
(u\,t)\{s/x\} &= (u\{s/x\})\,t\{s/x\} \\
(c\langle \vec{y} \rangle.t)\{s/x\} &= c\langle \vec{y} \rangle.t\{s/x\} \\
(c\langle \vec{y} \cdot x \rangle.t)\{s/x\} &= c\langle \vec{y} \cdot \vec{z} \rangle.t\{s/x\} \\
u[\vec{y} \leftarrow t]\{s/x\} &= u\{s/x\}[\vec{y} \leftarrow t\{s/x\}] \\
u[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \rangle \,\overline{[\Gamma]}]\{s/x\} &= u[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \rangle \,\overline{[\Gamma]}\{s/x\}] \\
u[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \cdot x \rangle \,\overline{[\Gamma]}]\{s/x\} &= u[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \cdot \vec{z} \rangle \,\overline{[\Gamma]}\{s/x\}] \\
u[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \rangle \{s/x\}\overline{[\Gamma]}] &= u\{s/x\}[\overrightarrow{e\langle \vec{w} \rangle} \,|\, c\langle \vec{y} \rangle \,\overline{[\Gamma]}] \\
u[e\{e_i\langle \vec{w} \cdot x \rangle\} \,|\, c\langle \vec{y} \rangle \{s/x\}\overline{[\Gamma]}] &= u\{s/x\}[e\{e_i\langle \vec{w} \cdot \vec{z} \rangle\} \,|\, c\langle \vec{y} \rangle \,\overline{[\Gamma]}]
\end{aligned}
$$

Where $\vec{z} = (s)_{fv}$

Although substitution performs some book-keeping on phantom-abstractions, we define an explicit notion that updates the variables stored in a free-cover i.e. for a term t, $e\langle \vec{x} \rangle \in (t)_{fc}$ then $e\langle \vec{y} \rangle \in (t\{\vec{y}/e\}_b)_{fc}$.

▶ **Definition 38** (Book-Keeping). *The operation* book-keeping *is defined as*

$$
\begin{aligned}
x\{\vec{w}/e\}_b &= x \\
s\,t\{\vec{w}/e\}_b &= (s\{\vec{w}/e\}_b)\,t\{\vec{w}/e\}_b \\
e\langle \vec{z} \rangle.t\{\vec{w}/e\}_b &= e\langle \vec{w} \rangle.t \\
(c\langle \vec{z} \rangle.t)\{\vec{w}/e\}_b &= c\langle \vec{z} \rangle.t\{\vec{w}/e\}_b \\
u[\vec{z} \leftarrow t]\{\vec{w}/e\}_b &= u\{\vec{w}/e\}_b[\vec{z} \leftarrow t\{\vec{w}/e\}_b] \\
u[\overrightarrow{f\langle \vec{y} \rangle} \,|\, e\langle \vec{z} \rangle \,\overline{[\Gamma]}]\{\vec{w}/e\}_b &= u[\overrightarrow{f\langle \vec{y} \rangle} \,|\, e\langle \vec{w} \rangle \,\overline{[\Gamma]}] \\
u[\overrightarrow{f\langle \vec{y} \rangle} \,|\, c\langle \vec{z} \rangle \,\overline{[\Gamma]}]\{\vec{w}/e\}_b &= u[\overrightarrow{f\langle \vec{y} \rangle} \,|\, c\langle \vec{z} \rangle \,\overline{[\Gamma]}\{\vec{w}/e\}_b] \\
u[\overrightarrow{f\langle \vec{y} \rangle} \,|\, c\langle \vec{z} \rangle \{\vec{w}/e\}_b\overline{[\Gamma]}] &= u\{\vec{w}/e\}_b[\overrightarrow{f\langle \vec{y} \rangle} \,|\, c\langle \vec{z} \rangle \,\overline{[\Gamma]}]
\end{aligned}
$$

▶ **Definition 39** (Exorcism). *The operation* exorcism *is defined as*

$$
y\{c\langle \vec{x} \rangle\}_e = y
$$

$$
\begin{aligned}
s\,t\{c\langle\,\vec{x}\,\rangle\}_e &= (s\{c\langle\,\vec{x}\,\rangle\}_e)\,t\{c\langle\,\vec{x}\,\rangle\}_e\\
c\langle\,\vec{x}\,\rangle.t\{c\langle\,\vec{x}\,\rangle\}_e &= c\langle\,c\,\rangle.t[\vec{x}\leftarrow c]\\
d\langle\,\vec{y}\,\rangle.t\{c\langle\,\vec{x}\,\rangle\}_e &= d\langle\,\vec{y}\,\rangle.t\{c\langle\,\vec{x}\,\rangle\}_e\\
u[\vec{y}\leftarrow t]\{c\langle\,\vec{x}\,\rangle\}_e &= u\{c\langle\,\vec{x}\,\rangle\}_e[\vec{y}\leftarrow t\{c\langle\,\vec{x}\,\rangle\}_e]\\
u[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}]\{c\langle\,\vec{x}\,\rangle\}_e &= u[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,c\langle\,c\,\rangle\,\overline{[\Gamma]}[\vec{x}\leftarrow c]]\\
u[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}]\{c\langle\,\vec{x}\,\rangle\}_e &= u[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}\{c\langle\,\vec{x}\,\rangle\}_e]\\
u[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,d\langle\,\vec{y}\,\rangle\,\{c\langle\,\vec{x}\,\rangle\}_e\overline{[\Gamma]}] &= u\{c\langle\,\vec{w}\,\rangle\}_e[\overrightarrow{e\langle\,\vec{w}\,\rangle}\,|\,d\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}]
\end{aligned}
$$

First, observe the following example that demonstrates the rewrite rules.

▶ **Example 40.** Take the λ-term $M = (\lambda f.\lambda x.f\,(f\,x))\,\lambda g.\lambda y.g\,(g\,y)$.
Then $(\!|\,M\,|\!) = (f\langle\,f\,\rangle.x\langle\,x\,\rangle.f_1\,(f_2\,x)[f_1,f_2\leftarrow f])\,(g\langle\,g\,\rangle.y\langle\,y\,\rangle.g_1\,(g_2\,y)[g_1,g_2\leftarrow g])$.
We then may have the following reduction sequence.

$$
\begin{aligned}
& (f\langle\,f\,\rangle.x\langle\,x\,\rangle.f_1\,(f_2\,x)[f_1,f_2\leftarrow f])\,(g\langle\,g\,\rangle.y\langle\,y\,\rangle.g_1\,(g_2\,y)[g_1,g_2\leftarrow g])\\
\rightsquigarrow_\beta\ & x\langle\,x\,\rangle.f_1\,(f_2\,x)[f_1,f_2\leftarrow g\langle\,g\,\rangle.y\langle\,y\,\rangle.g_1\,(g_2\,y)[g_1,g_2\leftarrow g]] && (\beta)\\
\rightsquigarrow_D\ & x\langle\,x\,\rangle.(f_1\langle\,w_1\,\rangle.w_1\,((f_2\langle\,w_2\,\rangle.w_2)\,x))\\
& \quad [f_1\langle\,w_1\,\rangle,f_2\langle\,w_2\,\rangle\,|\,g\langle\,g\,\rangle\,[w_1,w_2\leftarrow y\langle\,y\,\rangle.g_1\,(g_2\,y)[g_1,g_2\leftarrow g]]] && (d_2)\\
\rightsquigarrow_D\ & x\langle\,x\,\rangle.(f_1\langle\,z_1\,\rangle.y_1\langle\,z_1\,\rangle.z_1\,((f_2\langle\,z_2\,\rangle.y_2\langle\,z_2\,\rangle.z_2)\,x))\\
& \quad [f_1\langle\,z_1\,\rangle,f_2\langle\,z_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,z_1\,\rangle,y_2\langle\,z_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad [z_1,z_2\leftarrow g_1\,(g_2\,y)[g_1,g_2\leftarrow g]]]]] && (d_2)\\
\rightsquigarrow_L\ & x\langle\,x\,\rangle.(f_1\langle\,z_1\,\rangle.y_1\langle\,z_1\,\rangle.z_1\,((f_2\langle\,z_2\,\rangle.y_2\langle\,z_2\,\rangle.z_2)\,x))\\
& \quad [f_1\langle\,z_1\,\rangle,f_2\langle\,z_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,z_1\,\rangle,y_2\langle\,z_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad [z_1,z_2\leftarrow g_1\,(g_2\,y)][g_1,g_2\leftarrow g]]] && (l_4)\\
\rightsquigarrow_D\ & x\langle\,x\,\rangle.((f_1\langle\,a_1,b_1\,\rangle.y_1\langle\,a_1,b_1\,\rangle.a_1\,b_1)\,((f_2\langle\,a_2,b_2\,\rangle.y_2\langle\,a_2,b_2\,\rangle.a_2\,b_2)\,x))\\
& \quad [f_1\langle\,a_1,b_1\,\rangle,f_2\langle\,a_2,b_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,a_1,b_1\,\rangle,y_2\langle\,a_2,b_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad [a_1,a_2\leftarrow g_1][b_1,b_2\leftarrow g_2\,y][g_1,g_2\leftarrow g]]] && (d_1)\\
\rightsquigarrow_C\ & x\langle\,x\,\rangle.((f_1\langle\,a_1,b_1\,\rangle.y_1\langle\,a_1,b_1\,\rangle.a_1\,b_1)\,((f_2\langle\,a_2,b_2\,\rangle.y_2\langle\,a_2,b_2\,\rangle.a_2\,b_2)\,x))\\
& \quad [f_1\langle\,a_1,b_1\,\rangle,f_2\langle\,a_2,b_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,a_1,b_1\,\rangle,y_2\langle\,a_2,b_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad [b_1,b_2\leftarrow g_2\,y][a_1,a_2,g_2\leftarrow g]]] && (c_1)\\
\rightsquigarrow_D\ & x\langle\,x\,\rangle.((f_1\langle\,a_1,b_1,c_1\,\rangle.y_1\langle\,a_1,b_1,c_1\,\rangle.a_1\,(b_1\,c_1))\\
& \quad ((f_2\langle\,a_2,b_2,c_2\,\rangle.y_2\langle\,a_2,b_2,c_2\,\rangle.a_2\,(b_2\,c_2))\,x)\\
& \qquad [f_1\langle\,a_1,b_1,c_1\,\rangle,f_2\langle\,a_2,b_2,c_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,a_1,b_1,c_1\,\rangle,y_2\langle\,a_2,b_2,c_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad\quad [b_1,b_2\leftarrow g_2][c_1,c_2\leftarrow y][a_1,a_2,g_2\leftarrow g]]] && (d_1)\\
\rightsquigarrow_C\ & x\langle\,x\,\rangle.((f_1\langle\,a_1,b_1,c_1\,\rangle.y_1\langle\,a_1,b_1,c_1\,\rangle.a_1\,(b_1\,c_1))\\
& \quad ((f_2\langle\,a_2,b_2,c_2\,\rangle.y_2\langle\,a_2,b_2,c_2\,\rangle.a_2\,(b_2\,c_2))\,x)\\
& \qquad [f_1\langle\,a_1,b_1,c_1\,\rangle,f_2\langle\,a_2,b_2,c_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,a_1,b_1,c_1\,\rangle,y_2\langle\,a_2,b_2,c_2\,\rangle\,|\,y\langle\,y\,\rangle\\
& \qquad\quad [c_1,c_2\leftarrow y][a_1,b_1,a_2,b_2\leftarrow g]]] && (c_1)\\
\rightsquigarrow_L\ & x\langle\,x\,\rangle.((f_1\langle\,a_1,b_1,c_1\,\rangle.y_1\langle\,c_1\,\rangle.a_1\,(b_1\,c_1))\\
& \quad ((f_2\langle\,a_2,b_2,c_2\,\rangle.y_2\langle\,c_2\,\rangle.a_2\,(b_2\,c_2))\,x)\\
& \qquad [f_1\langle\,a_1,b_1,c_1\,\rangle,f_2\langle\,a_2,b_2,c_2\,\rangle\,|\,g\langle\,g\,\rangle[y_1\langle\,c_1\,\rangle,y_2\langle\,c_2\,\rangle\,|\,y\langle\,y\,\rangle
\end{aligned}
$$

$$[c_1, c_2 \leftarrow y]][a_1, b_1, a_2, b_2 \leftarrow g]] \qquad (l_5)$$

$$\rightsquigarrow_L \quad x\langle x \rangle.((f_1\langle a_1, b_1 \rangle.y_1\langle c_1 \rangle.a_1\,(b_1\,c_1))\,((f_2\langle a_2, b_2 \rangle.y_2\langle c_2 \rangle.a_2\,(b_2\,c_2))\,x))$$
$$[f_1\langle a_1, b_1 \rangle, f_2\langle a_2, b_2 \rangle \,|\, g\langle g \rangle\,[a_1, b_1, a_2, b_2 \leftarrow g]]$$
$$[y_1\langle c_1 \rangle, y_2\langle c_2 \rangle \,|\, y\langle y \rangle\,[c_1, c_2 \leftarrow y]] \qquad (l_5)$$

$$\rightsquigarrow_D \quad x\langle x \rangle.((f_1\langle f_1 \rangle.y_1\langle c_1 \rangle.a_1\,(b_1\,c_1)[a_1, b_1 \leftarrow f_1])$$
$$((f_2\langle f_2 \rangle.y_2\langle c_2 \rangle.a_2\,(b_2\,c_2)[a_2, b_2 \leftarrow f_2])\,x)$$
$$[y_1\langle c_1 \rangle, y_2\langle c_2 \rangle \,|\, y\langle y \rangle\,[c_1, c_2 \leftarrow y]] \qquad (d_3)$$

$$\rightsquigarrow_D \quad x\langle x \rangle.((f_1\langle f_1 \rangle.y_1\langle y_1 \rangle.a_1\,(b_1\,y_1)[a_1, b_1 \leftarrow f_1])$$
$$((f_2\langle f_2 \rangle.y_2\langle y_2 \rangle.a_2\,(b_2\,y_2)[a_2, b_2 \leftarrow f_2])\,x) \qquad (d_3)$$

In this section we provide the proof for Proposition 41: Given $M \in \Lambda$ such that for all $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$, the translation $[\![\, u \,|\, \sigma \,|\, \gamma \,]\!]$ commutes with substitution $\{M/x\}$ in the following way

$$[\![\, u\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma[x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]] \,|\, \gamma \,]\!]$$

Proof. We prove this by induction on u

Base Case: Variable
$[\![\, x\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, t \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, x \,|\, \sigma' \,|\, \gamma \,]\!]$

$[\![\, y \,|\, \sigma \,|\, \gamma \,]\!] = \sigma(y) = \sigma'(y) = [\![\, y \,|\, \sigma' \,|\, \gamma \,]\!]$

Inductive Case: Application
$[\![\, u\,s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]\,[\![\, s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} [\![\, u \,|\, \sigma' \,|\, \gamma \,]\!]\,[\![\, s \,|\, \sigma' \,|\, \gamma \,]\!] = [\![\, u\,s \,|\, \sigma' \,|\, \gamma \,]\!]$

Inductive Case: Abstraction
$[\![\, (c\langle c \rangle.s)\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = \lambda c.[\![\, s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} \lambda c.[\![\, s \,|\, \sigma' \,|\, \gamma \,]\!] = [\![\, c\langle c \rangle.s \,|\, \sigma' \,|\, \gamma \,]\!]$

Inductive Case: Phantom-Abstraction
$[\![\, (c\langle x_1, \ldots, x_n \rangle.s)\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]$

Case: $x \in \{x_1, \ldots, x_n\}$
$= [\![\, (c\langle x_1, \ldots, x_n, x \rangle.s)\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, c\langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle.s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]$
where $\{y_1, \ldots, y_m\} = (t)_{fv}$
$= \lambda c.[\![\, s\{t/x\} \,|\, \sigma'' \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} \lambda c.[\![\, s \,|\, \sigma_1''' \,|\, \gamma \,]\!] = \lambda c.[\![\, s \,|\, \sigma_2''' \,|\, \gamma \,]\!] = [\![\, c\langle x_1, \ldots, x_n, x \rangle.s \,|\, \sigma' \,|\, \gamma \,]\!]$

where $\sigma''(z) = \begin{cases} \sigma(z)\{c/\gamma(c)\} & \text{if } z \in \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \\ \sigma(z) & \text{otherwise} \end{cases}$

$\sigma_1''' = \sigma''[x \mapsto [\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]]$

$$\sigma_2'''(z) = \begin{cases} [\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]\{c/\gamma(c)\} & z = x \\ \sigma(z)\{c/\gamma(c)\} & z \in \{x_1, \ldots, x_n\} \\ \sigma(z) & \text{otherwise} \end{cases}$$

Case: $x \notin \{x_1, \ldots, x_n\}$
$= [\![\, c\langle x_1, \ldots, x_n \rangle.s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!] = \lambda c.[\![\, s\{t/x\} \,|\, \sigma'' \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} \lambda c.[\![\, t \,|\, \sigma''[x \mapsto [\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]] \,|\, \gamma \,]\!] = \lambda c.[\![\, t \,|\, \sigma''[x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]] \,|\, \gamma \,]\!] = [\![\, c\langle x_1, \ldots, x_n \rangle.s \,|\, \sigma[x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]] \,|\, \gamma \,]\!]$
where

$\sigma'' = \sigma[x_i \mapsto \sigma(x_i)\{c/\gamma(c)\}]_{i\in[n]}$

Inductive Case: Sharing
$[\![u[z_1,\dots,z_n \leftarrow s]\{t/x\} \,|\, \sigma \,|\, \gamma]\!] = [\![u\{t/x\}[z_1,\dots,z_n \leftarrow s\{t/x\}] \,|\, \sigma \,|\, \gamma]\!] = [\![u\{t/x\} \,|\, \sigma'' \,|\, \gamma]\!]$
$\overset{\text{I.H.}}{=} [\![u \,|\, \sigma''' \,|\, \gamma]\!] = [\![u[z_1,\dots,z_n \leftarrow s] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\sigma'' = \sigma[z_1 \mapsto [\![s\{t/x\} \,|\, \sigma \,|\, \gamma]\!],\dots,z_n \mapsto [\![s\{t/x\} \,|\, \sigma \,|\, \gamma]\!]]$
$\sigma''' = \sigma'[z_1 \mapsto [\![s \,|\, \sigma' \,|\, \gamma]\!],\dots,z_n \mapsto [\![s \,|\, \sigma' \,|\, \gamma]\!]]$

Inductive Case: Distributor 1
$[\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma]\!]$
$= [\![u\overline{[\Gamma]}\{t/x\} \,|\, \sigma \,|\, \gamma']\!] \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma']\!]$
$= [\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\gamma' = \gamma[e_1 \mapsto c,\dots,e_n \mapsto c]$

Inductive Case: Distributor 2
$[\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma]\!]$
$= [\![u\overline{[\Gamma]}\{t/x\} \,|\, \sigma'' \,|\, \gamma']\!] \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma']\!]$
$= [\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\gamma' = \gamma[e_1 \mapsto c,\dots,e_n \mapsto c]$ ◀

The proof for Proposition 13 (repeated here) is shown below. Book-keeping commutes with the translation in the following way

if $c\langle y_1,\dots,y_m \rangle. \in (u)_{fc}$ such that $\{x_1,\dots,x_n\} \subset \{y_1,\dots,y_m\}$
and for those $z \in \{y_1,\dots,y_m\}/\{x_1,\dots,x_n\}$, $\gamma(c) \notin (\sigma(z))_{fv}$
or if simply $\{x_1,\dots,x_n\} \cap (u)_{fv} = \{\}$

$$[\![u\{x_1,\dots,x_n/c\}_b \,|\, \sigma \,|\, \gamma]\!] = [\![u \,|\, \sigma \,|\, \gamma]\!]$$

Proof. We prove this by induction on u

Base Case: Variable
$[\![x\{x_1,\dots,x_n/c\}_b \,|\, \sigma \,|\, \gamma]\!] = [\![x \,|\, \sigma \,|\, \gamma]\!] = \sigma(x) = \sigma'(x) = [\![x \,|\, \sigma' \,|\, \gamma']\!]$
Since is cannot be that $x \in \{x_1,\dots,x_n\}$

Base Case: Phantom-Abstraction
$[\![(c\langle y_1,\dots,y_m \rangle.t)\{x_1,\dots,x_n/c\}_b \,|\, \sigma \,|\, \gamma]\!] = [\![c\langle x_1,\dots,x_n \rangle.t \,|\, \sigma \,|\, \gamma]\!]$
$= \lambda c.[\![t \,|\, \sigma'' \,|\, \gamma]\!] = \lambda c.[\![t \,|\, \sigma'' \,|\, \gamma']\!] = [\![c\langle y_1,\dots,y_m \rangle.t \,|\, \sigma' \,|\, \gamma']\!]$
where
$\sigma = \sigma_1[x_1 \mapsto M_1,\dots,x_n \mapsto M_n]$
$\sigma'' = \sigma_1[x_1 \mapsto M_1\{c/d\},\dots,x_n \mapsto M_n\{c/d\}]$
$\gamma(c) = d$
Note: due to condition of Proposition any $\{y_i \mapsto M_i\{c/d\}\} = \{y_i \mapsto M_i\}$

Base Case: Distributor
$[\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle y_1,\dots,y_m \rangle \overline{[\Gamma]}]\{x_1,\dots,x_n/c\}_b \,|\, \sigma \,|\, \gamma]\!]$
$= [\![u[e_1\langle \vec{w}_1 \rangle,\dots,e_n\langle \vec{w}_n \rangle \,|\, c\langle x_1,\dots,x_n \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!] = [\![u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma']\!]$

$= \llbracket u[e_1\langle \vec{w_1} \rangle, \ldots, e_n\langle \vec{w_n} \rangle \mid c\langle y_1, \ldots, y_m \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket$
where $\gamma' = \gamma[e_1 \mapsto c, \ldots, e_n \mapsto c]$
$\sigma = \sigma_1[x_1 \mapsto M_1, \ldots, x_n \mapsto M_n]$
$\sigma' = \sigma_1[x_1 \mapsto M_1\{c/\gamma(c)\}, \ldots, x_n \mapsto M_n\{c/\gamma(c)\}]$

Inductive Case: Application
$\llbracket (s\,t)\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \llbracket s\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket \llbracket t\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket$
$\overset{\text{I.H.}}{=} \llbracket s \mid \sigma \mid \gamma \rrbracket \llbracket t \mid \sigma \mid \gamma \rrbracket = \llbracket s\,t \mid \sigma \mid \gamma \rrbracket$

Inductive Case: Abstraction
$\llbracket (z\langle z \rangle.t)\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \lambda z.\llbracket t\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket \overset{\text{I.H.}}{=} \lambda z.\llbracket t \mid \sigma \mid \gamma \rrbracket = \llbracket z\langle z \rangle.t \mid \sigma \mid \gamma \rrbracket$

Inductive Case: Phantom-Abstraction
$\llbracket (d\langle z_1, \ldots, z_m \rangle.t)\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \lambda d.\llbracket t\{x_1, \ldots, x_n/c\}_b \mid \sigma' \mid \gamma \rrbracket$
$\overset{\text{I.H.}}{=} \lambda d.\llbracket t \mid \sigma' \mid \gamma \rrbracket = \llbracket d\langle z_1, \ldots, z_m \rangle.t \mid \sigma \mid \gamma \rrbracket$

Inductive Case: Sharing
$\llbracket u[z_1, \ldots, z_m \leftarrow t]\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\{x_1, \ldots, x_n/c\}_b[z_1, \ldots, z_m \leftarrow t\{x_1, \ldots, x_n/c\}_b] \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\{x_1, \ldots, x_n/c\}_b \mid \sigma' \mid \gamma \rrbracket \overset{\text{I.H.}}{=} \llbracket u \mid \sigma'' \mid \gamma \rrbracket = \llbracket u[z_1, \ldots, z_m \leftarrow t] \mid \sigma \mid \gamma \rrbracket$

Inductive Case: Distributor
$\llbracket u[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid d\langle d \rangle \overline{[\Gamma]}]\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\underline{[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid d\langle d \rangle \overline{[\Gamma]\{x_1, \ldots, x_n/c\}_b}]} \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\overline{[\Gamma]}\{x_1, \ldots, x_n/c\}_b \mid \sigma \mid \gamma' \rrbracket \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \mid \sigma \mid \gamma' \rrbracket$
$= \llbracket u[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid d\langle d \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket$ ◀

The proof for 14 (repeated here) is below. Exorcisms commute with the translation in the following way

if $c\langle x_1, \ldots, x_n \rangle. \in (u)_{fc}$ or $\{x_1, \ldots, x_n\} \cap (u)_{fv} = \{\}$

$$\llbracket u\{c\langle x_1, \ldots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x_i \mapsto c]_{i \in [n]} \mid \gamma \rrbracket$$

Proof. We prove this by induction on u

Base Case: Variable
$\llbracket z\{c\langle x_1, \ldots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \llbracket z \mid \sigma \mid \gamma \rrbracket = \sigma(z) = \sigma'(z) = \llbracket z \mid \sigma' \mid \gamma \rrbracket$

Base Case: Phantom-Abstraction
$\llbracket (c\langle x_1, \ldots, x_n \rangle.t)\{c\langle x_1, \ldots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \llbracket c\langle c \rangle.t[x_1, \ldots, x_n \leftarrow c] \mid \sigma \mid \gamma \rrbracket$
$= \lambda c.\llbracket t[x_1, \ldots, x_n \leftarrow c] \mid \sigma \mid \gamma \rrbracket = \lambda c.\llbracket t \mid \sigma' \mid \gamma \rrbracket = \llbracket c\langle x_1, \ldots, x_n \rangle.t \mid \sigma' \mid \gamma \rrbracket$

Base Case: Distributor
$\llbracket u[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid c\langle x_1, \ldots, x_n \rangle \overline{[\Gamma]}]\{c\langle x_1, \ldots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\underline{[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid c\langle c \rangle \overline{[\Gamma][x_1, \ldots, x_n \leftarrow c]}]} \mid \sigma \mid \gamma \rrbracket$
$= \llbracket u\overline{[\Gamma]}[x_1, \ldots, x_n \leftarrow c] \mid \sigma \mid \gamma' \rrbracket = \llbracket u\overline{[\Gamma]} \mid \sigma' \mid \gamma' \rrbracket$
$= \llbracket u[e_1\langle \vec{w_1} \rangle, \ldots, e_m\langle \vec{w_m} \rangle \mid c\langle x_1, \ldots, x_n \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket$

Inductive Case: Application

$\llbracket (s\,t)\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket = \llbracket s\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket \, \llbracket t\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket$
$\overset{\text{I.H.}}{=} \llbracket s \,|\, \sigma' \,|\, \gamma \rrbracket \, \llbracket t \,|\, \sigma' \,|\, \gamma \rrbracket = \llbracket s\,t \,|\, \sigma' \,|\, \gamma \rrbracket$

Inductive Case: Abstraction
$\llbracket (z\langle\, z\,\rangle.t)\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket = \lambda z.\llbracket t\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket$
$\overset{\text{I.H.}}{=} \lambda z.\llbracket t \,|\, \sigma' \,|\, \gamma \rrbracket = \llbracket z\langle\, z\,\rangle.t \,|\, \sigma' \,|\, \gamma \rrbracket$

Inductive Case: Phantom-Abstraction
$\llbracket (d\langle\, z_1,\ldots,z_m\,\rangle.t)\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket = \lambda d.\llbracket t\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma'' \,|\, \gamma \rrbracket$
$\overset{\text{I.H.}}{=} \lambda d.\llbracket t \,|\, \sigma''' \,|\, \gamma \rrbracket = \llbracket d\langle\, z_1,\ldots,z_m\,\rangle.t \,|\, \sigma' \,|\, \gamma \rrbracket$

Inductive Case: Sharing
$\llbracket u[z_1,\ldots,z_m \leftarrow t]\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket$
$= \llbracket u\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e[z_1,\ldots,z_m \leftarrow t\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e] \,|\, \sigma \,|\, \gamma \rrbracket$
$= \llbracket u\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma'' \,|\, \gamma \rrbracket \overset{\text{I.H.}}{=} \llbracket u \,|\, \sigma''' \,|\, \gamma \rrbracket = \llbracket u[z_1,\ldots,z_m \leftarrow t] \,|\, \sigma' \,|\, \gamma \rrbracket$

Inductive Case: Distributor
$\llbracket u[e_1\langle\, \vec{w_1}\,\rangle,\ldots,e_m\langle\, \vec{w_m}\,\rangle \,|\, d\langle\, d\,\rangle\, \overline{[\Gamma]}]\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket$
$= \llbracket u[e_1\langle\, \vec{w_1}\,\rangle,\ldots,e_m\langle\, \vec{w_m}\,\rangle \,|\, d\langle\, d\,\rangle\, \overline{[\Gamma]}\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e] \,|\, \sigma \,|\, \gamma \rrbracket$
$= \llbracket u\overline{[\Gamma]}\{c\langle\, x_1,\ldots,x_n\,\rangle\}_e \,|\, \sigma \,|\, \gamma' \rrbracket \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma' \rrbracket$
$= \llbracket u[e_1\langle\, \vec{w_1}\,\rangle,\ldots,e_m\langle\, \vec{w_m}\,\rangle \,|\, d\langle\, d\,\rangle\, \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma' \rrbracket$ ◀

We prove Lemma 16 on a case by case basis. If $s \rightsquigarrow_{L,D,C} t$ then $\llbracket s \,|\, \sigma \,|\, \gamma \rrbracket = \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket$

Proof. We prove this by induction. First we to a case-by-case basis for the base case.

Case: (c_1)
$$u[\vec{w} \leftarrow y][\vec{x}\cdot y \leftarrow t] \rightsquigarrow_C u[\vec{x}\cdot\vec{w} \leftarrow t]$$
$\llbracket u[\vec{w} \leftarrow y][\vec{x}\cdot y \leftarrow t] \,|\, \sigma \,|\, \gamma \rrbracket = \llbracket u[\vec{w} \leftarrow y] \,|\, \sigma' \,|\, \gamma \rrbracket = \llbracket u \,|\, \sigma'' \,|\, \gamma \rrbracket = \llbracket u[\vec{x}\cdot\vec{w} \leftarrow t] \,|\, \sigma \,|\, \gamma \rrbracket$
where
$\sigma' = \sigma[x \mapsto \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket]_{\forall x\in\vec{x}}[y \mapsto \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket]$
$\sigma'' = \sigma'[w \mapsto \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket]_{\forall w\in\vec{w}}$

Case: (c_2)
$$u[x \leftarrow t] \rightsquigarrow_C u\{t/x\}$$
$\llbracket u[x \leftarrow t] \,|\, \sigma \,|\, \gamma \rrbracket = \llbracket u \,|\, \sigma[x \mapsto \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket] \,|\, \gamma \rrbracket = \llbracket u\{t/x\} \,|\, \sigma \,|\, \gamma \rrbracket$

Case: (d_1)
$$u[x_1\ldots x_n \leftarrow s\,t] \rightsquigarrow_D u\{z_1\,y_1/x_1\}\ldots\{z_n\,y_n/x_n\}[z_1\ldots z_n \leftarrow s][y_1\ldots y_n \leftarrow t]$$
$\llbracket u[x_1\ldots x_n \leftarrow s\,t] \,|\, \sigma \,|\, \gamma \rrbracket = \llbracket u \,|\, \sigma' \,|\, \gamma \rrbracket$
where
$\sigma' = \sigma[x_i \mapsto \llbracket s\,t \,|\, \sigma \,|\, \gamma \rrbracket]_{1\le i\le n} = \sigma[x_i \mapsto \llbracket s \,|\, \sigma \,|\, \gamma \rrbracket \, \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket]_{1\le i\le n}$

$\llbracket u\{z_1\,y_1/x_1\}\ldots\{z_n\,y_n/x_n\}[z_1\ldots z_n \leftarrow s][y_1\ldots y_n \leftarrow t] \,|\, \sigma \,|\, \gamma \rrbracket$
$= \llbracket u\{z_1\,y_1/x_1\}\ldots\{z_n\,y_n/x_n\} \,|\, \sigma'' \,|\, \gamma \rrbracket$
where

$\sigma'' = \sigma[z_i \mapsto [\![\, s \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}[y_i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$ since $y_i \notin (s)_{fv}$
$= [\![\, u \,|\, \sigma''' \,|\, \gamma \,]\!]$
where
$\sigma''' = \sigma''[x_i \mapsto [\![\, z_i\, y_i \,|\, \sigma'' \,|\, \gamma \,]\!]]_{1 \leq i \leq n} = \sigma[x_i \mapsto [\![\, s \,|\, \sigma \,|\, \gamma \,]\!]\, [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$
since z_i and $y_i \notin (u)_{fv}$

Case: (d_2)

$$u[x_1, \ldots, x_n \leftarrow c\langle\, \vec{y} \,\rangle.t] \rightsquigarrow_D$$

$$u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle\, w_1^1 \,\rangle \ldots e_n\langle\, w_1^n \,\rangle \,|\, c\langle\, \vec{y} \,\rangle\, [w_1^1, \ldots, w_1^n \leftarrow t]]$$

SubCase: $\vec{y} = c$
$[\![\, u[x_1, \ldots, x_n \leftarrow c\langle\, c \,\rangle.t] \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma' \,|\, \gamma \,]\!]$
where $\sigma' = \sigma[x_i \mapsto [\![\, c\langle\, c \,\rangle.t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n} = \sigma[x_i \mapsto \lambda c.[\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$

$[\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle\, w_1^1 \,\rangle \ldots e_n\langle\, w_1^n \,\rangle \,|\, c\langle\, c \,\rangle\, [w_1^1, \ldots, w_1^n \leftarrow t]] \,|\, \sigma \,|\, \gamma \,]\!]$
$[\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \ldots, w_1^n \leftarrow t] \,|\, \sigma \,|\, \gamma' \,]\!]$
$= [\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n} \,|\, \sigma' \,|\, \gamma' \,]\!]$
where
$\gamma' = \gamma[e_1 \mapsto c, \ldots, e_n \mapsto c]$
$\sigma' = \sigma[w_1^i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma' \,]\!]]_{1 \leq i \leq n} = \sigma[w_1^i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$
$= [\![\, u \,|\, \sigma'' \,|\, \gamma' \,]\!] = [\![\, u \,|\, \sigma'' \,|\, \gamma \,]\!]$
where
$\sigma'' = \sigma'[x_i \mapsto [\![\, e_i\langle\, w_1^i \,\rangle.w_1^i \,|\, \sigma' \,|\, \gamma' \,]\!]]_{1 \leq i \leq n} = \sigma'[x_i \mapsto \lambda e_i.[\![\, w_1^i \,|\, \sigma_i' \,|\, \gamma' \,]\!]]_{1 \leq i \leq n}$
$\quad = \sigma'[x_i \mapsto \lambda e_i.[\![\, t \,|\, \sigma \,|\, \gamma \,]\!]\{e_i/c\}]_{1 \leq i \leq n} =_\alpha \sigma'[x_i \mapsto \lambda c.[\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$
$\quad = \sigma[x_i \mapsto \lambda c.[\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$ since $w_1^i \notin (u)_{fv}$

SubCase: $\vec{y} = \{y_1, \ldots, y_m\}$
$[\![\, u[x_1, \ldots, x_n \leftarrow c\langle\, y_1, \ldots, y_m \,\rangle.t] \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma' \,|\, \gamma \,]\!]$
where
$\sigma' = \sigma[x_i \mapsto [\![\, c\langle\, y_1, \ldots, y_m \,\rangle.t \,|\, \sigma \,|\, \gamma \,]\!]]_{1 \leq i \leq n} = \sigma[x_i \mapsto \lambda c.[\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$
$\sigma = \sigma_1[y_1 \mapsto M_1, \ldots, y_m \mapsto M_m]$
$\sigma'' = \sigma_1[y_1 \mapsto M_1\{c/\gamma(c)\}, \ldots, y_m \mapsto M_m\{c/\gamma(c)\}]$

$[\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle\, w_1^1 \,\rangle \ldots e_n\langle\, w_1^n \,\rangle \,|\, c\langle\, y_1, \ldots, y_m \,\rangle\, [w_1^1, \ldots, w_1^n \leftarrow t]] \,|\, \sigma \,|\, \gamma \,]\!]$
$[\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \ldots, w_1^n \leftarrow t] \,|\, \sigma'' \,|\, \gamma' \,]\!]$
where $\gamma' = \gamma[e_1 \mapsto c, \ldots, e_n \mapsto c]$
$= [\![\, u\{e_i\langle\, w_1^i \,\rangle.w_1^i/x_i\}_{1 \leq i \leq n} \,|\, \sigma''' \,|\, \gamma' \,]\!]$
where $\sigma''' = \sigma''[w_1^i \mapsto [\![\, t \,|\, \sigma'' \,|\, \gamma' \,]\!]]_{1 \leq i \leq n} = \sigma''[w_1^i \mapsto [\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$
$= [\![\, u \,|\, \sigma'''' \,|\, \gamma' \,]\!] = [\![\, u \,|\, \sigma'''' \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma'' \,|\, \gamma \,]\!]$
where $\sigma'''' = \sigma'''[x_i \mapsto [\![\, e_i\langle\, w_1^i \,\rangle.w_1^i \,|\, \sigma''' \,|\, \gamma' \,]\!]]_{1 \leq i \leq n} = \sigma'''[x_i \mapsto \lambda e_i.[\![\, w_1^i \,|\, \sigma_i''' \,|\, \gamma' \,]\!]]_{1 \leq i \leq n}$
$\quad = \sigma'''[x_i \mapsto \lambda e_i.[\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]\{e_i/\gamma'(e_i)\}]_{1 \leq i \leq n} =_\alpha \sigma'''[x_i \mapsto \lambda c.[\![\, t \,|\, \sigma'' \,|\, \gamma \,]\!]]_{1 \leq i \leq n}$

Case: (d_3)

$$u[e_1\langle\, \vec{w}_1 \,\rangle \ldots e_n\langle\, \vec{w}_n \,\rangle \,|\, c\langle\, c \,\rangle\, [\vec{w}_1, \ldots, \vec{w}_n \leftarrow c]] \rightsquigarrow_D u\{e_1\langle\, \vec{w}_1 \,\rangle\}_e \ldots \{e_n\langle\, \vec{w}_n \,\rangle\}_e$$

$[\![\, u[e_1\langle\, \vec{w}_1 \,\rangle \ldots e_n\langle\, \vec{w}_n \,\rangle \,|\, c\langle\, c \,\rangle\, [\vec{w}_1, \ldots, \vec{w}_n \leftarrow c]] \,|\, \sigma \,|\, \gamma \,]\!]$
$= [\![\, u[\vec{w}_1, \ldots, \vec{w}_n \leftarrow c] \,|\, \sigma \,|\, \gamma' \,]\!] = [\![\, u \,|\, \sigma' \,|\, \gamma' \,]\!]$

$= [\![u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e \,|\, \sigma \,|\, \gamma']\!] = [\![u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]$

For the remaining cases, we say $[\![t[\Gamma] \,|\, \sigma \,|\, \gamma]\!]$ produces $[\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]$ where σ_Γ and γ_Γ are the resulting maps from interpreting the closure $[\Gamma]$

Case: (l_1)

$$s[\Gamma]\, t \rightsquigarrow_L (s\, t)[\Gamma]$$

$[\![s[\Gamma]\, t \,|\, \sigma \,|\, \gamma]\!] = [\![s \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]\, [\![t \,|\, \sigma \,|\, \gamma]\!] = [\![s \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]\, [\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![(s\, t)[\Gamma] \,|\, \sigma \,|\, \gamma]\!]$

Case: (l_2)

$$s[\Gamma]\, t \rightsquigarrow_L (s\, t)[\Gamma]$$

$[\![s\,(t[\Gamma]) \,|\, \sigma \,|\, \gamma]\!] = [\![s \,|\, \sigma \,|\, \gamma]\!]\, [\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![s \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]\, [\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![(s\, t)[\Gamma] \,|\, \sigma \,|\, \gamma]\!]$

Case: (l_3)

$$d\langle \vec{x} \rangle.t[\Gamma] \rightsquigarrow_L (d\langle \vec{x} \rangle.t)[\Gamma]$$

SubCase: $\vec{x} = d$

$[\![d\langle d \rangle.t[\Gamma] \,|\, \sigma \,|\, \gamma]\!] = \lambda d.[\![t[\Gamma] \,|\, \sigma \,|\, \gamma]\!] = \lambda d.[\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![d\langle d \rangle.t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![(d\langle d \rangle.t)[\Gamma] \,|\, \sigma \,|\, \gamma]\!]$

SubCase: $\vec{x} = x_1, \dots, x_n$

$[\![d\langle x_1, \dots, x_n \rangle.t[\Gamma] \,|\, \sigma \,|\, \gamma]\!] = \lambda d.[\![t[\Gamma] \,|\, \sigma' \,|\, \gamma]\!] = \lambda d.[\![t \,|\, \sigma'_\Gamma \,|\, \gamma_\Gamma]\!] = [\![d\langle x_1, \dots, x_n \rangle.t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]$
$= [\![(d\langle x_1, \dots, x_n \rangle.t)[\Gamma] \,|\, \sigma \,|\, \gamma]\!]$
since we know $x_1, \dots, x_n \notin ([\Gamma])_{fv}$

Case: (l_4)

$$u[\vec{x} \leftarrow t[\Gamma]] \rightsquigarrow_L u[\vec{x} \leftarrow t][\Gamma]$$

$[\![u[\vec{x} \leftarrow t[\Gamma]] \,|\, \sigma \,|\, \gamma]\!] = [\![u \,|\, \sigma' \,|\, \gamma]\!] = [\![u \,|\, \sigma'' \,|\, \gamma_\Gamma]\!] = [\![u[\vec{x} \leftarrow t] \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!] = [\![u[\vec{x} \leftarrow t[\Gamma]] \,|\, \sigma \,|\, \gamma]\!]$
where
$\sigma' = \sigma[x \mapsto [\![t[\Gamma] \,|\, \sigma \,|\, \gamma]\!]]_{\forall x \in \vec{x}} = \sigma[x \mapsto [\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]]_{\forall x \in \vec{x}}$
$\sigma'' = \sigma_\Gamma[x \mapsto [\![t \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]]_{\forall x \in \vec{x}}$

Cases: (l_5)

$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle\, \overline{[\Gamma]}[\Gamma]] \rightsquigarrow_L$$
$$u\{(\vec{w}_i/\vec{z})/e_i\}_{b_{i \in [n]}}[e_1\langle \vec{w}_1/\vec{z} \rangle \dots e_n\langle \vec{w}_n/\vec{z} \rangle \,|\, c\langle \vec{x} \rangle\, \overline{[\Gamma]}][\Gamma]$$

SubCase: $\vec{x} = c$

$[\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle\, \overline{[\Gamma]}[\Gamma]] \,|\, \sigma \,|\, \gamma]\!] = [\![u\overline{[\Gamma]}[\Gamma] \,|\, \sigma \,|\, \gamma']\!] = [\![u\overline{[\Gamma]} \,|\, \sigma_\Gamma \,|\, \gamma'_\Gamma]\!]$
$= [\![u\overline{[\Gamma]}\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \,|\, \sigma_\Gamma \,|\, \gamma'_\Gamma]\!] = [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \overline{[\Gamma]} \,|\, \sigma_\Gamma \,|\, \gamma'_\Gamma]\!]$
$= [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \,|\, c\langle c \rangle\, \overline{[\Gamma]}] \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]$
$= [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \,|\, c\langle c \rangle\, \overline{[\Gamma]}][\Gamma] \,|\, \sigma \,|\, \gamma]\!]$

SubCase: $\vec{x} = x_1, \dots, x_m$

$[\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle x_1, \dots, x_m \rangle\, \overline{[\Gamma]}[\Gamma]] \,|\, \sigma \,|\, \gamma]\!]$
$= [\![u\overline{[\Gamma]}[\Gamma] \,|\, \sigma' \,|\, \gamma']\!] = [\![u\overline{[\Gamma]} \,|\, \sigma'_\Gamma \,|\, \gamma'_\Gamma]\!] = [\![u\overline{[\Gamma]}\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \,|\, \sigma_\Gamma \,|\, \gamma'_\Gamma]\!]$
$= [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \overline{[\Gamma]} \,|\, \sigma_\Gamma \,|\, \gamma'_\Gamma]\!]$
$= [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \,|\, c\langle x_1, \dots, x_n \rangle\, \overline{[\Gamma]}] \,|\, \sigma_\Gamma \,|\, \gamma_\Gamma]\!]$
$= [\![u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \,|\, c\langle x_1, \dots, x_n \rangle\, \overline{[\Gamma]}][\Gamma] \,|\, \sigma \,|\, \gamma]\!]$

Inductive Case: Application $t \rightsquigarrow_{(C,D,L)} t'$

$[\![\, t\, s \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]\, [\![\, s \,|\, \sigma \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} [\![\, t' \,|\, \sigma \,|\, \gamma \,]\!]\, [\![\, s \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, t'\, s \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Application $s \rightsquigarrow_{(C,D,L)} s'$

$[\![\, t\, s \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]\, [\![\, s \,|\, \sigma \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]\, [\![\, s' \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, t\, s' \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Abstraction $t \rightsquigarrow_{(C,D,L)} t'$

$[\![\, x\langle\, x \,\rangle.t \,|\, \sigma \,|\, \gamma \,]\!] = \lambda x.[\![\, t \,|\, \sigma[x \mapsto x] \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} \lambda x.[\![\, t' \,|\, \sigma[x \mapsto x] \,|\, \gamma \,]\!] = [\![\, x\langle\, x \,\rangle.t' \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Phantom-Abstraction $t \rightsquigarrow_{(C,D,L)} t'$

$[\![\, c\langle\, \vec{x} \,\rangle.t \,|\, \sigma \,|\, \gamma \,]\!] = \lambda c.[\![\, t \,|\, \sigma' \,|\, \gamma \,]\!] \overset{\text{I.H.}}{=} \lambda c.[\![\, t' \,|\, \sigma' \,|\, \gamma \,]\!] = [\![\, c\langle\, \vec{x} \,\rangle.t' \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Sharing $t \rightsquigarrow_{(C,D,L)} t'$

$[\![\, u[x_1, \ldots, x_n \leftarrow t] \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma[x_i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{i \in [n]} \,|\, \gamma \,]\!]$

$\overset{\text{I.H.}}{=} [\![\, u \,|\, \sigma[x_i \mapsto [\![\, t' \,|\, \sigma \,|\, \gamma \,]\!]]_{i \in [n]} \,|\, \gamma \,]\!] = [\![\, u[x_1, \ldots, x_n \leftarrow t'] \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Sharing $u \rightsquigarrow_{(C,D,L)} u'$

$[\![\, u[x_1, \ldots, x_n \leftarrow t] \,|\, \sigma \,|\, \gamma \,]\!] = [\![\, u \,|\, \sigma[x_i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{i \in [n]} \,|\, \gamma \,]\!]$

$\overset{\text{I.H.}}{=} [\![\, u' \,|\, \sigma[x_i \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]]_{i \in [n]} \,|\, \gamma \,]\!] = [\![\, u'[x_1, \ldots, x_n \leftarrow t] \,|\, \sigma \,|\, \gamma \,]\!]$

Inductive Case: Distributor $u[\overrightarrow{e\langle\, \vec{x} \,\rangle} \,|\, c\langle\, c \,\rangle\, \overline{[\Gamma]}] \rightsquigarrow_{(C,D,L)} u'[\overrightarrow{e\langle\, \vec{x}' \,\rangle} \,|\, c\langle\, c \,\rangle\, \overline{[\Gamma']}]$

$[\![\, u[\overrightarrow{e\langle\, \vec{x} \,\rangle} \,|\, c\langle\, c \,\rangle\, \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma \,]\!] \;=\; [\![\, u\overline{[\Gamma]} \,|\, \sigma \,|\, \gamma' \,]\!] \;\overset{\text{I.H.}}{=}\; [\![\, u'\overline{[\Gamma']} \,|\, \sigma \,|\, \gamma' \,]\!] \;=\; [\![\, u'[\overrightarrow{e\langle\, \vec{x}' \,\rangle} \,|\, c\langle\, c \,\rangle\, \overline{[\Gamma']}] \,|\, \sigma \,|\, \gamma \,]\!]$

◀

B Strong Normalisation of Sharing Reductions

The weakening calculus is used to show preservation of strong normalisation with respect to the λ-calculus. A β-step in our calculus may occur within a weakening, and therefore is simulated by zero β-steps in the λ-calculus. Therefore if there is an infinite reduction path located inside a weakening in Λ_a^S, then the reduction path is not preserved in the corresponding λ-term as there are no weakenings. To deal with this, just as done in [2, 15, 16], we make use of the weakening calculus. A β-step is non-deleteing precisely because of the weakening construct. If a β-step would be deleting in the λ-calculus, then the weakening calculus would instead keep the deleted term around as 'garbage', which can continue to reduce unless explicitly 'garbage-collected' by extra (non-β) reduction steps. The weakening calculus has already been shown to preserve strong normalisation through the use of a perpetual strategy in [15]. A part of proving PSN is then using the weakening calculus to prove that if $t \in \Lambda_a^S$ has a infinite reduction path, then its translation into the weakening calculus also has an infinite reduction path.

We demonstrate that our readback translation (Definition 18) is truly an extention of the translation into the λ-calculus (Definition 9). We therefore demonstrate that our operations (substitution, book-keeping, and exorcisms) commute with the two translation functions in the same way.

▶ **Proposition 41.** *Given $M \in \Lambda$ such that for all $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$, the translation $[\![\, u \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$ commutes with substitution $\{M/x\}$ in the following way*

$$[\![\, u\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}} = [\![\, u \,|\, \sigma[x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}] \,|\, \gamma \,]\!]_{\mathcal{W}}$$

Proof. We prove this by induction on u. The argument is similar to the proof of Proposition 41. We only discuss here to cases involving the three special cases defined in Definition 18.

Inductive Case: Weakening

$[\![\, u[\leftarrow s]\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}} = [\![\, u\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}[\leftarrow [\![\, s\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}]$

$\overset{\text{I.H.}}{=} [\![\, u \,|\, \sigma' \,|\, \gamma \,]\!]_{\mathcal{W}}[\leftarrow [\![\, s \,|\, \sigma' \,|\, \gamma \,]\!]_{\mathcal{W}}] = [\![\, u[\leftarrow s] \,|\, \sigma' \,|\, \gamma \,]\!]_{\mathcal{W}}$

Inductive Case: Distributor

$[\![\, u[\,|\, c\langle\, \vec{x} \,\rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$

SubCase: $\vec{x} = c$

$[\![\, u[\,|\, c\langle\, c \,\rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}} = [\![\, u[\,|\, c\langle\, c \,\rangle \overline{[\Gamma]}\{t/x\}] \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$

$= [\![\, u\overline{[\Gamma]}\{t/x\} \,|\, \sigma'' \,|\, \gamma' \,]\!]_{\mathcal{W}} \overset{\text{I.H.}}{=} [\![\, u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma' \,]\!]_{\mathcal{W}} = [\![\, u[\,|\, c\langle\, c \,\rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma \,]\!]_{\mathcal{W}}$

where

$\sigma'' = \sigma[c \mapsto \bullet]$

$\sigma''' = \sigma[c \mapsto \bullet][x \mapsto [\![\, t \,|\, \sigma'' \,|\, \gamma' \,]\!]_{\mathcal{W}}] = \sigma[c \mapsto \bullet][x \mapsto [\![\, t \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}]$

SubCase: $\vec{x} = x_1, \ldots, x_n$

$[\![\, u[\,|\, c\langle\, x_1, \ldots, x_n \,\rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$

SubSubCase: $\vec{x} = x_1, \ldots, x_n, x$

$[\![\, u[\,|\, c\langle\, x_1, \ldots, x_n, x \,\rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$

$[\![\, u[\,|\, c\langle\, x_1, \ldots, x_n, y_1, \ldots, y_m \,\rangle \overline{[\Gamma]}\{t/x\}] \,|\, \sigma \,|\, \gamma \,]\!]_{\mathcal{W}}$

where $\{y_1, \ldots, y_m\} = (t)_{fv}$

$= [\![\, u\overline{[\Gamma]}\{t/x\} \,|\, \sigma'' \,|\, \gamma \,]\!]_{\mathcal{W}}$

where
$\sigma = \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n, y_1 \mapsto N_1, \dots, y_m \mapsto N_m]$
$\sigma'' = \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}, y_1 \mapsto N_1\{\bullet/\gamma(c)\}, \dots, y_m \mapsto N_m\{\bullet/\gamma(c)\}]$
$\overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, c\langle x_1, \dots, x_n, x \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}}$
where $\sigma''' = \sigma''[x \mapsto \llbracket t \,|\, \sigma'' \,|\, \gamma \rrbracket_{\mathcal{W}}] = \sigma''[x \mapsto \llbracket t \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}}\{\bullet/\gamma(c)\}]$
since $\{y_1, \dots, y_m\} = (t)_{fv}$

SubSubCase: $\vec{x} = x_1, \dots, x_n$
$\llbracket u[\,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}]\{t/x\} \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}\{t/x\}] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
$\llbracket u\overline{[\Gamma]}\{t/x\} \,|\, \sigma'' \,|\, \gamma \rrbracket_{\mathcal{W}} \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}}$
$\sigma = \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n]$
$\sigma'' = \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}]$
$\sigma''' = \sigma''[x \mapsto \llbracket t \,|\, \sigma'' \,|\, \gamma \rrbracket_{\mathcal{W}}] = \sigma''[x \mapsto \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}]$
since $\{x_1, \dots, x_n\} \cap (t)_{fv} = \{\}$ ◀

▶ **Proposition 42.** *Book-keeping commutes with the translation in the following way*
if $c\langle y_1, \dots, y_m \rangle. \in (u)_{fc}$ *such that* $\{x_1, \dots, x_n\} \subset \{y_1, \dots, y_m\}$
and for those $z \in \{y_1, \dots, y_m\}/\{x_1, \dots, x_n\}$, $\gamma(c) \notin (\sigma(z))_{fv}$
or if simply $\{x_1, \dots, x_n\} \cap (u)_{fv} = \{\}$

$$\llbracket u\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$$

Proof. We prove this by induction on u. The argument is similar to the proof of Proposition 13. We only discuss here to cases involving the three special cases defined in Definition 18.

Inductive Case: Weakening
$\llbracket u[\leftarrow t]\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}[\leftarrow \llbracket t\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}]$
$\overset{\text{I.H.}}{=} \llbracket u \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}[\leftarrow \llbracket t \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}] = \llbracket u[\leftarrow t] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$

Base Case: Distributor
$\llbracket u[\,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}]\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
$\llbracket u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
where $\sigma' = \sigma[x_1 \mapsto \sigma(x_1)\{\bullet/\gamma(c)\}, \dots, x_n \mapsto \sigma(x_n)\{\bullet/\gamma(c)\}]$
and notice for $x_i \neq y \in \vec{x}$, $[y \mapsto N] = [y \mapsto N\{\bullet/\gamma(c)\}]$

Inductive Case: Distributor
$\llbracket u[\,|\, d\langle d \rangle \overline{[\Gamma]}]\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, d\langle d \rangle \overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
$\llbracket u\overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}} \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, d\langle d \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
where $\sigma' = \sigma[d \mapsto \bullet]$

$\llbracket u[\,|\, d\langle z_1, \dots, z_n \rangle \overline{[\Gamma]}]\{x_1, \dots, x_n/c\}_b \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, d\langle z_1, \dots, z_n \rangle \overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
$\llbracket u\overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}} \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\,|\, d\langle z_1, \dots, z_n \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}}$
where
$\sigma' = \sigma[z_1 \mapsto \sigma(x_1)\{\bullet/\gamma(d)\}, \dots, z_n \mapsto \sigma(x_n)\{\bullet/\gamma(d)\}]$ ◀

▶ **Proposition 43.** *Exorcisms commute with the translation in the following way*
if $c\langle x_1, \dots, x_n \rangle. \in (u)_{fc}$ *or* $\{x_1, \dots, x_n\} \cap (u)_{fv} = \{\}$

$$\llbracket u\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma \rrbracket_{\mathcal{W}} = \llbracket u \,|\, \sigma' \,|\, \gamma \rrbracket_{\mathcal{W}}$$

where

$\sigma' = \sigma \cup \{x_1 \mapsto c, \dots, x_n \mapsto c\}$

Proof. We prove this by induction on u. The argument is similar to the proof of Proposition 14. We only discuss here to cases involving the three special cases defined in Definition 18.

Inductive Case: Weakening

$[\![u[\leftarrow t]\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$

$\stackrel{\text{I.H.}}{=} [\![u \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}] = [\![u[\leftarrow t] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}$

Base Case: Distributor

$[\![u[\,|\, c\langle x_1, \dots, x_n \rangle\, \overline{[\Gamma]}]\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, c\langle c \rangle\, \overline{[\Gamma]}[x_1, \dots, x_n \leftarrow c]] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

$= [\![u\overline{[\Gamma]}[x_1, \dots, x_n \leftarrow c] \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, c\langle x_1, \dots, x_n \rangle\, \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}$

where

$\sigma'' = \sigma[c \mapsto \bullet]$

$\sigma''' = \sigma[x_1 \mapsto \bullet, \dots, x_n \mapsto \bullet]$

Inductive Case: Distributor

$[\![u[\,|\, d\langle d \rangle\, \overline{[\Gamma]}]\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, d\langle d \rangle\, \overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

$= [\![u\overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} \stackrel{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, d\langle d \rangle\, \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}$

where

$\sigma'' = \sigma[d \mapsto \bullet]$

$\sigma''' = \sigma''[x_1 \mapsto c, \dots, x_n \mapsto c]$

$[\![u[\,|\, d\langle z_1, \dots, z_m \rangle\, \overline{[\Gamma]}]\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

$= [\![u[\,|\, d\langle z_1, \dots, z_m \rangle\, \overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

$= [\![u\overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} \stackrel{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, d\langle d \rangle\, \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}$

where

$\sigma'' = \sigma[z_1 \mapsto \sigma(z_1)\{\bullet/\gamma(d)\}, \dots, z_m \mapsto \sigma(z_m)\{\bullet/\gamma(d)\}]$

$\sigma''' = \sigma''[x_1 \mapsto c, \dots, x_n \mapsto c]$ ◀

Some of our proofs in the future also extract substitutions out of the map σ and apply them to the resulting term. We use the following proposition to demonstrate how we do this. We use $\sigma\{M/x\}$ to denote for all variables z, $\sigma\{M/x\}(z) = \sigma(z)\{M/x\}$.

▶ **Proposition 44.** *Given $M \in \Lambda_{\mathcal{W}}$ such that for all $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$*

$$[\![u \,|\, \sigma' \,|\, \gamma]\!] = [\![u \,|\, \sigma \,|\, \gamma]\!]\{M/x\}$$

where $\sigma' = (\sigma\{M/x\})[x \mapsto M]$

Proof. We prove this by induction on u

Base Case: Variable

$[\![x \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = x\{M/x\} = M = [\![x \,|\, \sigma' \,|\, \gamma]\!]$

$[\![y \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = N\{M/x\} = [\![y \,|\, \sigma' \,|\, \gamma]\!]$

Inductive Case: Application

$[\![s\, t \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = [\![s \,|\, \sigma \,|\, \gamma]\!]\{M/x\}\, [\![t \,|\, \sigma \,|\, \gamma]\!]\{M/x\} \stackrel{\text{I.H.}}{=} [\![s \,|\, \sigma \,|\, \gamma]\!]\, [\![t \,|\, \sigma' \,|\, \gamma]\!] = [\![s\, t \,|\, \sigma' \,|\, \gamma]\!]$

Inductive Case: Abstraction
$[\![c\langle c \rangle.t \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = \lambda c.[\![t \,|\, \sigma \,|\, \gamma]\!]\{M/x\} \overset{\text{I.H.}}{=} \lambda c.[\![t \,|\, \sigma' \,|\, \gamma]\!] = [\![c\langle c \rangle.t \,|\, \sigma' \,|\, \gamma]\!]$

Inductive Case: Phantom-Abstraction
$[\![c\langle x_1, \ldots, x_n \rangle.t \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = (\lambda c.[\![t \,|\, \sigma'' \,|\, \gamma]\!])\{M/x\} = \lambda c.[\![t \,|\, \sigma'' \,|\, \gamma]\!]\{M/x\} \overset{\text{I.H.}}{=} \lambda c.[\![t \,|\, \sigma''' \,|\, \gamma]\!]$
$= [\![c\langle x_1, \ldots, x_n \rangle.t \,|\, \sigma' \,|\, \gamma]\!]$
where
$\sigma'' = \sigma[x_1 \mapsto \sigma(x_1)\{c/d\}, \ldots, x_n \mapsto \sigma(x_n)\{c/d\}]$
$\sigma''' = \sigma''\{M/x\}[x \mapsto M]$
$\sigma''' = \sigma\{M/x\}[x_1 \mapsto \sigma(x_1)\{M/x\}\{c/d\}, \ldots, x_n \mapsto \sigma(x_n)\{M/x\}\{c/d\}, x \mapsto M]$

Inductive Case: Sharing
$[\![u[z_1, \ldots, z_n \leftarrow t] \,|\, \sigma \,|\, \gamma]\!]\{M/x\} = [\![u \,|\, \sigma'' \,|\, \gamma]\!]\{M/x\} \overset{\text{I.H.}}{=} [\![u \,|\, \sigma''' \,|\, \gamma]\!] = [\![u[z_1, \ldots, z_n \leftarrow t] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\sigma'' = \sigma[z_i \mapsto [\![t \,|\, \sigma \,|\, \gamma]\!]]_{i \in [n]}$
$\sigma''' = \sigma\{M/x\}[z_i \mapsto [\![t \,|\, \sigma\{x/M\}[x \mapsto M] \,|\, \gamma]\!], x \mapsto M]_{i \in [n]}$

Inductive Case: Distributor 1
$[\![u[e_1\langle \vec{w}_1 \rangle, \ldots, e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]\{M/x\}$
$= [\![u\overline{[\Gamma]} \,|\, \sigma \,|\, \gamma']\!]\{M/x\} \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma']\!]$
$= [\![u[e_1\langle \vec{w}_1 \rangle, \ldots, e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\gamma' = \gamma[e_1 \mapsto c, \ldots, e_n \mapsto c]$

Inductive Case: Distributor 2
$[\![u[e_1\langle \vec{w}_1 \rangle, \ldots, e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]\{M/x\}$
$= [\![u\overline{[\Gamma]} \,|\, \sigma'' \,|\, \gamma']\!]\{M/x\} \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma']\!]$
$= [\![u[e_1\langle \vec{w}_1 \rangle, \ldots, e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]$
where
$\gamma' = \gamma[e_1 \mapsto c, \ldots, e_n \mapsto c]$

Inductive Case: Weakening
$[\![u[\leftarrow t] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma' \,|\, \gamma]\!][\leftarrow [\![t \,|\, \sigma' \,|\, \gamma]\!]] \overset{\text{I.H.}}{=} [\![u \,|\, \sigma \,|\, \gamma]\!]\{M/x\}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]\{M/x\}]$
$= [\![u \,|\, \sigma \,|\, \gamma]\!][\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]]\{M/x\} = [\![u[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\{M/x\}$

Inductive Case: Distributor
$[\![u[\,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}$

SubCase: $\vec{x} = c$
$[\![u[\,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma]\!]_{\mathcal{W}}\{M/x\}$
$= [\![u[\,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\{M/x\}$
where
$\sigma''' = \sigma[c \mapsto \bullet]$
$\sigma'' = \sigma'[c \mapsto \bullet]$

SubCase $\vec{x} = x_1, \ldots, x_n$
$[\![u[\,|\, c\langle x_1, \ldots, x_n \rangle \overline{[\Gamma]}] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \,|\, \sigma''' \,|\, \gamma]\!]_{\mathcal{W}}\{M/x\}$
$= [\![u[\,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\{M/x\}$

where
$\sigma' = \sigma_1\{M/x\}[x_1 \mapsto M_1\{M/x\}, \dots, x_n \mapsto M_n\{M/x\}][x \mapsto M]$
$\sigma'' = \sigma_1\{M/x\}[x_1 \mapsto M_1\{M/x\}\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{M/x\}\{\bullet/\gamma(c)\}][x \mapsto M]$
$\sigma''' = \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}]$ ◀

Below we repeat Proposition 20.
For $N \in \Lambda$ and $t \in \Lambda_a^S$ the following properties hold

$$\begin{tikzcd} \Lambda_a^S \arrow[rr, "\llbracket - \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}}"] \arrow[dr, "\llbracket - \mid \sigma^{\Lambda} \mid \gamma \rrbracket"'] & & \Lambda_{\mathcal{W}} \arrow[dl, "\lfloor - \rfloor"] \\ & \Lambda & \end{tikzcd}
\qquad
\begin{tikzcd} \Lambda_a^S \arrow[rr, "\llbracket - \rrbracket^{\mathcal{W}}"] & & \Lambda_{\mathcal{W}} \\ & \Lambda \arrow[ul, "\llparenthesis - \rrparenthesis"] \arrow[ur, "\llparenthesis - \rrparenthesis^{\mathcal{W}}"'] & \end{tikzcd}
\qquad
\begin{tikzcd} & \Lambda_{\mathcal{W}} \arrow[dr, "\lfloor - \rfloor"] & \\ \Lambda \arrow[ur, "\llparenthesis - \rrparenthesis^{\mathcal{W}}"] \arrow[rr, "="] & & \Lambda \end{tikzcd}$$

$\lfloor \llbracket t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \llbracket t \mid \sigma^{\Lambda} \mid \gamma \rrbracket$ $\quad$ $\llbracket \llparenthesis N \rrparenthesis \rrbracket^{\mathcal{W}} = \llparenthesis N \rrparenthesis^{\mathcal{W}}$ $\quad$ $\lfloor \llparenthesis N \rrparenthesis^{\mathcal{W}} \rfloor = N$

where $\sigma^{\Lambda}(z) = \lfloor \sigma^{\mathcal{W}}(z) \rfloor$.

Proof. We prove $\lfloor \llbracket u \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \llbracket u \mid \sigma^{\Lambda} \mid \gamma \rrbracket$ by induction on u.

Base Case: Variable
$\lfloor \llbracket x \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lfloor \sigma^{\mathcal{W}}(x) \rfloor = \llbracket x \mid \sigma^{\Lambda} \mid \gamma \rrbracket$

Inductive Case: Application
$\lfloor \llbracket s\, t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lfloor \llbracket s \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \lfloor \llbracket t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \llbracket s \mid \sigma^{\Lambda} \mid \gamma \rrbracket \llbracket t \mid \sigma^{\Lambda} \mid \gamma \rrbracket = \llbracket s\, t \mid \sigma^{\Lambda} \mid \gamma \rrbracket$

Inductive Case: Abstraction
$\lfloor \llbracket x\langle x \rangle.t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lambda x.\lfloor \llbracket t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \lambda x.\llbracket t \mid \sigma^{\Lambda} \mid \gamma \rrbracket = \llbracket x\langle x \rangle.t \mid \sigma^{\Lambda} \mid \gamma \rrbracket$

Inductive Case: Phantom-Abstraction
$\lfloor \llbracket c\langle x_1, \dots, x_n \rangle.t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lambda c.\lfloor \llbracket t \mid \sigma_1^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \lambda c.\llbracket x \mid \sigma_1^{\Lambda} \mid \gamma \rrbracket = \llbracket c\langle x_1, \dots, x_n \rangle.t \mid \sigma^{\Lambda} \mid \gamma \rrbracket$
where
$\sigma_1^{\mathcal{W}} = \sigma[x_1 \mapsto \sigma(x_1)\{c/\gamma(c)\}, \dots, x_n \mapsto \sigma(x_n)\{c/\gamma(c)\}]$
$\sigma_1^{\Lambda} = \sigma[x_1 \mapsto \lfloor \sigma(x_1) \rfloor\{c/\gamma(c)\}, \dots, x_n \mapsto \lfloor \sigma(x_n) \rfloor\{c/\gamma(c)\}]$

Inductive Case: Weakening
$\lfloor \llbracket u[\leftarrow t] \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lfloor \llbracket u \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \llbracket u \mid \sigma^{\Lambda} \mid \gamma \rrbracket = \llbracket u[\leftarrow t] \mid \sigma^{\Lambda} \mid \gamma \rrbracket$

Inductive Case: Sharing
$\lfloor \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor = \lfloor \llbracket u \mid \sigma_1^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \llbracket u \mid \sigma_1^{\Lambda} \mid \gamma \rrbracket = \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma^{\Lambda} \mid \gamma \rrbracket$
where
$\sigma_1^{\mathcal{W}} = \sigma^{\mathcal{W}}[x_i \mapsto \llbracket t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}}]_{1 \le i \le n}$
$\sigma_1^{\Lambda} = \sigma^{\Lambda}[x_i \mapsto \lfloor \llbracket t \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor]_{1 \le i \le n} \overset{\text{I.H.}}{=} \sigma^{\Lambda}[x_i \mapsto \llbracket t \mid \sigma^{\Lambda} \mid \gamma \rrbracket]_{1 \le i \le n}$

Inductive Case: Distributor
$\lfloor \llbracket u[e_1\langle \vec{w}_1 \rangle, \dots, e_m\langle \vec{w}_m \rangle \mid c\langle \vec{x} \rangle \overline{[\Gamma]}] \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor$

SubCase: $\vec{x} = c$
$\lfloor \llbracket u[e_1\langle \vec{w}_1 \rangle, \dots, e_m\langle \vec{w}_m \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma^{\mathcal{W}} \mid \gamma \rrbracket_{\mathcal{W}} \rfloor$
$= \lfloor \llbracket u\overline{[\Gamma]} \mid \sigma \mid \gamma' \rrbracket_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} \llbracket u\overline{[\Gamma]} \mid \sigma^{\Lambda} \mid \gamma' \rrbracket$

$= [\![u[e_1\langle \vec{w_1} \rangle, \dots, e_m\langle \vec{w_m} \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma^{\Lambda} \mid \gamma]\!]$

SubCase: $\vec{x} = x_1, \dots, x_n$

$\lfloor [\![u[e_1\langle \vec{w_1} \rangle, \dots, e_m\langle \vec{w_m} \rangle \mid c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \mid \sigma^{\mathcal{W}} \mid \gamma]\!]_{\mathcal{W}} \rfloor$

$\lfloor [\![u\overline{[\Gamma]} \mid \sigma_1^{\mathcal{W}} \mid \gamma']\!]_{\mathcal{W}} \rfloor \overset{\text{I.H.}}{=} [\![u\overline{[\Gamma]} \mid \sigma_1^{\Lambda} \mid \gamma']\!]$

$= [\![u[e_1\langle \vec{w_1} \rangle, \dots, e_m\langle \vec{w_m} \rangle \mid c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \mid \sigma^{\Lambda} \mid \gamma]\!]$

where

$\sigma_1^{\mathcal{W}} = \sigma[x_1 \mapsto \sigma(x_1)\{c/\gamma(c)\}, \dots, x_n \mapsto \sigma(x_n)\{c/\gamma(c)\}]$

$\sigma_1^{\Lambda} = \sigma[x_1 \mapsto \lfloor \sigma(x_1) \rfloor\{c/\gamma(c)\}, \dots, x_n \mapsto \lfloor \sigma(x_n) \rfloor\{c/\gamma(c)\}]$

We prove $[\![(\!| N |\!)]\!]^{\mathcal{W}} = (\!| N |\!)^{\mathcal{W}}$ by induction on N. We prove this statement by first proving it for closed terms.

Base Case: Variable

$[\![(\!| x |\!)']\!]^{\mathcal{W}} = [\![x]\!]^{\mathcal{W}} = x = (\!| x |\!)^{\mathcal{W}}$

Inductive Case: Application

$[\![(\!| M\,N |\!)']\!]^{\mathcal{W}} = [\![(\!| M |\!)']\!]^{\mathcal{W}} [\![(\!| N |\!)']\!]^{\mathcal{W}} \overset{\text{I.H.}}{=} (\!| M |\!)^{\mathcal{W}} (\!| N |\!)^{\mathcal{W}} = (\!| M\,N |\!)^{\mathcal{W}}$

Inductive Case: Abstraction

$[\![(\!| \lambda x.M |\!)']\!]^{\mathcal{W}}$

SubCase: $|M|_x = 0$

$= \lambda x.[\![(\!| M |\!)'[\leftarrow x]]\!]^{\mathcal{W}} = \lambda x.[\![(\!| M |\!)']\!]^{\mathcal{W}}[\leftarrow x] \overset{\text{I.H.}}{=} \lambda x.(\!| M |\!)^{\mathcal{W}}[\leftarrow x] = (\!| \lambda x.M |\!)^{\mathcal{W}}$

SubCase: $|M|_x = 1$

$= \lambda x.[\![(\!| M |\!)']\!]^{\mathcal{W}} \overset{\text{I.H.}}{=} \lambda x.(\!| M |\!)^{\mathcal{W}} = (\!| \lambda x.M |\!)^{\mathcal{W}}$

SubCase: $|M|_x = n > 1$

$= [\![(\!| M\frac{n}{x} |\!)'[x^1, \dots, x^n \leftarrow x]]\!]^{\mathcal{W}} = [\![(\!| M\frac{n}{x} |\!)' \mid \sigma \mid I]\!]_{\mathcal{W}} \overset{\text{prop 44}}{=} [\![(\!| M\frac{n}{x} |\!)']\!]^{\mathcal{W}}\{x/x_i\}_{1 \le i \le n}$

$\overset{\text{I.H.}}{=} (\!| M\frac{n}{x} |\!)^{\mathcal{W}}\{x/x_i\}_{1 \le i \le n} = (\!| M |\!)^{\mathcal{W}}$

Now that we have proven is works for closed terms, we can show the statement $[\![(\!| N |\!)]\!]^{\mathcal{W}} = (\!| N |\!)^{\mathcal{W}}$ holds

$[\![(\!| N |\!)]\!]^{\mathcal{W}} = [\![(\!| N\frac{n_1}{x_1} \dots \frac{n_k}{x_k} |\!)'[x_1^1, \dots, x_1^{n_1} \leftarrow x_1] \dots [x_k^1, \dots, x_k^{n_k} \leftarrow x_k]]\!]^{\mathcal{W}}$

$\overset{\text{prop 44}}{=} [\![(\!| N\frac{n_1}{x_1} \dots \frac{n_k}{x_k} |\!)']\!]^{\mathcal{W}}\{x_i/x_i^j\}_{1 \le i \le k, 1 \le j \le n_i} = (\!| N\frac{n_1}{x_1} \dots \frac{n_k}{x_k} |\!)^{\mathcal{W}}\{x_i/x_i^j\}_{1 \le i \le k, 1 \le j \le n_i} = (\!| N |\!)^{\mathcal{W}}$ ◀

We also discuss the proofs for Lemma 24 and Lemma 25. These are: Given $t \rightsquigarrow_{\beta} u$ then

$$[\![t]\!]^{\mathcal{W}} \to_{\beta}^{+} [\![u]\!]^{\mathcal{W}}$$

and given $t \rightsquigarrow_{(C,D,L)} u$ and for any $x \in (t)_{bv} \cup (t)_{fp}$ and for all z, $x \notin (\sigma(z))_{fv}$.

$$[\![t \mid \sigma \mid \gamma]\!]_{\mathcal{W}} \to_{\mathcal{W}}^{*} [\![u \mid \sigma \mid \gamma]\!]_{\mathcal{W}}$$

Proof. We prove this by induction. We first discuss all the case bases. $[\![(x\langle x \rangle.t)\, s]\!]^{\mathcal{W}} = (\lambda x.T)\, S = T\{S/x\} = [\![t\{s/x\}]\!]^{\mathcal{W}}$
where $T = [\![t]\!]^{\mathcal{W}}$ and $S = [\![s]\!]^{\mathcal{W}}$.

We prove this is true case-by-case, which is an extension of the proof for Lemma 16. Therefore, we only show the interesting cases.

Case: (d_1)

$$u[\leftarrow s\,t] \rightsquigarrow_R u[\leftarrow s][\leftarrow t]$$

$[\![u[\leftarrow s\,t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![s \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\, [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$\rightarrow_{\mathcal{W}} [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![s \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}][\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}] = [\![u[\leftarrow s][\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case: (d_2)

$$u[\leftarrow c\langle\, \vec{x} \,\rangle.t] \rightsquigarrow_R u[\,|\, c\langle\, \vec{x} \,\rangle\, [\leftarrow t]]$$

$[\![u[\leftarrow c\langle\, \vec{x} \,\rangle.t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

SubCase: $\vec{x} = c$

$[\![u[\leftarrow c\langle\, c \,\rangle.t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow \lambda c.[\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}] \rightarrow_{\mathcal{W}} [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\{\bullet/c\}]$
$\overset{\text{prop 44}}{=} [\![u[\leftarrow t] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, c\langle\, c \,\rangle\, [\leftarrow t]] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$
where $\sigma' = \sigma[c \mapsto \bullet]$

SubCase: $\vec{x} = x_1, \ldots, x_n$

$[\![u[\leftarrow c\langle\, x_1, \ldots, x_n \,\rangle.t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![c\langle\, x_1, \ldots, x_n \,\rangle.t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$[\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow \lambda c.[\![t \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}] \rightarrow_{\mathcal{W}} [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}\{\bullet/c\}]$
$\overset{\text{prop 44}}{=} [\![u[\leftarrow t] \,|\, \sigma'' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\,|\, c\langle\, x_1, \ldots, x_n \,\rangle\, [\leftarrow t]] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case: (d_3)

$$u[\,|\, c\langle\, c \,\rangle\, [\leftarrow c]] \rightsquigarrow_R u$$

$[\![u[\,|\, c\langle\, c \,\rangle\, [\leftarrow c]] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u[\leftarrow c] \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow \bullet]$
$= [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow \bullet] \rightarrow_{\mathcal{W}} [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case (c_2)

$$u[x \leftarrow t] \rightsquigarrow_C u\{t/x\}$$

$[\![u[x \leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}} = [\![u\{t/x\} \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$
where
$\sigma' = \sigma[x \mapsto [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$

For the remaining cases, we only show the cases for $[\![u[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$. The other cases are similar to those in the proof for Lemma 16.

Case: (l_1)

$$s[\leftarrow t]\, u \rightsquigarrow_L (s\, u)[\leftarrow t]$$

$[\![s[\leftarrow t]\, u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = [\![s \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]\, [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} \rightarrow_{\mathcal{W}} ([\![s \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}\, [\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}})[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$[\![(s\, u)[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

The proofs for lifting past application (right) (l_2) and sharing (l_4) follow a similar argument so we choose to omit these cases

Case: (l_3)

$$d\langle\, \vec{x} \,\rangle.u[\leftarrow t] \rightsquigarrow_L (d\langle\, \vec{x} \,\rangle.u)[\leftarrow t] \text{ iff } \vec{x} \notin (t)_{fv}$$

SubCase: $\vec{x} = d$

$[\![d\langle d \rangle.u[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = \lambda d.([\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]) \to_{\mathcal{W}} \lambda d.[\![u \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$= [\![(d\langle \vec{x} \rangle.u)[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

SubCase: $\vec{x} = x_1, \dots, x_n$

$[\![d\langle x_1, \dots, x_n \rangle.u[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}} = \lambda d.([\![u \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}])$
$\to_{\mathcal{W}} \lambda d.[\![u \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma]\!]_{\mathcal{W}}] = [\![(d\langle x_1, \dots, x_n \rangle.u)[\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case: (l_5)

$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}[\leftarrow t]] \rightsquigarrow_L$$
$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}][\leftarrow t]$$

iff all $\vec{x} \notin (t)_{fv}$

$[\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle \vec{x} \rangle \overline{[\Gamma]}[\leftarrow t]] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case: $\vec{x} = c$

$= [\![u\overline{[\Gamma][\leftarrow t]} \,|\, \sigma \,|\, \gamma']\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \,|\, \sigma \,|\, \gamma']\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma']\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma']\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle c \rangle \overline{[\Gamma]}][\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$

Case: $\vec{x} = x_1, \dots, x_n$

$= [\![u\overline{[\Gamma][\leftarrow t]} \,|\, \sigma' \,|\, \gamma']\!]_{\mathcal{W}} = [\![u\overline{[\Gamma]} \,|\, \sigma' \,|\, \gamma']\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma']\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma' \,|\, \gamma']\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}[\leftarrow [\![t \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}]$
$= [\![u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \,|\, c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}][\leftarrow t] \,|\, \sigma \,|\, \gamma]\!]_{\mathcal{W}}$ ◀

B.1 Sharing Measure

We prove strong normalisation of sharing reductions through the use of *multisets*. Intuitively, a multiset can be interpreted as a set where elements can be repeated, or equivalently as lists that are considered equal up to the permutation of elements. We use multisets to measure aspects of a term, and show that these aspects strictly decrease via $\rightsquigarrow_{(R,D,L)}$ reduction.

▶ **Definition 45** (Multisets). *A* multiset m *is a pair* (A, f) *where* A *is a set and* $f : A \to \mathcal{N}$ *is a function that maps elements of* A *to a natural number.*

The formal definition of multisets in Definition 45 follows intuition when we consider the function f to tell us the number of occurrences of an element $x \in A$ in the multiset m.

▶ **Example 46.** Let $m = (\{x, y, z\}, f)$ and $f(x) = 2$, $f(y) = 1$ and $f(z) = 3$. Then this multiset can also be written as $\{x, x, y, z, z, z\}$ or equivalently as $\{x^2, y^1, z^3\}$

▶ Remark 47. The empty multiset is written as $\{\}$

We will need to be able to reason about multisets in order to use them as part of our reasoning for strong normalisation. First we discuss the union of multisets, which will be needed when measuring a term recursively, e.g. in an application $s\,t$ we will need to measure aspects of s and unionise them with the multiset corresponding to the measure of the same of t, to obtain the overall measure of the application.

▶ **Definition 48** (Union of Multisets). *The* union *(or* sum*) of two multisets* $m = (A, f)$ *and* $n = (B, g)$ *is the multiset* $m \uplus n = (A \cup B, h)$ *such that for all* $x \in A \cup B$, $h(x) = f(x) + g(x)$.

▶ **Example 49.** Let $m = \{a^1, b^3, c^2\}$ and $n = \{c^3, d^1\}$, then $m \uplus n = \{a^1, b^3, c^5, d^1\}$

▶ Remark 50. The notion $A \cup B$ is the union of the sets and *not* a disjoint union.

To show strong normalisation of sharing reductions, we need to show that aspects of terms that can be represented as multisets strictly decrease during reduction. In order to show this, we need to be able determine when a multiset is larger/smaller than another i.e. we need to be able to apply an ordering.

▶ **Definition 51** (Ordering of Multisets)**.** *Given a totally ordered set A and two multisets $m = (A, f)$ and $n = (A, g)$, we say m is strictly larger than n, $m > n$, if the following conditions hold*

- $m \neq n$
- $\forall x \in A.(g(x) > f(x) \rightarrow \exists y \in A.[(y > x) \wedge (f(y) > g(y))])$

▶ **Example 52.** $\{1^5, 2^2, 3^1\} < \{1^3, 2^4, 3^3\}$

The *height* of a term is intuitively a multiset of integers that record the scope of each sharing. The scope is measured by the number of constructors from the sharing node to the root of the term in its graphical notation. The formal definition of the height is given in Definition 32. First we prove Lemma 27 on a case-by-case basis.

If $t \leadsto_{(L)} u$ then $\mathcal{H}^i(t) > \mathcal{H}^i(u)$

Proof.

$$s[\Gamma]\,t \leadsto_L (s\,t)[\Gamma]$$

$\mathcal{H}^i((s[\Gamma])\,t) = \mathcal{H}^{i+1}(s[\Gamma]) \uplus \mathcal{H}^{i+1}(t) = \mathcal{H}^{i+1}(s) \uplus \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^{i+1}([\Gamma]) \uplus \{i+1\}$
$\mathcal{H}^i((s\,t)[\Gamma]) = \mathcal{H}^i(s\,t) \uplus \mathcal{H}^i([\Gamma]) = \mathcal{H}^{i+1}(s) \uplus \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^i([\Gamma]) \uplus \{i\}$

$$s\,t[\Gamma] \leadsto_L (s\,t)[\Gamma]$$

This case is similar to the one above and we omit it.

$$d\langle\,\vec{x}\,\rangle.t[\Gamma] \leadsto_L (d\langle\,\vec{x}\,\rangle.t)[\Gamma] \text{ iff all } \vec{x} \in (t)_{fv}$$

$\mathcal{H}^i(c\langle\,\vec{x}\,\rangle.t[\Gamma]) = \mathcal{H}^{i+1}(t[\Gamma]) = \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^{i+1}([\Gamma]) \uplus \{i+1\}$
$\mathcal{H}^i((c\langle\,\vec{x}\,\rangle.t)[\Gamma]) = \mathcal{H}^i(c\langle\,\vec{x}\,\rangle.t) \uplus \mathcal{H}^i([\Gamma]) \uplus \{i\} = \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^i([\Gamma]) \uplus \{i\}$

$$u[\vec{x} \leftarrow t[\Gamma]] \leadsto_L u[\vec{x} \leftarrow t][\Gamma]$$

$\mathcal{H}^i(u[\vec{x} \leftarrow t[\Gamma]]) = \mathcal{H}^i(u) \uplus \mathcal{H}^i([\vec{x} \leftarrow t[\Gamma]]) \uplus \{i\} = \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^{i+1}([\Gamma]) \uplus \{i, i+1\}$
$\mathcal{H}^i(u[\vec{x} \leftarrow t][\Gamma]) = \mathcal{H}^i(u[\vec{x} \leftarrow t]) \uplus \mathcal{H}^i([\Gamma]) \uplus \{i\} = \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(t) \uplus \mathcal{H}^{i+1}([\Gamma]) \uplus \{i, i\}$

$$u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle \,|\, c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}[\vec{y} \leftarrow t]] \leadsto_L$$

$$u\{(\vec{w}_1/\vec{y})/e_1\}_b \dots \{(\vec{w}_n/\vec{y})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{y}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{y}\,\rangle \,|\, c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}][\vec{y} \leftarrow t]$$

iff all $\vec{x} \notin (t)_{fv}$
$\mathcal{H}^i(u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle \,|\, c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}[\vec{y} \leftarrow t]])$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^i([e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle \,|\, c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}[\vec{y} \leftarrow t]]) \uplus \{i\}$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}([\vec{y} \leftarrow t]) \uplus \{i, (i+1)^{n+1}\}$
where n is the number of closures in the environment $\overline{[\Gamma]}$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+2}(t) \uplus \{i, (i+1)^{n+1}\}$
$\mathcal{H}^i(u\{(\vec{w}_1/\vec{y})/e_1\}_b \dots \{(\vec{w}_n/\vec{y})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{y}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{y}\,\rangle \,|\, c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}][\vec{y} \leftarrow t])$

$= \mathcal{H}^i(u\{(\vec{w}_1/\vec{y})/e_1\}_b \dots \{(\vec{w}_n/\vec{y})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{y}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{y}\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}]) \uplus \mathcal{H}^{i+1}(t) \uplus \{i\}$
$= \mathcal{H}^i(u\{(\vec{w}_1/\vec{y})/e_1\}_b \dots \{(\vec{w}_n/\vec{y})/e_n\}_b) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}(t) \uplus \{i^2, (i+1)^n\}$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}(t) \uplus \{i^2, (i+1)^n\}$

$$u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}][\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,d\langle\,\vec{a}\,\rangle\,\overline{[\Gamma']}]] \rightsquigarrow_L$$
$$u\{(\vec{w}_1/\vec{z})/e_1\}_b \dots \{(\vec{w}_n/\vec{z})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{z}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{z}\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}][\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,d\langle\,\vec{a}\,\rangle\,\overline{[\Gamma']}]$$

iff all $\vec{x} \in (u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}])_{fv}$
$\mathcal{H}^i(u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,d\langle\,\vec{a}\,\rangle\,\overline{[\Gamma']}]])$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}([\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,d\langle\,\vec{a}\,\rangle\,\overline{[\Gamma']}]) \uplus \{i, (i+1)^{n+1}\}$
where n is the number of closures in $\overline{[\Gamma]}$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+2}(\overline{[\Gamma']}) \uplus \{i, (i+1)^{n+1}, (i+2)^m\}$
where m is the number of closures in $\overline{[\Gamma']}$
$\mathcal{H}^i(u\{(\vec{w}_1/\vec{z})/e_1\}_b \dots \{(\vec{w}_n/\vec{z})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{z}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{z}\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}][\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,d\langle\,\vec{a}\,\rangle\,\overline{[\Gamma']}])$
$\mathcal{H}^i(u\{(\vec{w}_1/\vec{z})/e_1\}_b \dots \{(\vec{w}_n/\vec{z})/e_n\}_b[e_1\langle\,\vec{w}_1/\vec{z}\,\rangle \dots e_n\langle\,\vec{w}_n/\vec{z}\,\rangle\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}])$
$\quad \uplus \mathcal{H}^{i+1}(\overline{[\Gamma']}) \uplus \{i, (i+1)^m\}$
$= \mathcal{H}^i(u\{(\vec{w}_1/\vec{z})/e_1\}_b \dots \{(\vec{w}_n/\vec{z})/e_n\}_b) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma']}) \uplus \{i, (i+1)^{n+m}\}$
$= \mathcal{H}^i(u) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma]}) \uplus \mathcal{H}^{i+1}(\overline{[\Gamma']}) \uplus \{i, (i+1)^{n+m}\}$ ◀

The *weight* of a term is intuitively the number or copies each constructor (abstraction, application and variable) will exist after duplication. Figure 4 illustrates this, by showing a side-by-side comparison of the term

$$x\langle\,x\,\rangle.c_1\langle\,w_1\,\rangle.w_1\,((c_2\langle\,w_2\,\rangle.w_2)\,x)$$

$$[c_1\langle\,w_1\,\rangle c_2\langle\,w_2\,\rangle\,|\,y\langle\,y\,\rangle\,[w_1, w_2 \leftarrow z\langle\,z\,\rangle.z_1\,(z_2\,y)[z_1, z_2 \leftarrow z]]]$$

and its equivalent in the $\Lambda_{\mathcal{W}}$-calculus obtained by $[\![-]\!]^{\mathcal{W}}$. Each red line shows the connection between the abstraction and application constructors in both calculi. The weight of a constructor is then the number of red lines associated with it, e.g. the weight of the example is the multiset $\{1^6, 2^4, 4^1\}$.

▶ **Proposition 53.** *For $e \notin \vec{w}$, $\mathcal{W}^i(t) = \mathcal{W}^i(t\{\vec{w}/e\}_b)$*

Proof. To prove this, first we need to prove that book-keeping does not affect the function $\mathcal{V}^i(t)$. We prove this by induction on t.
Base Case: Variable
Vacuously True

Base Case: Abstraction
$\mathcal{V}^i(e\langle\,\vec{y}\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{V}^i(e\langle\,\vec{w}\,\rangle.t) = \mathcal{V}^i(t) \uplus \{e \mapsto i\} = \mathcal{V}^i(e\langle\,\vec{y}\,\rangle.t)$

Base Case: Distributor
$\mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,e\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,e\langle\,\vec{w}\,\rangle\,\overline{[\Gamma]}])$
$= \mathcal{V}^i(u\overline{[\Gamma]})\ \{\vec{e}\} = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,e\langle\,\vec{y}\,\rangle\,\overline{[\Gamma]}])$

Inductive Case: Application
$\mathcal{V}^i(s\,t\{\vec{w}/e\}_b) = \mathcal{V}^i((s\{\vec{w}/e\}_b)\,t\{\vec{w}/e\}_b) = \mathcal{V}^i(s\{\vec{w}/e\}_b) \uplus \mathcal{V}^i(t\{\vec{w}/e\}_b) \overset{\text{I.H.}^2}{=} \mathcal{V}^i(s) \uplus \mathcal{V}^i(t) = \mathcal{V}^i(s\,t)$

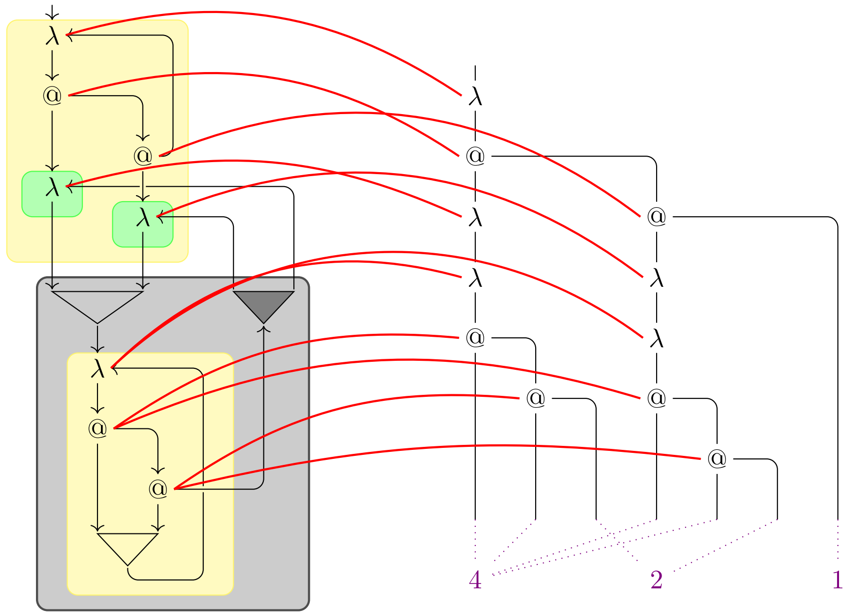


Figure 4 The weight is the multiset of incoming red arcs for each application and abstraction; here $\{1^5, 2^3\}$, together with the number of purple dotted lines for each variable; here $\{1, 2, 4\}$. Thus the overall weight is $\{1^6, 2^4, 4\}$

Inductive Case: Abstraction

Case 1

$\mathcal{V}^i((c\langle\, c\,\rangle.t)\{\vec{w}/e\}_b) = \mathcal{V}^i(c\langle\, c\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{V}^i(t\{\vec{w}/e\}_b)/\{c\} \overset{\text{I.H.}}{=} \mathcal{V}^i(t)/\{c\} = \mathcal{V}^i(c\langle\, c\,\rangle.t)$

Case 2

$\mathcal{V}^i((c\langle\,\vec{x}\,\rangle.t)\{\vec{w}/e\}_b) = \mathcal{V}^i(c\langle\,\vec{x}\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{V}^i(t\{\vec{w}/e\}_b) \uplus \{c \mapsto i\} \overset{\text{I.H.}}{=} \mathcal{V}^i(t) \uplus \{c \mapsto i\} = \mathcal{V}^i(c\langle\,\vec{x}\,\rangle.t)$

Inductive Case: Weakening

$\mathcal{V}^i(u[\leftarrow t]\{\vec{w}/e\}_b) = \mathcal{V}^i(u\{\vec{w}/e\}_b[\leftarrow t\{\vec{w}/e\}_b]) = \mathcal{V}^i(u\{\vec{w}/e\}_b) \uplus \mathcal{V}^1(t\{\vec{w}/e\}_b)$
$\overset{\text{I.H.}^2}{=} \mathcal{V}^i(u) \uplus \mathcal{V}^1(t) = \mathcal{V}^i(u[\leftarrow t])$

Inductive Case: Sharing

$\mathcal{V}^i(u[x_1 \dots x_n \leftarrow t]\{\vec{w}/e\}_b) = \mathcal{V}^i(u\{\vec{w}/e\}_b[x_1 \dots x_n \leftarrow t\{\vec{w}/e\}_b])$
$= (\mathcal{V}^i(u\{\vec{w}/e\}_b)/\{x_1, \dots, x_n\}) \uplus \mathcal{V}(tj\{\vec{w}/e\}_b)$ where $j = \mathcal{V}^i(t\{\vec{w}/e\}_b) + \dots + \mathcal{V}^i(t\{\vec{w}/e\}_b)$
$\overset{\text{I.H.}^{n+2}}{=} (\mathcal{V}^i(u)/\{x_1, \dots, x_n\}) \uplus \mathcal{V}(t)$ where $j = \mathcal{V}^i(t) + \dots + \mathcal{V}^i(t) = \mathcal{V}^i(u[x_1, \dots, x_n \leftarrow t])$

Inductive Case: Distributor

Case 1

$\mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\, c\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\, c\,\rangle\,\overline{[\Gamma]}\{\vec{w}/e\}_b]) = \mathcal{V}^i(u\overline{[\Gamma]}\{\vec{w}/e\}_b)/\{c, \vec{f}\}$
$\overset{\text{I.H.}}{=} \mathcal{V}^i(u\overline{[\Gamma]})/\{c, \vec{f}\} = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\, c\,\rangle\,\overline{[\Gamma]}])$

Case 2

$\mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}\{\vec{w}/e\}_b])$
$= \mathcal{V}^i(u\overline{[\Gamma]}\{\vec{w}/e\}_b)/\{\vec{f}\} \uplus \{c \mapsto i\}$
$\overset{\text{I.H.}}{=} \mathcal{V}^i(u\overline{[\Gamma]})/\{\vec{f}\} \uplus \{c \mapsto i\} = \mathcal{V}^i(u[\overrightarrow{f\langle\,\vec{z}\,\rangle}\,|\,c\langle\,\vec{x}\,\rangle\,\overline{[\Gamma]}])$

We now prove this proposition by induction on t

Base Case: Variable

$\mathcal{W}^i(x\{\vec{w}/e\}_b) = \mathcal{W}^i(x)$

Base Case: Abstraction

$\mathcal{W}^i(e\langle \vec{y}\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{W}^i(e\langle \vec{w}\,\rangle.t) = \mathcal{W}^i(t) \uplus \{i\} = \mathcal{W}^i(e\langle \vec{y}\,\rangle.t)$

Base Case: Distributor

$\mathcal{W}^i(u[\overrightarrow{e\langle \vec{z}\,\rangle}\,|\,e\langle \vec{y}\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{W}^i(u[\overrightarrow{e\langle \vec{z}\,\rangle}\,|\,e\langle \vec{w}\,\rangle\,\overline{[\Gamma]}]) = \mathcal{W}^i(u\overline{[\Gamma]})$

$= \mathcal{W}^i(u[\overrightarrow{e\langle \vec{z}\,\rangle}\,|\,e\langle \vec{y}\,\rangle\,\overline{[\Gamma]}])$

Inductive Case: Application

$\mathcal{W}^i(s\,t\{\vec{w}/e\}_b) = \mathcal{W}^i((s\{\vec{w}/e\}_b)\,t\{\vec{w}/e\}_b) = \mathcal{W}^i(s\{\vec{w}/e\}_b) \uplus \mathcal{W}^i(t\{\vec{w}/e\}_b) \uplus \{i\}$

$\overset{\text{I.H.}^2}{=} \mathcal{W}^i(s) \uplus \mathcal{W}^i(t) \uplus \{i\} = \mathcal{W}^i(s\,t)$

Inductive Case: Abstraction

Case 1

$\mathcal{W}^i((c\langle c\,\rangle.t)\{\vec{w}/e\}_b) = \mathcal{W}^i(c\langle c\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{W}^i(t\{\vec{w}/e\}_b) \uplus \{i, \mathcal{V}^i(t\{\vec{w}/e\}_b)(c)\}$

$\overset{\text{I.H.}}{=} \mathcal{W}^i(t) \uplus \{i, \mathcal{V}^i(t)(c)\} = \mathcal{W}^i(c\langle c\,\rangle.t)$

Case 2

$\mathcal{W}^i((c\langle \vec{x}\,\rangle.t)\{\vec{w}/e\}_b) = \mathcal{W}^i(c\langle \vec{x}\,\rangle.t\{\vec{w}/e\}_b) = \mathcal{W}^i(t\{\vec{w}/e\}_b) \uplus \{i\} \overset{\text{I.H.}}{=} \mathcal{W}^i(t) \uplus \{i\}$

$= \mathcal{W}^i(c\langle \vec{x}\,\rangle.t)$

Inductive Case: Weakening

$\mathcal{W}^i(u[\leftarrow t]\{\vec{w}/e\}_b) = \mathcal{W}^i(u\{\vec{w}/e\}_b[\leftarrow t\{\vec{w}/e\}_b]) = \mathcal{W}^i(u\{\vec{w}/e\}_b) \uplus \mathcal{W}^1(t\{\vec{w}/e\}_b)$

$\overset{\text{I.H.}^2}{=} \mathcal{W}^i(u) \uplus \mathcal{W}^1(t) = \mathcal{W}^i(u[\leftarrow t])$

Inductive Case: Sharing

$\mathcal{W}^i(u[x_1, \ldots, x_n \leftarrow t]\{\vec{w}/e\}_b) = \mathcal{W}^i(u\{\vec{w}/e\}_b[x_1, \ldots, x_n \leftarrow t\{\vec{w}/e\}_b])$

$= \mathcal{W}^i(u\{\vec{w}/e\}_b) \uplus \mathcal{W}^j(t\{\vec{w}/e\}_b)$ where $j = \mathcal{V}^i(u\{\vec{w}/e\}_b)(x_1) + \cdots + \mathcal{V}^i(u\{\vec{w}/e\}_b)(x_n)$

$\overset{\text{I.H.}^{n+2}}{=} \mathcal{W}^i(u) \uplus \mathcal{W}^j(t)$ where $j = \mathcal{V}^i(u)(x_1) + \cdots + \mathcal{V}^i(u)(x_1) = \mathcal{W}^i(u[x_1, \ldots, x_n \leftarrow t])$

Inductive Case: Distributor

Case 1

$\mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle c\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle c\,\rangle\,\overline{[\Gamma]}\{\vec{w}/e\}_b])$

$= \mathcal{W}^i(u\overline{[\Gamma]}\{\vec{w}/e\}_b) \uplus \{\mathcal{V}^i(u\overline{[\Gamma]}\{\vec{w}/e\}_b)(c)\} \overset{\text{I.H.}}{=} \mathcal{W}^i(u\overline{[\Gamma]}) \uplus \{\mathcal{V}^i(u\overline{[\Gamma]})(c)\}$

$= \mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle c\,\rangle\,\overline{[\Gamma]}])$

Case 2

$\mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle \vec{x}\,\rangle\,\overline{[\Gamma]}]\{\vec{w}/e\}_b) = \mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle \vec{x}\,\rangle\,\overline{[\Gamma]}\{\vec{w}/e\}_b])$

$= \mathcal{W}^i(u\overline{[\Gamma]}\{\vec{w}/e\}_b) \overset{\text{I.H.}}{=} \mathcal{W}^i(u\overline{[\Gamma]}) = \mathcal{W}^i(u[\overrightarrow{f\langle \vec{z}\,\rangle}\,|\,c\langle \vec{x}\,\rangle\,\overline{[\Gamma]}])$ ◀

We now prove Lemma 30 and Lemma 31 (respectively) on a case-by-case basis.

$$\text{If } t \rightsquigarrow_D u \text{ then } \mathcal{W}^i(t) > \mathcal{W}^i(u)$$

$$\text{If } t \rightsquigarrow_{(L,C)} u \text{ then } \mathcal{W}^i(t) = \mathcal{W}^i(u)$$

Proof. Duplication Rules

$$u^*[x_1 \dots x_n \leftarrow s\,t] \rightsquigarrow_D u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t]$$

$\mathcal{W}^i(u^*[x_1 \dots x_n \leftarrow s\,t]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(s\,t) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(s) \uplus \mathcal{W}^j(s) \uplus \{j\}$
where $j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n)$
$\mathcal{W}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t])$
$= \mathcal{W}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}[z_1 \dots z_n \leftarrow s]) \uplus \mathcal{W}^k(t)$
$= \mathcal{W}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}) \uplus \mathcal{W}^l(s) \uplus \mathcal{W}^k(t)$
where $k = \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}[z_1 \dots z_n \leftarrow s])(y_1) + \dots$
$\dots + \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}[z_1 \dots z_n \leftarrow s])(y_n)$
$= \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\})(y_1) + \dots + \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\})(y_n)$
$= \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) = j$
and where $l = \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\})(z_1) + \dots$
$\dots + \mathcal{V}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\})(z_n)$
$= \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) = j$
Therefore
$= \mathcal{W}^i(u^*\{z_1\,y_1/x_1\}\dots\{z_n\,y_n/x_n\}) \uplus \mathcal{W}^j(s) \uplus \mathcal{W}^j(t)$
$= \mathcal{W}^i(u) \uplus \mathcal{W}^j(s) \uplus \mathcal{W}^j(t) \uplus \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n)\}$

$$u[x_1, \dots, x_n \leftarrow c\langle\,\vec{y}\,\rangle.t] \rightsquigarrow_D$$
$$u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[e_1\langle\, w_1^1\,\rangle \dots e_n\langle\, w_1^n\,\rangle \,|\, c\langle\,\vec{y}\,\rangle\,[w_1^1, \dots, w_1^n \leftarrow t]]$$

Case 1:
$\mathcal{W}^i(u[x_1, \dots, x_n \leftarrow c\langle\,c\,\rangle.t]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(c\langle\,c\,\rangle.t) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus \{j, \mathcal{V}^j(t)(c)\}$
where $j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n)$
$\mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[e_1\langle\, w_1^1\,\rangle \dots e_n\langle\, w_1^n\,\rangle \,|\, c\langle\,c\,\rangle\,[w_1^1, \dots, w_1^n \leftarrow t]])$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t]) \uplus$
$\mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t])(c)$
$\mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t])(c) = \mathcal{V}^k(t)(c) = \mathcal{V}^j(t)(c)$
where $k = \mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t])(w_1^1) + \dots$
$\dots + \mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t])(w_1^n) = j$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t]) \uplus \mathcal{V}^j(t)(c)$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}) \uplus \mathcal{W}^k(t) \uplus \{\mathcal{V}^j(t)(c)\}$
$= \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n), \mathcal{V}^j(t)(c)\}$
Case: 2
$\mathcal{W}^i(u[x_1, \dots, x_n \leftarrow c\langle\,\vec{y}\,\rangle.t]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(c\langle\,\vec{y}\,\rangle.t) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus \{j\}$
where $j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n)$
$\mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[e_1\langle\, w_1^1\,\rangle \dots e_n\langle\, w_1^n\,\rangle \,|\, c\langle\,\vec{y}\,\rangle\,[w_1^1, \dots, w_1^n \leftarrow t]])$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}[w_1^1, \dots, w_1^n \leftarrow t])$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}) \uplus \mathcal{W}^k(t)$
where $k = \mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n})(w_1^1) + \dots + \mathcal{V}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n})(w_1^n) = j$
$= \mathcal{W}^i(u\{e_i\langle\, w_1^i\,\rangle.w_1^i/x_i\}_{1\le i\le n}) \uplus \mathcal{W}^j(t)$
$= \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n)\}$

$$u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle \,|\, c\langle\,c\,\rangle\,[\vec{w}_1, \dots, \vec{w}_n \leftarrow c]] \rightsquigarrow_D u\{e_1\langle\,\vec{w}_1\,\rangle\}_e \dots \{e_n\langle\,\vec{w}_n\,\rangle\}_e$$

$\mathcal{W}^i(u[e_1\langle\,\vec{w}_1\,\rangle \dots e_n\langle\,\vec{w}_n\,\rangle \,|\, c\langle\,c\,\rangle\,[\vec{w}_1, \dots, \vec{w}_n \leftarrow c]])$
$= \mathcal{W}^i(u[\vec{w}_1, \dots, \vec{w}_n \leftarrow c]) \uplus \{\mathcal{V}^i(u[\vec{w}_1, \dots, \vec{w}_n \leftarrow c](c))\}$
$= \mathcal{W}^i(u) \uplus \{\} \uplus \{j\}$

where $j = \mathcal{V}^i(u)(\vec{w_1}) + \cdots + \mathcal{V}^i(u)(\vec{w_n})$
$\mathcal{W}^i(u\{e_1\langle\,\vec{w_1}\,\rangle\}_e \ldots \{e_n\langle\,\vec{w_n}\,\rangle\}_e) = \mathcal{W}^i(u) \uplus \{\mathcal{V}^i(u)(\vec{w_1}), \ldots, \mathcal{V}^i(u)(\vec{w_n})\}$
where $\mathcal{V}^i(u)(\vec{w}) = \mathcal{V}^i(u)(w_1) + \cdots + \mathcal{V}^i(u)(w_n)$ and $\vec{w} = \{w_1, \ldots, w_n\}$

Lifting and Compound

$$u[\vec{w} \leftarrow y][\vec{x} \cdot y \leftarrow t] \rightsquigarrow_C u[\vec{x} \cdot \vec{w} \leftarrow t]$$

$\mathcal{W}^i(u[\vec{w} \leftarrow y][\vec{x} \cdot y \leftarrow t]) = \mathcal{W}^i(u[\vec{w} \leftarrow y]) \uplus \mathcal{W}^j(t)$
where $j = \mathcal{V}^i(u[\vec{w} \leftarrow y])(\vec{x}) + \mathcal{V}^i(u[\vec{w} \leftarrow y])(y) = \mathcal{V}^i(u[\vec{w} \leftarrow y])(\vec{x}) + \mathcal{V}^i(u)(\vec{w})$
$= \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) = \mathcal{W}^i(u[\vec{x} \cdot \vec{w} \leftarrow t])$

$$u[x \leftarrow t] \rightsquigarrow_C u\{t/x\}$$

$\mathcal{W}^i(u[x \leftarrow t]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t)$
where $j = \mathcal{V}^i(u)(x)$
$\mathcal{W}^i(u\{t/x\}) = \mathcal{W}^i(u) \uplus \mathcal{W}^{\mathcal{V}^i(u)(x)}(t)$

For the other lifting rules, we show that $\mathcal{V}^i(u[\Gamma])$ outputs the same integers before and after lifting for each variable bounded by $[\Gamma]$. Then we can know it produces some multiset M.

$$(s[\Gamma])\,t \rightsquigarrow_L (s\,t)[\Gamma]$$

$\mathcal{W}^i((s[\Gamma])\,t) = \mathcal{W}^i(s[\Gamma]) \uplus \mathcal{W}^i(t) = \mathcal{W}^i(s) \uplus \mathcal{W}^i(t) \uplus M_1$
$\mathcal{W}^i((s\,t)[\Gamma]) = \mathcal{W}^i(s\,t) \uplus M_2 = \mathcal{W}^i(s) \uplus \mathcal{W}^i(t) \uplus M_2$
$M_1 = M_2$ since $\mathcal{V}^i(s)(x) = \mathcal{V}^i(s\,t)(x)$ for $x \in (s)_{fv}$ and $[\Gamma]$ only binds variables in s.

$$s\,t[\Gamma] \rightsquigarrow_L (s\,t)[\Gamma]$$

This case is very similar to the one above and we omit it.

$$d\langle\,\vec{x}\,\rangle.t[\Gamma] \rightsquigarrow_L (d\langle\,\vec{x}\,\rangle.t)[\Gamma] \text{ iff all } \vec{x} \in (t)_{fv}$$

Case 1:
$\mathcal{W}^i(d\langle\,d\,\rangle.(t[\Gamma])) = \mathcal{W}^i(t[\Gamma]) \uplus \{i, \mathcal{V}^i(t[\Gamma])(d)\} = \mathcal{W}^i(t) \uplus M_1 \uplus \{i, \mathcal{V}^i(t)(d)\}$
$\mathcal{W}^i((d\langle\,d\,\rangle.t)[\Gamma]) = \mathcal{W}^i(d\langle\,d\,\rangle.t) \uplus M_2 = \mathcal{W}^i(t) \uplus M_2 \uplus \{i, \mathcal{V}^i(t)(d)\}$
$M_1 = M_2$ since $\mathcal{V}^i(t)(x) = \mathcal{W}^i(d\langle\,d\,\rangle.t)(x)$ where $x \neq d$ and d is not bound by $[\Gamma]$
Case 2:
$\mathcal{W}^i(d\langle\,\vec{x}\,\rangle.(t[\sigma])) = \mathcal{W}^i(t[\sigma]) \uplus \{i\} = \mathcal{W}^i(t) \uplus M_1 \uplus \{i\}$
$\mathcal{W}^i((d\langle\,\vec{x}\,\rangle.t)[\sigma]) = \mathcal{W}^i(d\langle\,\vec{x}\,\rangle.t) \uplus M_2 = \mathcal{W}^i(t) \uplus M_2 \uplus \{i\}$
$M_1 = M_2$ since $\mathcal{V}^i(t)(x) = \mathcal{W}^i(d\langle\,\vec{x}\,\rangle.t)(x)$ where $x \neq d$ and d is not bound by $[\Gamma]$

$$u[\vec{x} \leftarrow t[\Gamma]] \rightsquigarrow_L u[\vec{x} \leftarrow t][\Gamma]$$

Case 1:
$\mathcal{W}^i(u[\vec{x} \leftarrow t[\Gamma]]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t[\Gamma]) = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus M_1$
where $j = \mathcal{V}^i(u)(x_1) + \cdots + \mathcal{V}^i(u)(x_n)$
$\mathcal{W}^i(u[\vec{x} \leftarrow t][\Gamma]) = \mathcal{W}^i(u[\vec{x} \leftarrow t]) \uplus M_2 = \mathcal{W}^i(u) \uplus \mathcal{W}^j(t) \uplus M_2$
$M_1 = M_2$ since $\mathcal{V}^j(t)(x) = \mathcal{V}^i(u[\vec{x} \leftarrow t])(x)$ for $x \in (t)_{fv}$ and $[\Gamma]$ only binds variables in t
Case 2:
$\mathcal{W}^i(u[\leftarrow t[\Gamma]]) = \mathcal{W}^i(u) \uplus \mathcal{W}^1(t[\Gamma]) = \mathcal{W}^i(u) \uplus \mathcal{W}^1(t) \uplus M_1$
$\mathcal{W}^i(u[\leftarrow t][\Gamma]) = \mathcal{W}^i(u[\leftarrow t]) \uplus M_2 = \mathcal{W}^i(u) \uplus \mathcal{W}^1(t) \uplus M_2$

$M_1 = M_2$ since $\mathcal{V}^1(t)(x) = \mathcal{V}^i(u[\leftarrow t])(x)$ for $x \in (t)_{fv}$ and $[\Gamma]$ only binds variables in t

$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\vec{y} \leftarrow t]] \rightsquigarrow_L$$
$$u\{(\vec{w}_1/\vec{y})/e_1\}_b \dots \{(\vec{w}_n/\vec{y})/e_n\}_b[e_1\langle \vec{w}_1/\vec{y} \rangle \dots e_n\langle \vec{w}_n/\vec{y} \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}][\vec{y} \leftarrow t]$$

$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\overrightarrow{f\langle \vec{z} \rangle} | d\langle \vec{a} \rangle \overline{[\Gamma']}]] \rightsquigarrow_L$$
$$u\{(\vec{w}_1/\vec{z})/e_1\}_b \dots \{(\vec{w}_n/\vec{z})/e_n\}_b[e_1\langle \vec{w}_1/\vec{z} \rangle \dots e_n\langle \vec{w}_n/\vec{z} \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}][\overrightarrow{f\langle \vec{z} \rangle} | d\langle \vec{a} \rangle \overline{[\Gamma']}]$$

Since book-keeping operations do not affect the weight of a term (Proposition 53), we simplify these two rules into one, where u' is u with some book-keepings applied.
Note: Proposition 53 is relevant here since the book-keepings produced by this rule cannot be of the form $\{e/e\}_b$ without breaking linearity.

$$u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\Gamma]] \rightsquigarrow_L u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}][\Gamma]$$

Case 1:
$\mathcal{W}^i(u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle \overline{[\Gamma]}[\Gamma]]) = \mathcal{W}^i(u\overline{[\Gamma]}[\Gamma]) \uplus \{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma](c))\}$
$= \mathcal{W}^i(u\overline{[\Gamma]}) \uplus M_1 \uplus \{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma](c))\}$
$\mathcal{W}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}][\Gamma]) = \mathcal{W}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}]) \uplus M_2$
$= \mathcal{W}^i(u'[\Gamma]) \uplus M_2 \uplus \{\mathcal{V}^i(u[\Gamma])(c)\}$
$M_1 = M_2$ since $\mathcal{V}^i(u\overline{[\Gamma]})(x) = \mathcal{V}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}])(x)$
for $x \in (u\overline{[\Gamma]}/\{c, e_1, \dots, e_n\})_{fv}$ and the variables $c, e_1, \dots, e_n$ are not bound by $[\Gamma]$
$\{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma])(c)\} = \{\mathcal{V}^i(u\overline{[\Gamma]})(c)\}$ since $c \in (\overline{[\Gamma]})_{fv}$ and $\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma]) = \mathcal{V}^i(u\overline{[\Gamma]}) \uplus \mathcal{V}^j([\Gamma])$.
Case 2:
$\mathcal{W}^i(u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\Gamma]]) = \mathcal{W}^i(u\overline{[\Gamma]}[\Gamma]) = \mathcal{W}^i(u\overline{[\Gamma]}) \uplus M_1$
$\mathcal{W}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}][\Gamma]) = \mathcal{W}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}]) \uplus M_2$
$= \mathcal{W}^i(u'\overline{[\Gamma]}) \uplus M_2$
$M_1 = M_2$ since $\mathcal{V}^i(u\overline{[\Gamma]})(x) = \mathcal{V}^i(u'[e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}])(x)$
for $x \in (u\overline{[\Gamma]}/\{c, e_1, \dots, e_n\})_{fv}$ and the variables $c, e_1, \dots, e_n$ are not bound by $[\Gamma]$ ◀