

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:

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

$$\text{Distribution: } \frac{A \rightarrow (B \wedge C)}{(A \rightarrow B) \wedge (A \rightarrow C)}^d \quad \text{Switch: } \frac{(A \rightarrow B) \wedge C}{A \rightarrow (B \wedge C)}^s$$

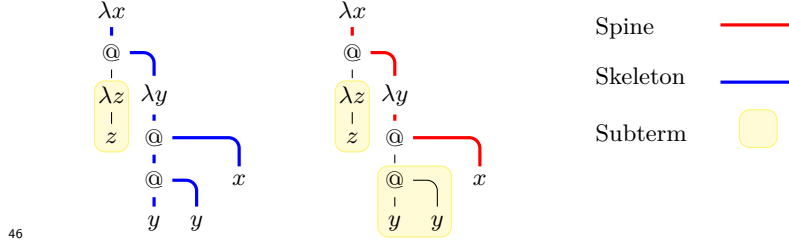
In this work, we investigate the computational interpretation of another characteristic deep-inference proof rule: the *switch* rule above right []. Our result is the *spinal atomic lambda-calculus*, a term calculus in which the natural strategy is a refined form of full laziness, *spine duplication*. This strategy duplicates only the *spine* of an abstraction: the paths to its bound variables in the syntax tree of the term.

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*.



23:2 Spinal Atomic Lambda-Calculus

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]:

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

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:

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

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, \dots, 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 identified 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.

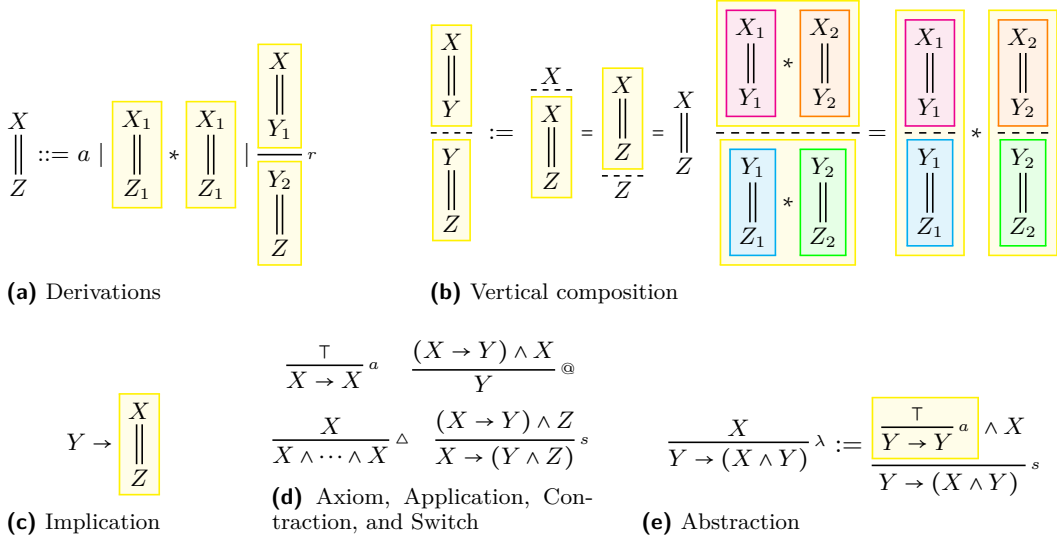
End-of-scope markers in the λ -calculus have been seen throughout literature. *Berklings' lambda bar* [5] has shown to remove the need for variable names while maintaining correctness; improving efficiency by removing the need for alpha conversion [6]. This result was generalized by *Adbmal* (invert 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.



■ **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:

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

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):

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

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 $\{t_i/x_i\}_{i \in [n]}$.

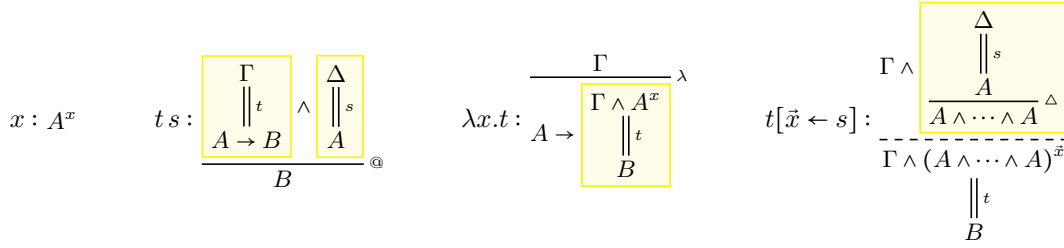
► **Definition 2.** The interpretation of a term t to the λ -term $\llbracket t \rrbracket$ given as follows

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

138 The translation $\langle N \rangle$ of a λ -term N is the unique sharing-normal term t such that $N = \llbracket t \rrbracket$.

139 A term t will be typed by a derivation with restricted types, as shown below, where the
140 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
141 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

142 3 The Spinal Atomic λ -Calculus

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

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

$$\begin{aligned} t &::= x \mid st \mid x\langle \vec{y} \rangle.t \mid t[\Gamma] \\ [\Gamma] &::= [\vec{x} \leftarrow t] \mid [e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid d\langle \vec{y} \rangle \overline{[\Gamma]}] \quad \overline{[\Gamma]} ::= [\Gamma] \mid \overline{[\Gamma]}[\Gamma] \end{aligned}$$

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

156 The distributor $u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid d\langle \vec{y} \rangle \overline{[\Gamma]}]$ captures the phantom-variables e_1, \dots, e_n
157 in u and the covers associated with those phantom-variables are captured by the environment
158 $\overline{[\Gamma]}$. We sometimes write the distributor as $u[\overrightarrow{e\langle x \rangle} \mid d\langle \vec{y} \rangle \overline{[\Gamma]}]$ when we are not concerned
159 about the binding of phantom-variables. Terms are then pre-terms with sensible and correct
160 bindings. To define terms, we first define *free* and *bound* variables and phantom variables.

161 ► **Definition 4 (Free and Bound Variables).** The free variables $(-)_f$ and bound variables
162 $(-)_b$ of a pre-term t is defined as follows

$$\begin{aligned} (x)_f &= \{x\} & (x)_b &= \{\} \\ (st)_f &= (s)_f \cup (t)_f & (st)_b &= (s)_b \cup (t)_b \end{aligned}$$

$$\begin{array}{ll}
165 & (x\langle x \rangle.t)_{fv} = (t)_{fv} - \{x\} & (x\langle x \rangle.t)_{bv} = (t)_{bv} \cup \{x\} \\
166 & (c\langle \vec{x} \rangle.t)_{fv} = (t)_{fv} & (c\langle \vec{x} \rangle.t)_{bv} = (t)_{bv} \\
167 & (u[\vec{x} \leftarrow t])_{fv} = (u)_{fv} \cup (t)_{fv} - \{\vec{x}\} & (u[\vec{x} \leftarrow t])_{bv} = (u)_{bv} \cup (t)_{bv} \cup \{\vec{x}\} \\
168 & (u[\overrightarrow{e\langle x \rangle} \mid c\langle c \rangle \overline{[\Gamma]}])_{fv} = (u[\overline{[\Gamma]}])_{fv} - \{c\} & (u[\overrightarrow{e\langle x \rangle} \mid c\langle c \rangle \overline{[\Gamma]}])_{bv} = (u[\overline{[\Gamma]}])_{bv} \\
169 & (u[\overrightarrow{e\langle x \rangle} \mid c\langle \vec{y} \rangle \overline{[\Gamma]}])_{fv} = (u[\overline{[\Gamma]}])_{fv} \cup \{c\} & (u[\overrightarrow{e\langle x \rangle} \mid c\langle \vec{y} \rangle \overline{[\Gamma]}])_{bv} = (u[\overline{[\Gamma]}])_{bv} \\
170 & & \\
171 & &
\end{array}$$

172 ► **Definition 5** (Free and Bound Phantom-Variables). *The free phantom-variables $(-)_fp$ and*
173 *bound phantom-variables $(-)_bp$ of the pre-term t is defined as follows*

$$\begin{array}{ll}
174 & (x)_{fp} = \{\} & (x)_{bp} = \{\} \\
175 & (st)_{fp} = (s)_{fp} \cup (t)_{fp} & (st)_{bp} = (s)_{bp} \cup (t)_{bp} \\
176 & (x\langle x \rangle.t)_{fp} = (t)_{fp} & (x\langle x \rangle.t)_{bp} = (t)_{bp} \\
177 & (c\langle \vec{x} \rangle.t)_{fp} = (t)_{fp} \cup \{c\} & (c\langle \vec{x} \rangle.t)_{bp} = (t)_{bp} \\
178 & (u[\vec{x} \leftarrow t])_{fp} = (u)_{fp} \cup (t)_{fp} & (u[\vec{x} \leftarrow t])_{bp} = (u)_{bp} \cup (t)_{bp} \\
180 & & \\
181 & (u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}])_{fp} = (u[\overline{[\Gamma]}])_{fp} - \{e_1, \dots, e_n\} \\
182 & (u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}])_{bp} = (u[\overline{[\Gamma]}])_{bp} \cup \{e_1, \dots, e_n\} \\
183 & (u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid c\langle \vec{y} \rangle \overline{[\Gamma]}])_{fp} = (u[\overline{[\Gamma]}])_{fp} \cup \{c\} - \{e_1, \dots, e_n\} \\
184 & (u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid c\langle \vec{y} \rangle \overline{[\Gamma]}])_{bp} = (u[\overline{[\Gamma]}])_{bp} \cup \{e_1, \dots, e_n\} \\
185 & &
\end{array}$$

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

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

- 189 1. *Each variable may occur at most once.*
- 190 2. *In an abstraction $x\langle x \rangle.t$, $x \in (t)_{fv}$.*
- 191 3. *In a phantom-abstraction $c\langle x_1, \dots, x_n \rangle.t$, $\{x_1, \dots, x_n\} \subset (t)_{fv}$.*
- 192 4. *In a sharing $u[x_1, \dots, x_n \leftarrow t]$, $\{x_1, \dots, x_n\} \subset (u)_{fv}$.*
- 193 5. *In a distributor $u[e_1\langle w_1^1, \dots, w_{k_1}^1 \rangle \dots e_n\langle w_1^n, \dots, w_{k_n}^n \rangle \mid c\langle c \rangle \overline{[\Gamma]}]$*
194 *a. For all $1 \leq i \leq n$ and $1 \leq m \leq k_n$, $w_m^i(u)_{fv}$ and becomes bound by $\overline{[\Gamma]}$.*
195 *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*
196 *becomes bound.*
197 *c. The variable c occurs somewhere in the environments $\overline{[\Gamma]}$.*
- 198 6. *In a distributor $u[e_1\langle w_1^1, \dots, w_{k_1}^1 \rangle \dots e_n\langle w_1^n, \dots, w_{k_n}^n \rangle \mid c\langle y_1, \dots, y_m \rangle \overline{[\Gamma]}]$*
199 *a. Both 5(a) and 5(b) hold.*
200 *b. For all $1 \leq i \leq m$, y_i occurs in the environments $\overline{[\Gamma]}$.*

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

$$204 \quad u[\vec{x} \leftarrow t] \sim u[\vec{x}_P \leftarrow t] \quad c\langle \vec{x} \rangle.t \sim c\langle \vec{x}_P \rangle.t$$

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

$$\frac{A \rightarrow (B_1 \wedge \dots \wedge B_n)}{(A \rightarrow B_1) \wedge \dots \wedge (A \rightarrow B_n)}^d$$

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

$$\begin{array}{c}
 \frac{(A \rightarrow \Gamma) \wedge \Delta}{A^c \rightarrow \left[\begin{array}{c} \Gamma^{\vec{x}} \wedge \Delta \\ \parallel_t \\ C \end{array} \right] s} \\
 c\langle \vec{x} \rangle.t :
 \end{array}
 \quad
 \xrightarrow{u[e\langle x \rangle \mid c\langle \vec{z} \rangle \overline{[\Gamma]}]} :
 \quad
 \frac{
 \begin{array}{c}
 \frac{(C \rightarrow \Gamma) \wedge \Delta}{C^c \rightarrow \left[\begin{array}{c} \Gamma^{\vec{z}} \wedge \Delta \\ \parallel_{[\Gamma]} \\ \Sigma_1 \wedge \dots \wedge \Sigma_n \end{array} \right] s} \\
 (C^{e_1} \rightarrow \Sigma_1^{x_1}) \wedge \dots \wedge (C^{e_n} \rightarrow \Sigma_n^{x_n}) \wedge \Omega
 \end{array}
 }{
 (C \rightarrow \Sigma_1) \wedge \dots \wedge (C \rightarrow \Sigma_n) \wedge \Omega
 }
 \begin{array}{c}
 d \\
 \parallel_u \\
 E
 \end{array}$$

■ **Figure 3** Typing derivations for phantom-abstractions and distributors

3.1 Compilation and Readback

210 We now define the translations between Λ_a^S and the original λ -calculus. First we define the
 211 interpretation $\Lambda \rightarrow \Lambda_a^S$ (*compilation*). Intuitively, it replaces each abstraction $\lambda x. -$ with
 212 the term $x\langle x \rangle. - [x_1, \dots, x_n \leftarrow x]$ where x_1, \dots, x_n replace the occurrences of x . Actual
 213 substitutions are denoted as $\{t/x\}$. Let $|M|_x$ denote the number of occurrences of x in M ,
 214 and if $|M|_x = n$ let M_x^n denote M with the occurrences of x by fresh, distinct variables
 215 x^1, \dots, x^n . First, the translation of a *closed* term M is $\llbracket M \rrbracket'$, defined below

217 ► **Definition 7** (Compilation). *The interpretation for closed lambda terms, $\llbracket \Lambda \rrbracket' : \Lambda \rightarrow \Lambda_a^S$ is*
 218 *defined below*

$$\begin{aligned}
 \llbracket x \rrbracket' &= x \\
 \llbracket M N \rrbracket' &= \llbracket M \rrbracket' \llbracket N \rrbracket' \\
 \llbracket \lambda x. M \rrbracket' &= \begin{cases} x\langle x \rangle. \llbracket M \rrbracket' & \text{if } |M|_x = 1 \\ x\langle x \rangle. \llbracket M_x^n \rrbracket' [x^1, \dots, x^n \leftarrow x] & \text{if } |M|_x = n \neq 1 \end{cases}
 \end{aligned}$$

223 For an arbitrary term M , if x_1, \dots, x_k are the free variables of M such that $|M|_{x_i} = n_i > 1$,
 224 the translation $\llbracket M \rrbracket$ is

$$\llbracket M \frac{n_1}{x_1} \dots \frac{n_k}{x_k} \rrbracket' [x_1^1, \dots, x_1^{n_1} \leftarrow x_1] \dots [x_k^1, \dots, x_k^{n_k} \leftarrow x_k]$$

225 The readback into the λ -calculus is slightly more complicated, specifically due to the bind-
 226 ings induced by the distributor. Interpreting a distributor $\llbracket u[e_1\langle \vec{x}_1 \rangle \dots e_n\langle \vec{x}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \rrbracket$
 227 construct as a λ -term requires (1) converting the phantom-abstractions it binds in u into ab-
 228 stractions (2) collapsing the environment (3) maintaining the bindings between the converted
 229 abstractions and the intended variables located in the environment.

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

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

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

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

$$\begin{aligned}
\llbracket x \mid \sigma \mid \gamma \rrbracket &= \sigma(x) \\
\llbracket st \mid \sigma \mid \gamma \rrbracket &= \llbracket s \mid \sigma \mid \gamma \rrbracket \llbracket t \mid \sigma \mid \gamma \rrbracket \\
\llbracket c \langle c \rangle . t \mid \sigma \mid \gamma \rrbracket &= \lambda c. \llbracket t \mid \sigma[c \mapsto c] \mid \gamma \rrbracket \\
\llbracket c \langle x_1, \dots, x_n \rangle . t \mid \sigma \mid \gamma \rrbracket &= \lambda c. \llbracket t \mid \sigma[x_i \mapsto \sigma(x_i)\{c/\gamma(c)\}]_{i \in [n]} \mid \gamma \rrbracket \\
\llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket &= \llbracket u \mid \sigma[x_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i \in [n]} \mid \gamma \rrbracket \\
\llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_n \langle \vec{w}_n \rangle \mid c \langle c \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket &= \llbracket u[\overline{[\Gamma]}] \mid \sigma \mid \gamma[e_i \mapsto c]_{i \in [n]} \rrbracket \\
\llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_n \langle \vec{w}_n \rangle \mid c \langle x_1, \dots, x_m \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket &= \llbracket u[\overline{[\Gamma]}] \mid \sigma' \mid \gamma[e_i \mapsto c]_{i \in [n]} \rrbracket \\
&\text{where } \sigma' = \sigma[x_i \mapsto \sigma(x_i)\{c/\gamma(c)\}]_{i \in [n]}
\end{aligned}$$

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 $N \in \Lambda$.*

$$\llbracket \langle N \rangle' \rrbracket = N \quad \llbracket \langle t \rangle \rrbracket' = t \quad \exists_{M \in \Lambda}. t = \langle M \rangle'$$

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 occurrences 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 $\llbracket u \mid \sigma \mid \gamma \rrbracket$ commutes with substitution $\{M/x\}$ in the following way*

$$\llbracket u\{t/x\} \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket] \mid \gamma \rrbracket$$

269 ► **Proposition 13.** *Book-keeping commutes with the translation in the following way*

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

$$\llbracket u\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma \mid \gamma \rrbracket$$

273 ► **Proposition 14.** *Exorcisms commute with the translation in the following way*

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

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

275 Using these operations, we define the rewrite rules that allow for spinal duplication.
 276 Firstly we have beta reduction (\rightsquigarrow_β), which strictly requires an abstraction (not a phantom).
 277

$$278 \quad (x\langle x \rangle.t)s \rightsquigarrow_\beta t\{s/x\} \quad (\beta)$$

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

$$283 \quad \frac{\frac{\Gamma}{A \rightarrow \frac{A^x \wedge \Gamma}{\parallel_t B} \wedge \frac{\Delta}{\parallel_s A}} \quad \frac{\Delta}{\parallel_s A} \wedge \Gamma}{B} \quad \rightsquigarrow_\beta \quad \frac{\frac{\Delta}{\parallel_s A} \wedge \Gamma}{A \wedge \Gamma \parallel_t B}$$

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

$$287 \quad s[\Gamma]t \rightsquigarrow_L (st)[\Gamma] \quad (l_1)$$

$$288 \quad st[\Gamma] \rightsquigarrow_L (st)[\Gamma] \quad (l_2)$$

$$289 \quad d\langle \vec{x} \rangle.t[\Gamma] \rightsquigarrow_L (d\langle \vec{x} \rangle.t)[\Gamma] \text{ if } \{\vec{x}\} \cap (t)_{fv} = \{\vec{x}\} \quad (l_3)$$

$$290 \quad u[\vec{x} \leftarrow t[\Gamma]] \rightsquigarrow_L u[\vec{x} \leftarrow t][\Gamma] \quad (l_4)$$

292 For the case of lifting a closure outside a distributor, we use a notation $\parallel [\Gamma] \parallel$ to identify the
 293 variables captured by a closure, i.e. $\parallel [\vec{x} \leftarrow t] \parallel = \{\vec{x}\}$ and $\parallel [e_1\langle \vec{x}_1 \rangle, \dots, e_n\langle \vec{x}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \parallel =$
 294 $\{\vec{x}_1, \dots, \vec{x}_n\}$. Then let $\{\vec{z}\} = \parallel [\Gamma] \parallel$ in the following rewrite rule, that can only occur if
 295 $\{\vec{x}\} \cap ([\Gamma])_{fv} = \{\}$.

$$296 \quad \begin{aligned} & u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \mid c\langle \vec{x} \rangle \overline{[\Gamma]}][\Gamma] \\ & \rightsquigarrow_L u\{(\vec{w}_i/\vec{z})/e_i\}_{b_i \in [n]}[e_1\langle \vec{w}_1/\vec{z} \rangle \dots e_n\langle \vec{w}_n/\vec{z} \rangle \mid c\langle \vec{x} \rangle \overline{[\Gamma]}][\Gamma] \end{aligned} \quad (l_5)$$

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

$$\begin{array}{ccc}
 \frac{(C \rightarrow \Gamma) \wedge \Delta \wedge \Omega}{\frac{C^c \rightarrow \frac{\Gamma^{\vec{x}} \wedge \Delta \wedge \frac{\Omega \parallel [\Gamma]}{A \wedge \dots \wedge A}}{\Sigma_1^{w_1} \dots \Sigma_n^{w_n}}}{(C \rightarrow \Sigma_1) \wedge \dots \wedge (C \rightarrow \Sigma_n)}^d}^s & \rightsquigarrow_L & \frac{(C \rightarrow \Gamma) \wedge \Delta \wedge \frac{\Omega \parallel [\Gamma]}{A \wedge \dots \wedge A}}{\frac{C^c \rightarrow \frac{\Gamma^{\vec{x}} \wedge \Delta \wedge A \dots A}{\Sigma_1^{w_1} \dots \Sigma_n^{w_n}}}{(C \rightarrow \Sigma_1) \wedge \dots \wedge (C \rightarrow \Sigma_n)}^d}^s
 \end{array}$$

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

$$\begin{array}{ccc}
 \frac{(C \rightarrow \Gamma^{\vec{x}}) \wedge \Delta \wedge A}{\frac{C^c \rightarrow \frac{\Gamma \wedge \Delta \parallel [\Gamma]}{\Sigma_1 \wedge \dots \wedge \Sigma_n} \wedge A^z}{\Sigma_1 \wedge \dots \wedge \Sigma_i \wedge A \wedge \dots \wedge \Sigma_n}}^s & \rightsquigarrow & \frac{(C \rightarrow \Gamma^{\vec{x}}) \wedge \Delta}{\frac{C^c \rightarrow \frac{\Gamma \wedge \Delta \parallel [\Gamma]}{\Sigma_1 \wedge \dots \wedge \Sigma_n} \wedge A^z}{\Sigma_1 \wedge \dots \wedge \Sigma_i \wedge A \wedge \dots \wedge \Sigma_n}}^s \wedge A^z \\
 \dots \wedge (C^{e_i} \rightarrow \Sigma_i^{\vec{w}} \wedge A) \wedge \dots & & \dots \wedge \frac{(C^{e_i} \rightarrow \Sigma_i^{\vec{w}}) \wedge A}{C \rightarrow \Sigma_i \wedge A}^s \wedge \dots
 \end{array}$$

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

$$310 \quad u[w_1, \dots, w_m \leftarrow y_i][y_1, \dots, y_n \leftarrow t] \rightsquigarrow_C u[y_1, \dots, y_{i-1}, w_1, \dots, w_m, y_{i+1}, \dots, y_n \leftarrow t] \quad (c_1)$$

$$311 \quad u[x \leftarrow t] \rightsquigarrow_C u\{t/x\} \quad (c_2)$$

$$313 \quad \frac{A}{A \wedge \frac{A}{A \wedge A}^\Delta}^\Delta \rightsquigarrow_C \frac{A}{A \wedge A \wedge A}^\Delta \quad \frac{A}{A}^\Delta \rightsquigarrow_C A$$

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

$$317 \quad u[x_1 \dots x_n \leftarrow st] \rightsquigarrow_D u\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t] \quad (d_1)$$

$$318 \quad \frac{(A \rightarrow B) \wedge A}{\frac{B}{B \wedge B}^\Delta}^\Delta \rightsquigarrow_D \frac{\frac{(A \rightarrow B)}{(A \rightarrow B) \wedge (A \rightarrow B)}^\Delta \wedge \frac{B}{B \wedge B}^\Delta}{\frac{(A \rightarrow B) \wedge A}{B}^\Delta \wedge \frac{(A \rightarrow B) \wedge A}{B}^\Delta}^\Delta$$

$$319 \quad u[x_1, \dots, x_n \leftarrow c\langle \vec{y} \rangle.t] \rightsquigarrow_D u\{e_i\langle w_i \rangle.w_i/x_i\}_{i \in [n]}[e_1\langle w_1 \rangle \dots e_n\langle w_n \rangle | c\langle \vec{y} \rangle[w_1, \dots, w_n \leftarrow t]] \quad (d_2)$$

$$\begin{array}{c}
320 \quad \frac{(A \rightarrow B) \wedge \Gamma}{A \rightarrow \boxed{\begin{array}{c} B \wedge \Gamma \\ \parallel \\ C \end{array}}}^s \quad \rightsquigarrow_D \quad \frac{(A \rightarrow B) \wedge \Gamma}{A \rightarrow \boxed{\begin{array}{c} B \wedge \Gamma \\ \parallel \\ C \\ \hline C \wedge C \end{array}}}^s \\
\frac{\phantom{A \rightarrow \boxed{\begin{array}{c} B \wedge \Gamma \\ \parallel \\ C \end{array}}}}{(A \rightarrow C) \wedge (A \rightarrow C)}^\Delta \quad \frac{\phantom{A \rightarrow \boxed{\begin{array}{c} B \wedge \Gamma \\ \parallel \\ C \\ \hline C \wedge C \end{array}}}}{(A \rightarrow C) \wedge (A \rightarrow C)}^d
\end{array}$$

$$321 \quad u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\vec{w}_1, \dots, \vec{w}_n \leftarrow c]] \rightsquigarrow_D u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e \quad (d_3)$$

$$322 \quad \frac{\overline{A}^a}{A \rightarrow \overline{A \wedge A}^\Delta}^d \rightsquigarrow_D \overline{A \rightarrow A}^a \wedge \overline{A \rightarrow A}^a$$

323 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
 324 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 $u' = u\{e_i\langle z_i \rangle.z_i/z_i\}_{i \in [2]}$.
 325 Then by reductions (d_1, l_5) , we obtain the distributor $u''[e_1\langle z_1 \rangle, e_2\langle z_2 \rangle | x\langle x \rangle [z_1, z_2 \leftarrow$
 326 $\lambda y.(y y) x]]$ 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 distrib-
 327 utor $u'''[e_1\langle z_1 \rangle, e_2\langle z_2 \rangle | x\langle x \rangle [z_1, z_2 \leftarrow x]]$ which can be eliminated by (d_3) . A full example
 328 can be found in the Appendix.

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

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

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

336 ► **Lemma 16.** *If $s \rightsquigarrow_{(L, D, C)} t$ then $\llbracket s | \sigma | \gamma \rrbracket = \llbracket t | \sigma | \gamma \rrbracket$*

337 4 Strong Normalisation of Sharing Reductions

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

345 ► **Definition 17.** *The w -terms and the weakening calculus (Λ_w) are*

$$346 \quad T, U, V ::= x \mid \lambda x.T^* \mid UV \mid T[\leftarrow U] \mid \bullet \quad (*) \text{ where } x \in (T)_{fv}$$

347 The terms are variable, abstraction, application, weakening, and a bullet. In the weakening
 348 $T[\leftarrow U]$, the subterm U is *weakened*. The interpretation of atomic terms to weakening terms
 349 $\llbracket - | - | - \rrbracket_w$ can be seen as an extension of the translation into the λ -calculus (Definition 9)

350 ► **Definition 18.** *The interpretation $\llbracket - | - | - \rrbracket_w : \Lambda_a^S \times (V \rightarrow \Lambda_w) \times (V \rightarrow V) \rightarrow \Lambda_w$ with*
 351 *maps $\sigma : V \rightarrow \Lambda_w$ and $\gamma : V \rightarrow V$ is defined as an extension of the translation in (Definition*

23:12 Spinal Atomic Lambda-Calculus

9) with the following additional special cases.

$$\llbracket u[\leftarrow t] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}[\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}]$$

$$\llbracket u[\mid c \langle c \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\overline{[\Gamma]}] \mid \sigma[c \mapsto \bullet] \mid \gamma \rrbracket_{\mathcal{W}}$$

$$\llbracket u[\mid c \langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}}$$

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

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

► **Definition 19.** The interpretation $M \in \Lambda$, $\llbracket - \rrbracket^{\mathcal{W}} : \Lambda \rightarrow \Lambda_{\mathcal{W}}$ is defined by

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

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

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

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

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

$$((\lambda x. T) \overline{[\Gamma]}) U \rightarrow_{\beta} T\{U/x\} \overline{[\Gamma]} \quad (\mathcal{W}_{\beta})$$

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 $\llbracket N \rrbracket^{\mathcal{W}}$

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 $(\rightarrow_{\mathcal{W}})$ proceeds as follows.

$$\lambda x. T[\leftarrow U] \rightarrow_{\mathcal{W}} (\lambda x. T)[\leftarrow U] \quad \text{if } x \notin (U)_{fv} \quad (\mathcal{W}_1)$$

$$U[\leftarrow T] V \rightarrow_{\mathcal{W}} (U V)[\leftarrow T] \quad (\mathcal{W}_2)$$

$$U V[\leftarrow T] \rightarrow_{\mathcal{W}} (U V)[\leftarrow T] \quad (\mathcal{W}_3)$$

$$T[\leftarrow U[\leftarrow V]] \rightarrow_{\mathcal{W}} T[\leftarrow U][\leftarrow V] \quad (\mathcal{W}_4)$$

$$T[\leftarrow \lambda x. U] \rightarrow_{\mathcal{W}} T[\leftarrow U\{\bullet/x\}] \quad (\mathcal{W}_5)$$

$$T[\leftarrow U V] \rightarrow_{\mathcal{W}} T[\leftarrow U][\leftarrow V] \quad (\mathcal{W}_6)$$

$$T[\leftarrow \bullet] \rightarrow_{\mathcal{W}} T \quad (\mathcal{W}_7)$$

$$T[\leftarrow U] \rightarrow_{\mathcal{W}} T \quad \text{if } U \text{ is a subterm of } T \quad (\mathcal{W}_8)$$

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 $t[\leftarrow y][\tilde{x} \cdot y \cdot \tilde{z} \leftarrow u] \rightsquigarrow_C t[\tilde{x} \cdot \tilde{z} \leftarrow u]$. On the left hand side, the corresponding weakening-term (obtained by $(\downarrow)^w$) would have the weakening $[\leftarrow U]$ where $U = (\downarrow u)^w$. 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 resemblance is confirmed by the following Lemmas.

► **Lemma 24.** *If $t \rightsquigarrow_{\beta} u$ then $\llbracket t \rrbracket^w \rightarrow_{\beta}^+ \llbracket u \rrbracket^w$*

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

$$\llbracket t \mid \sigma \mid \gamma \rrbracket_w \rightarrow_w^* \llbracket u \mid \sigma \mid \gamma \rrbracket_w$$

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 \cup to denote the disjoint union of two multisets. We denote $\mathcal{H}^i([\Gamma_1]) \cup \dots \cup \mathcal{H}^i([\Gamma_n])$ as $\mathcal{H}^i(\overline{[\Gamma]})$ for the environment $\overline{[\Gamma]} = [\Gamma_1], \dots, [\Gamma_n]$.

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

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

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*

$$\begin{aligned}
\mathcal{V}^i(x) &= \{x \mapsto i\} \\
\mathcal{V}^i(st) &= \mathcal{V}^i(s) \cup \mathcal{V}^i(t) \\
\mathcal{V}^i(c\langle c \rangle.t) &= \mathcal{V}^i(t) / \{c\} \\
\mathcal{V}^i(c\langle \vec{x} \rangle.t) &= \mathcal{V}^i(t) \cup \{c \mapsto i\} \\
\mathcal{V}^i(t[\leftarrow s]) &= \mathcal{V}^i(t) \cup \mathcal{V}^1(s) \\
\mathcal{V}^i(t[x_1, \dots, x_n \leftarrow s]) &= \mathcal{V}^i(t) / \{x_1, \dots, x_n\} \cup \mathcal{V}^{f(x_1) + \dots + f(x_n)}(s) \text{ where } f = \mathcal{V}^i(t) \\
\mathcal{V}^i(t[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\overline{\Gamma}]]]) &= \mathcal{V}^i(t[\overline{\Gamma}]) / \{c, e_1, \dots, e_n\} \\
\mathcal{V}^i(t[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle [\overline{\Gamma}]]]) &= \mathcal{V}^i(t[\overline{\Gamma}]) / \{e_1, \dots, e_n\} \cup \{c \mapsto i\}
\end{aligned}$$

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*

$$\begin{aligned}
\mathcal{W}^i(x) &= \{\} \\
\mathcal{W}^i(st) &= \mathcal{W}^i(s) \cup \mathcal{W}^i(t) \cup \{i\} \\
\mathcal{W}^i(c\langle c \rangle.t) &= \mathcal{W}^i(t) \cup \{i\} \cup \{\mathcal{V}^i(t)(c)\} \\
\mathcal{W}^i(c\langle \vec{x} \rangle.t) &= \mathcal{W}^i(t) \cup \{i\} \\
\mathcal{W}^i(t[\leftarrow s]) &= \mathcal{W}^i(t) \cup \mathcal{W}^1(s) \\
\mathcal{W}^i(t[x_1, \dots, x_n \leftarrow s]) &= \mathcal{W}^i(t) \cup \mathcal{W}^{f(x_1) + \dots + f(x_n)}(s) \text{ where } f = \mathcal{V}^i(t) \\
\mathcal{W}^i(t[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\overline{\Gamma}]]]) &= \mathcal{W}^i(t[\overline{\Gamma}]) \cup \{\mathcal{V}^i(t[\overline{\Gamma}](c))\} \\
\mathcal{W}^i(t[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle [\overline{\Gamma}]]]) &= \mathcal{W}^i(t[\overline{\Gamma}])
\end{aligned}$$

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 it 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*

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

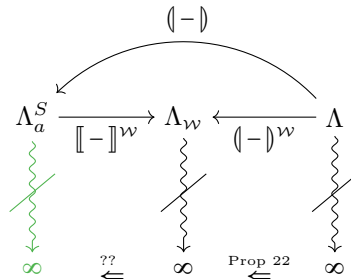
470 Now that we have proven the sharing reductions are strongly normalising, we can prove that
 471 they are confluent for closed terms.

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

473 **Proof.** Lemma 16 tells us that the preservation is preserved under reduction i.e. for $s \rightsquigarrow_{(D,L,C)}$
 474 t , $\llbracket s \rrbracket = \llbracket t \rrbracket$. Therefore given $t \rightsquigarrow_{(D,L,C)}^* s_1$ and $t \rightsquigarrow_{(D,L,C)}^* s_2$, $\llbracket t \rrbracket = \llbracket s_1 \rrbracket = \llbracket s_2 \rrbracket$. Since
 475 we know that sharing reductions are strongly normalising, we know there exists terms u_1
 476 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
 477 tells us that terms in closed terms in sharing normal form are in correspondence with their
 478 denotations i.e. $\langle \llbracket t \rrbracket \rangle' = t$. Since by Lemma 16 we know $\llbracket u_1 \rrbracket = \llbracket s_1 \rrbracket = \llbracket s_2 \rrbracket = \llbracket u_2 \rrbracket$, and
 479 by Lemma 11 $\langle \llbracket u_1 \rrbracket \rangle' = u_1$ and $\langle \llbracket u_2 \rrbracket \rangle' = u_2$, we can conclude $u_1 = u_2$. Hence, we prove
 480 confluence. ◀

481 5 Preservation of Strong Normalisation

482 Here we show how Λ_a^S preserves strong normalisation with respect to the λ -calculus. Recall
 483 that by Proposition 20 that for all $N \in \Lambda$, $\llbracket \langle N \rangle \rrbracket^w = \langle N \rangle^w$, and that Proposition 22 states if
 484 a term $N \in \Lambda$ is strongly normalising then so is $\langle N \rangle^w$. Observe that the statement ‘if term M
 485 has an infinite reduction sequence then term N has an infinite reduction sequence’ is equivalent
 486 to ‘if term N is strongly normalising then term M is strongly normalising’ by contraposition.
 487 Therefore, given a strongly normalising term $N \in \Lambda$, we know that its corresponding weakening
 488 term is also strongly normalising. Furthermore, since $\llbracket \langle N \rangle \rrbracket^w = \langle N \rangle^w$, we know that
 489 $\llbracket \langle N \rangle \rrbracket^w$ is also strongly normalising.



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

492 ▶ **Lemma 35.** *For $t \in \Lambda_a^S$ has an infinite reduction path, then $\llbracket t \rrbracket^w$ also has an infinite
 493 reduction path.*

494 **Proof.** Due to Theorem 34, we know that the infinite reduction path contains an infinite
 495 β -reduction. This means in the reduction sequence, between each β -reduction, there are
 496 finite many $\rightsquigarrow_{(D,L,C)}$ reduction steps. Lemma 25 says each $\rightsquigarrow_{(D,L,C)}$ step in Λ_a^S corresponds
 497 to zero or more weakening reductions (\rightsquigarrow_w^*). Lemma 24 says that each beta reduction in Λ_a^S
 498 corresponds to one or more β -steps in Λ_w . Therefore, it is inevitable that $\llbracket t \rrbracket^w$ also has an
 499 infinite reduction path. ◀

500 ▶ **Theorem 36.** *If $N \in \Lambda$ is strongly normalising, then so is $\langle N \rangle$.*

Proof. For a given $N \in \Lambda$ that is strongly normalising, we know by Lemma 22 that $(N)^\omega$ is strongly normalising. Then $\llbracket (N) \rrbracket^\omega$ is strongly normalising, since Proposition 20 states that $(N)^\omega = \llbracket (N) \rrbracket^\omega$. Then by Lemma 35, which states that if $\llbracket t \rrbracket^\omega$ is strongly normalising, then t is strongly normalising, proves that (N) is strongly normalising. \blacktriangleleft

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](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*

- 548 *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.
- 549
- 550 9 Kai Brännler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis
551 and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*
552 (*LPAR*), volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag,
553 2001. URL: <http://cs.bath.ac.uk/ag/kai/lcl-lpar.pdf>, doi:10.1007/3-540-45653-8\
554 _24.
- 555 10 Naim Cagman and J.Roger Hindley. Combinatory weak reduction in lambda cal-
556 culus. *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](https://doi.org/10.1016/S0304-3975(97)00250-8).
- 557
- 558
- 559 11 Maribel Fernández and Ian Mackie. Closed reductions in the λ -calculus. In Jörg Flum and
560 Mario Rodríguez-Artalejo, editors, *Computer Science Logic*, pages 220–234, Berlin, Heidelberg,
561 1999. Springer Berlin Heidelberg.
- 562 12 Maribel Fernández, Ian Mackie, and François-Régis Sinot. Closed reduction: explicit substi-
563 tutions without α -conversion. *Mathematical Structures in Computer Science*, 15(2):343–381,
564 2005. doi:10.1017/S0960129504004633.
- 565 13 Maribel Fernández, Ian Mackie, and François-Régis Sinot. Lambda-calculus with director
566 strings. *Applicable Algebra in Engineering, Communication and Computing*, 15(6):393–
567 437, Apr 2005. URL: <https://doi.org/10.1007/s00200-005-0169-9>, doi:10.1007/
568 s00200-005-0169-9.
- 569 14 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational*
570 *Logic*, 8(1):1–64, 2007. URL: <http://cs.bath.ac.uk/ag/p/SystIntStr.pdf>, doi:10.1145/
571 1182613.1182614.
- 572 15 Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic lambda calculus: A typed
573 lambda-calculus with explicit sharing. In Orna Kupferman, editor, *28th Annual IEEE*
574 *Symposium on Logic in Computer Science (LICS)*, pages 311–320. IEEE, 2013. URL:
575 <http://opus.bath.ac.uk/34527/1/AL.pdf>, doi:10.1109/LICS.2013.37.
- 576 16 Fanny He. *The Atomic Lambda-Mu Calculus*. PhD thesis, University of Bath, 2018. URL:
577 <https://fh341.github.io/pdf/HE-Thesis.pdf>.
- 578 17 Dimitri Hendriks and Vincent van Oostrom. Adbmal. In Franz Baader, editor, *Automated*
579 *Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach,*
580 *FL, USA, July 28 - August 2, 2003, Proceedings*, volume 2741 of *Lecture Notes in Computer*
581 *Science*, pages 136–150, 2003. URL: https://doi.org/10.1007/978-3-540-45085-6_11, doi:
582 10.1007/978-3-540-45085-6_11.
- 583 18 Richard Kennaway and Ronan Sleep. Director strings as combinators. *ACM Trans. Program.*
584 *Lang. Syst.*, 10(4):602–626, October 1988. URL: <http://doi.acm.org/10.1145/48022.48026>,
585 doi:10.1145/48022.48026.
- 586 19 Jan Willem Klop. *Combinatory Reduction Systems*. PhD thesis, Utrecht University, 1980.
- 587 20 Yves Lafont. From proof-nets to interaction nets. In *Advances in Linear Logic*, pages 225–247.
588 Cambridge University Press, 1994.
- 589 21 Jean-Jacques Lévy. Optimal reductions in the lambda calculus. *To HB Curry: Essays on*
590 *Combinatory Logic, Lambda Calculus and Formalism*, pages 159–191, 1980.
- 591 22 François-Régis Sinot, Maribel Fernández, and Ian Mackie. Efficient reductions with director
592 strings. In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications*, pages 46–60,
593 Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- 594 23 Alwen Tiu. A system of interaction and structure II: The need for deep inference. *Logical*
595 *Methods in Computer Science*, 2(2):4:1–24, 2006. URL: [https://arxiv.org/pdf/cs/0512036.](https://arxiv.org/pdf/cs/0512036.pdf)
596 pdf, doi:10.2168/LMCS-2(2:4)2006.
- 597 24 Vincent van Oostrom, Kees-Jan van de Looij, and Marijn Zwieterlood. Lambdascope: another
598 optimal implementation of the lambda-calculus. In *Workshop on Algebra and Logic on*
599 *Programming Systems (ALPS)*, 2004.

23:18 Spinal Atomic Lambda-Calculus

- 600 25 Christopher P. Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus*. PhD thesis,
601 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 $\llbracket s \rrbracket = \llbracket t \rrbracket$.

Proof. Let us consider the cases.

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

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

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. \blacktriangleleft

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

$$\llbracket \llbracket N \rrbracket' \rrbracket = N \quad \llbracket \llbracket t \rrbracket \rrbracket' = t \quad \exists_{M \in \Lambda}. t = \llbracket M \rrbracket'$$

Proof. We prove $\llbracket \llbracket N \rrbracket' \rrbracket = N$ by induction on N

Base Case: Variable

$$\llbracket \llbracket x \rrbracket' \rrbracket = \llbracket x \rrbracket = x$$

Inductive Case: Application

$$\llbracket \llbracket M N \rrbracket' \rrbracket = \llbracket \llbracket M \rrbracket' \rrbracket \llbracket \llbracket N \rrbracket' \rrbracket = M N$$

Inductive Case: Abstraction

$$\llbracket \llbracket \lambda x. M \rrbracket' \rrbracket$$

$$\text{Case: } |M|_x = 1$$

$$= \lambda x. \llbracket \llbracket M \rrbracket' \rrbracket = \lambda x. M$$

$$\text{Case: } |M|_x = n$$

$$= \lambda x. \llbracket \llbracket M_x^n \rrbracket' [x_1, \dots, x_n \leftarrow x] \rrbracket = \lambda x. \llbracket \llbracket M_x^n \rrbracket' \mid \sigma \mid I \rrbracket = \lambda x. \llbracket \llbracket M_x^n \rrbracket' \rrbracket \{x/x_i\}_{1 \leq i \leq n}$$

$$\stackrel{\text{I.H.}}{=} \lambda x. M_x^n \{x/x_i\}_{1 \leq i \leq n} = \lambda x. M$$

We prove $\llbracket \llbracket t \rrbracket \rrbracket' = t$ by induction on t

Base Case: Variable

$$\llbracket \llbracket x \rrbracket \rrbracket' = \llbracket x \rrbracket' = x$$

Inductive Case: Application

$$\llbracket \llbracket s t \rrbracket \rrbracket' = \llbracket \llbracket s \rrbracket \rrbracket' \llbracket \llbracket t \rrbracket \rrbracket' \stackrel{\text{I.H.}}{=} s t$$

Inductive Case: Abstraction

643 Case: $\langle \llbracket x \langle x \rangle . t \rrbracket \rangle' = x \langle x \rangle . \langle \llbracket t \rrbracket \rangle' \stackrel{\text{I.H.}}{=} x \langle x \rangle . t$

644

645 Case: $\langle \llbracket x \langle x \rangle . t[x_1, \dots, x_n \leftarrow x] \rrbracket \rangle' = \langle \lambda x. \llbracket t \mid \sigma \mid I \rrbracket \rangle'$
 646 $= \langle \lambda x. \llbracket t \rrbracket \{x/x_i\}_{1 \leq i \leq n} \rangle' = x \langle x \rangle . \langle \llbracket t \rrbracket \rangle' [x_1, \dots, x_n \leftarrow x]$
 647 $\stackrel{\text{I.H.}}{=} x \langle x \rangle . t[x_1, \dots, x_n \leftarrow x]$

648

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

650 A.2 Rewrite Rules

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

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

$$\begin{aligned}
 654 \quad & x\{s/x\} = s \\
 655 \quad & y\{s/x\} = y \\
 656 \quad & (ut)\{s/x\} = (u\{s/x\})t\{s/x\} \\
 657 \quad & (c\langle \bar{y} \rangle . t)\{s/x\} = c\langle \bar{y} \rangle . t\{s/x\} \\
 658 \quad & (c\langle \bar{y} \cdot x \rangle . t)\{s/x\} = c\langle \bar{y} \cdot \bar{z} \rangle . t\{s/x\} \\
 659 \quad & u[\bar{y} \leftarrow t]\{s/x\} = u\{s/x\}[\bar{y} \leftarrow t\{s/x\}] \\
 660 \quad & u[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \rangle \overline{[\Gamma]}]\{s/x\} = u[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \rangle \overline{[\Gamma]}]\{s/x\} \\
 661 \quad & u[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \cdot x \rangle \overline{[\Gamma]}]\{s/x\} = u[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \cdot \bar{z} \rangle \overline{[\Gamma]}]\{s/x\} \\
 662 \quad & u[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \rangle \{s/x\} \overline{[\Gamma]}] = u\{s/x\}[\overrightarrow{e\langle \bar{w} \rangle} \mid c\langle \bar{y} \rangle \overline{[\Gamma]}] \\
 663 \quad & u[e\langle e_i \langle \bar{w} \cdot x \rangle \rangle \mid c\langle \bar{y} \rangle \{s/x\} \overline{[\Gamma]}] = u\{s/x\}[e\langle e_i \langle \bar{w} \cdot \bar{z} \rangle \rangle \mid c\langle \bar{y} \rangle \overline{[\Gamma]}]
 \end{aligned}$$

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

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

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

$$\begin{aligned}
 672 \quad & x\{\bar{w}/e\}_b = x \\
 673 \quad & st\{\bar{w}/e\}_b = (s\{\bar{w}/e\}_b)t\{\bar{w}/e\}_b \\
 674 \quad & e\langle \bar{z} \rangle . t\{\bar{w}/e\}_b = e\langle \bar{w} \rangle . t \\
 675 \quad & (c\langle \bar{z} \rangle . t)\{\bar{w}/e\}_b = c\langle \bar{z} \rangle . t\{\bar{w}/e\}_b \\
 676 \quad & u[\bar{z} \leftarrow t]\{\bar{w}/e\}_b = u\{\bar{w}/e\}_b[\bar{z} \leftarrow t\{\bar{w}/e\}_b] \\
 677 \quad & u[\overrightarrow{f\langle \bar{y} \rangle} \mid e\langle \bar{z} \rangle \overline{[\Gamma]}]\{\bar{w}/e\}_b = u[\overrightarrow{f\langle \bar{y} \rangle} \mid e\langle \bar{w} \rangle \overline{[\Gamma]}] \\
 678 \quad & u[\overrightarrow{f\langle \bar{y} \rangle} \mid c\langle \bar{z} \rangle \overline{[\Gamma]}]\{\bar{w}/e\}_b = u[\overrightarrow{f\langle \bar{y} \rangle} \mid c\langle \bar{z} \rangle \overline{[\Gamma]}]\{\bar{w}/e\}_b \\
 679 \quad & u[\overrightarrow{f\langle \bar{y} \rangle} \mid c\langle \bar{z} \rangle \{ \bar{w}/e \}_b \overline{[\Gamma]}] = u\{\bar{w}/e\}_b[\overrightarrow{f\langle \bar{y} \rangle} \mid c\langle \bar{z} \rangle \overline{[\Gamma]}]
 \end{aligned}$$

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

$$682 \quad y\{c\langle \bar{x} \rangle\}_e = y$$

$$\begin{aligned}
683 \quad & st\{c\langle \tilde{x} \rangle\}_e = (s\{c\langle \tilde{x} \rangle\}_e) t\{c\langle \tilde{x} \rangle\}_e \\
684 \quad & c\langle \tilde{x} \rangle.t\{c\langle \tilde{x} \rangle\}_e = c\langle c \rangle.t[\tilde{x} \leftarrow c] \\
685 \quad & d\langle \tilde{y} \rangle.t\{c\langle \tilde{x} \rangle\}_e = d\langle \tilde{y} \rangle.t\{c\langle \tilde{x} \rangle\}_e \\
686 \quad & u[\tilde{y} \leftarrow t]\{c\langle \tilde{x} \rangle\}_e = u\{c\langle \tilde{x} \rangle\}_e[\tilde{y} \leftarrow t\{c\langle \tilde{x} \rangle\}_e] \\
687 \quad & u[\overrightarrow{e\langle \tilde{w} \rangle} \mid c\langle \tilde{x} \rangle [\overline{\Gamma}]]\{c\langle \tilde{x} \rangle\}_e = u[\overrightarrow{e\langle \tilde{w} \rangle} \mid c\langle c \rangle [\overline{\Gamma}][\tilde{x} \leftarrow c]] \\
688 \quad & u[\overrightarrow{e\langle \tilde{w} \rangle} \mid d\langle \tilde{y} \rangle [\overline{\Gamma}]]\{c\langle \tilde{x} \rangle\}_e = u[\overrightarrow{e\langle \tilde{w} \rangle} \mid d\langle \tilde{y} \rangle [\overline{\Gamma}]]\{c\langle \tilde{x} \rangle\}_e \\
689 \quad & u[\overrightarrow{e\langle \tilde{w} \rangle} \mid d\langle \tilde{y} \rangle \{c\langle \tilde{x} \rangle\}_e [\overline{\Gamma}]] = u\{c\langle \tilde{w} \rangle\}_e[\overrightarrow{e\langle \tilde{w} \rangle} \mid d\langle \tilde{y} \rangle [\overline{\Gamma}]]
\end{aligned}$$

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

► **Example 40.** Take the λ -term $M = (\lambda f.\lambda x.f(fx)) \lambda g.\lambda y.g(gy)$.

Then $\llbracket M \rrbracket = (f\langle f \rangle.x\langle x \rangle.f_1(f_2x)[f_1, f_2 \leftarrow f])(g\langle g \rangle.y\langle y \rangle.g_1(g_2y)[g_1, g_2 \leftarrow g])$.

We then may have the following reduction sequence.

$$\begin{aligned}
695 \quad & (f\langle f \rangle.x\langle x \rangle.f_1(f_2x)[f_1, f_2 \leftarrow f])(g\langle g \rangle.y\langle y \rangle.g_1(g_2y)[g_1, g_2 \leftarrow g]) \\
696 \quad & \rightsquigarrow_{\beta} x\langle x \rangle.f_1(f_2x)[f_1, f_2 \leftarrow g\langle g \rangle.y\langle y \rangle.g_1(g_2y)[g_1, g_2 \leftarrow g]] \quad (\beta) \\
697 \quad & \rightsquigarrow_D x\langle x \rangle.(f_1\langle w_1 \rangle.w_1((f_2\langle w_2 \rangle.w_2)x)) \\
698 \quad & \quad [f_1\langle w_1 \rangle, f_2\langle w_2 \rangle \mid g\langle g \rangle[w_1, w_2 \leftarrow y\langle y \rangle.g_1(g_2y)[g_1, g_2 \leftarrow g]]] \quad (d_2) \\
699 \quad & \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)) \\
700 \quad & \quad [f_1\langle z_1 \rangle, f_2\langle z_2 \rangle \mid g\langle g \rangle[y_1\langle z_1 \rangle, y_2\langle z_2 \rangle \mid y\langle y \rangle \\
701 \quad & \quad [z_1, z_2 \leftarrow g_1(g_2y)[g_1, g_2 \leftarrow g]]] \quad (d_2) \\
702 \quad & \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)) \\
703 \quad & \quad [f_1\langle z_1 \rangle, f_2\langle z_2 \rangle \mid g\langle g \rangle[y_1\langle z_1 \rangle, y_2\langle z_2 \rangle \mid y\langle y \rangle \\
704 \quad & \quad [z_1, z_2 \leftarrow g_1(g_2y)][g_1, g_2 \leftarrow g]]] \quad (l_4) \\
705 \quad & \rightsquigarrow_D x\langle x \rangle.((f_1\langle a_1, b_1 \rangle.y_1\langle a_1, b_1 \rangle.a_1b_1)((f_2\langle a_2, b_2 \rangle.y_2\langle a_2, b_2 \rangle.a_2b_2)x)) \\
706 \quad & \quad [f_1\langle a_1, b_1 \rangle, f_2\langle a_2, b_2 \rangle \mid g\langle g \rangle[y_1\langle a_1, b_1 \rangle, y_2\langle a_2, b_2 \rangle \mid y\langle y \rangle \\
707 \quad & \quad [a_1, a_2 \leftarrow g_1][b_1, b_2 \leftarrow g_2y][g_1, g_2 \leftarrow g]]] \quad (d_1) \\
708 \quad & \rightsquigarrow_C x\langle x \rangle.((f_1\langle a_1, b_1 \rangle.y_1\langle a_1, b_1 \rangle.a_1b_1)((f_2\langle a_2, b_2 \rangle.y_2\langle a_2, b_2 \rangle.a_2b_2)x)) \\
709 \quad & \quad [f_1\langle a_1, b_1 \rangle, f_2\langle a_2, b_2 \rangle \mid g\langle g \rangle[y_1\langle a_1, b_1 \rangle, y_2\langle a_2, b_2 \rangle \mid y\langle y \rangle \\
710 \quad & \quad [b_1, b_2 \leftarrow g_2y][a_1, a_2, g_2 \leftarrow g]]] \quad (c_1) \\
711 \quad & \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_1c_1)) \\
712 \quad & \quad ((f_2\langle a_2, b_2, c_2 \rangle.y_2\langle a_2, b_2, c_2 \rangle.a_2(b_2c_2))x) \\
713 \quad & \quad [f_1\langle a_1, b_1, c_1 \rangle, f_2\langle a_2, b_2, c_2 \rangle \mid g\langle g \rangle[y_1\langle a_1, b_1, c_1 \rangle, y_2\langle a_2, b_2, c_2 \rangle \mid y\langle y \rangle \\
714 \quad & \quad [b_1, b_2 \leftarrow g_2][c_1, c_2 \leftarrow y][a_1, a_2, g_2 \leftarrow g]]] \quad (d_1) \\
715 \quad & \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_1c_1)) \\
716 \quad & \quad ((f_2\langle a_2, b_2, c_2 \rangle.y_2\langle a_2, b_2, c_2 \rangle.a_2(b_2c_2))x) \\
717 \quad & \quad [f_1\langle a_1, b_1, c_1 \rangle, f_2\langle a_2, b_2, c_2 \rangle \mid g\langle g \rangle[y_1\langle a_1, b_1, c_1 \rangle, y_2\langle a_2, b_2, c_2 \rangle \mid y\langle y \rangle \\
718 \quad & \quad [c_1, c_2 \leftarrow y][a_1, b_1, a_2, b_2 \leftarrow g]]] \quad (c_1) \\
719 \quad & \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_1c_1)) \\
720 \quad & \quad ((f_2\langle a_2, b_2, c_2 \rangle.y_2\langle c_2 \rangle.a_2(b_2c_2))x) \\
721 \quad & \quad [f_1\langle a_1, b_1, c_1 \rangle, f_2\langle a_2, b_2, c_2 \rangle \mid g\langle g \rangle[y_1\langle c_1 \rangle, y_2\langle c_2 \rangle \mid y\langle y \rangle
\end{aligned}$$

$$\begin{aligned}
 & [c_1, c_2 \leftarrow y][a_1, b_1, a_2, b_2 \leftarrow g] \tag{I_5} \\
 \rightsquigarrow_L & x \langle x \rangle . ((f_1 \langle a_1, b_1 \rangle . y_1 \langle c_1 \rangle . a_1 (b_1 c_1)) ((f_2 \langle a_2, b_2 \rangle . y_2 \langle c_2 \rangle . a_2 (b_2 c_2)) x)) \\
 & [f_1 \langle a_1, b_1 \rangle, f_2 \langle a_2, b_2 \rangle | g \langle g \rangle [a_1, b_1, a_2, b_2 \leftarrow g]] \\
 & [y_1 \langle c_1 \rangle, y_2 \langle c_2 \rangle | y \langle y \rangle [c_1, c_2 \leftarrow y]] \tag{I_5} \\
 \rightsquigarrow_D & x \langle x \rangle . ((f_1 \langle f_1 \rangle . y_1 \langle c_1 \rangle . a_1 (b_1 c_1) [a_1, b_1 \leftarrow f_1]) \\
 & ((f_2 \langle f_2 \rangle . y_2 \langle c_2 \rangle . a_2 (b_2 c_2) [a_2, b_2 \leftarrow f_2]) x) \\
 & [y_1 \langle c_1 \rangle, y_2 \langle c_2 \rangle | y \langle y \rangle [c_1, c_2 \leftarrow y]] \tag{d_3} \\
 \rightsquigarrow_D & x \langle x \rangle . ((f_1 \langle f_1 \rangle . y_1 \langle y_1 \rangle . a_1 (b_1 y_1) [a_1, b_1 \leftarrow f_1]) \\
 & ((f_2 \langle f_2 \rangle . y_2 \langle y_2 \rangle . a_2 (b_2 y_2) [a_2, b_2 \leftarrow f_2]) x) \tag{d_3}
 \end{aligned}$$

719
720

721 In this section we provide the proof for Proposition 41: Given $M \in \Lambda$ such that for all
 722 $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$, the translation $\llbracket u | \sigma | \gamma \rrbracket$ commutes with substitution
 723 $\{M/x\}$ in the following way

$$\llbracket u \{t/x\} | \sigma | \gamma \rrbracket = \llbracket u | \sigma [x \mapsto \llbracket t | \sigma | \gamma \rrbracket] | \gamma \rrbracket$$

724 **Proof.** We prove this by induction on u

725

Base Case: Variable

$$727 \llbracket x \{t/x\} | \sigma | \gamma \rrbracket = \llbracket t | \sigma | \gamma \rrbracket = \llbracket x | \sigma' | \gamma \rrbracket$$

728

$$729 \llbracket y | \sigma | \gamma \rrbracket = \sigma(y) = \sigma'(y) = \llbracket y | \sigma' | \gamma \rrbracket$$

730

Inductive Case: Application

$$732 \llbracket u s \{t/x\} | \sigma | \gamma \rrbracket = \llbracket u \{t/x\} | \sigma | \gamma \rrbracket \llbracket s \{t/x\} | \sigma | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u | \sigma' | \gamma \rrbracket \llbracket s | \sigma' | \gamma \rrbracket = \llbracket u s | \sigma' | \gamma \rrbracket$$

733

Inductive Case: Abstraction

$$735 \llbracket (c \langle c \rangle . s) \{t/x\} | \sigma | \gamma \rrbracket = \lambda c. \llbracket s \{t/x\} | \sigma | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda c. \llbracket s | \sigma' | \gamma \rrbracket = \llbracket c \langle c \rangle . s | \sigma' | \gamma \rrbracket$$

736

Inductive Case: Phantom-Abstraction

$$738 \llbracket (c \langle x_1, \dots, x_n \rangle . s) \{t/x\} | \sigma | \gamma \rrbracket$$

$$739 \text{ Case: } x \in \{x_1, \dots, x_n\}$$

$$740 = \llbracket (c \langle x_1, \dots, x_n, x \rangle . s) \{t/x\} | \sigma | \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n, y_1, \dots, y_m \rangle . s \{t/x\} | \sigma | \gamma \rrbracket$$

$$741 \text{ where } \{y_1, \dots, y_m\} = (t)_{fv}$$

$$742 = \lambda c. \llbracket s \{t/x\} | \sigma'' | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda c. \llbracket s | \sigma_1''' | \gamma \rrbracket = \lambda c. \llbracket s | \sigma_2''' | \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n, x \rangle . s | \sigma' | \gamma \rrbracket$$

$$743 \text{ where } \sigma''(z) = \begin{cases} \sigma(z) \{c/\gamma(c)\} & \text{if } z \in \{x_1, \dots, x_n, y_1, \dots, y_m\} \\ \sigma(z) & \text{otherwise} \end{cases}$$

$$744 \sigma_1''' = \sigma''[x \mapsto \llbracket t | \sigma'' | \gamma \rrbracket]$$

$$745 \sigma_2'''(z) = \begin{cases} \llbracket t | \sigma'' | \gamma \rrbracket \{c/\gamma(c)\} & z = x \\ \sigma(z) \{c/\gamma(c)\} & z \in \{x_1, \dots, x_n\} \\ \sigma(z) & \text{otherwise} \end{cases}$$

746

$$747 \text{ Case: } x \notin \{x_1, \dots, x_n\}$$

$$748 = \llbracket c \langle x_1, \dots, x_n \rangle . s \{t/x\} | \sigma | \gamma \rrbracket = \lambda c. \llbracket s \{t/x\} | \sigma'' | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda c. \llbracket t | \sigma''[x \mapsto \llbracket t | \sigma'' | \gamma \rrbracket] | \gamma \rrbracket =$$

$$749 \lambda c. \llbracket t | \sigma''[x \mapsto \llbracket t | \sigma | \gamma \rrbracket] | \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n \rangle . s | \sigma[x \mapsto \llbracket t | \sigma | \gamma \rrbracket] | \gamma \rrbracket$$

750 where

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

752

753 Inductive Case: Sharing

$$\llbracket u[z_1, \dots, z_n \leftarrow s]\{t/x\} \mid \sigma \mid \gamma \rrbracket = \llbracket u\{t/x\}[z_1, \dots, z_n \leftarrow s\{t/x\}] \mid \sigma \mid \gamma \rrbracket = \llbracket u\{t/x\} \mid \sigma'' \mid \gamma \rrbracket$$

$$\stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma''' \mid \gamma \rrbracket = \llbracket u[z_1, \dots, z_n \leftarrow s] \mid \sigma' \mid \gamma \rrbracket$$

756 where

$$\sigma'' = \sigma[z_1 \mapsto \llbracket s\{t/x\} \mid \sigma \mid \gamma \rrbracket, \dots, z_n \mapsto \llbracket s\{t/x\} \mid \sigma \mid \gamma \rrbracket]$$

$$\sigma''' = \sigma'[z_1 \mapsto \llbracket s \mid \sigma' \mid \gamma \rrbracket, \dots, z_n \mapsto \llbracket s \mid \sigma' \mid \gamma \rrbracket]$$

759

760 Inductive Case: Distributor 1

$$\llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle \bar{c} \rangle [\bar{\Gamma}]]\{t/x\} \mid \sigma \mid \gamma \rrbracket$$

$$= \llbracket u[\bar{\Gamma}]\{t/x\} \mid \sigma \mid \gamma' \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u[\bar{\Gamma}] \mid \sigma' \mid \gamma' \rrbracket$$

$$= \llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle \bar{x} \rangle [\bar{\Gamma}]] \mid \sigma' \mid \gamma \rrbracket$$

764 where

$$\gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$$

766

767 Inductive Case: Distributor 2

$$\llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle \bar{x} \rangle [\bar{\Gamma}]]\{t/x\} \mid \sigma \mid \gamma \rrbracket$$

$$= \llbracket u[\bar{\Gamma}]\{t/x\} \mid \sigma'' \mid \gamma' \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u[\bar{\Gamma}] \mid \sigma''' \mid \gamma' \rrbracket$$

$$= \llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle \bar{x} \rangle [\bar{\Gamma}]] \mid \sigma' \mid \gamma \rrbracket$$

771 where

$$\gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$$

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

775 if $c \langle y_1, \dots, y_m \rangle. \in (u)_{fc}$ such that $\{x_1, \dots, x_n\} \subset \{y_1, \dots, y_m\}$

776 and for those $z \in \{y_1, \dots, y_m\} / \{x_1, \dots, x_n\}$, $\gamma(c) \notin (\sigma(z))_{fv}$

777 or if simply $\{x_1, \dots, x_n\} \cap (u)_{fv} = \{\}$

$$\llbracket u\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma \mid \gamma \rrbracket$$

778 **Proof.** We prove this by induction on u

779

780 Base Case: Variable

$$\llbracket x\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \llbracket x \mid \sigma \mid \gamma \rrbracket = \sigma(x) = \sigma'(x) = \llbracket x \mid \sigma' \mid \gamma' \rrbracket$$

782 Since it cannot be that $x \in \{x_1, \dots, x_n\}$

783

784 Base Case: Phantom-Abstraction

$$\llbracket (c \langle y_1, \dots, y_m \rangle. t)\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n \rangle. t \mid \sigma \mid \gamma \rrbracket$$

$$= \lambda c. \llbracket t \mid \sigma'' \mid \gamma \rrbracket = \lambda c. \llbracket t \mid \sigma'' \mid \gamma' \rrbracket = \llbracket c \langle y_1, \dots, y_m \rangle. t \mid \sigma' \mid \gamma' \rrbracket$$

787 where

$$\sigma = \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n]$$

$$\sigma'' = \sigma_1[x_1 \mapsto M_1\{c/d\}, \dots, x_n \mapsto M_n\{c/d\}]$$

$$\gamma(c) = d$$

791 Note: due to condition of Proposition any $\{y_i \mapsto M_i\{c/d\}\} = \{y_i \mapsto M_i\}$

792

793 Base Case: Distributor

$$\llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle y_1, \dots, y_m \rangle [\bar{\Gamma}]]\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket$$

$$= \llbracket u[e_1 \langle \bar{w}_1 \rangle, \dots, e_n \langle \bar{w}_n \rangle \mid c \langle x_1, \dots, x_n \rangle [\bar{\Gamma}]] \mid \sigma \mid \gamma \rrbracket = \llbracket u[\bar{\Gamma}] \mid \sigma' \mid \gamma' \rrbracket$$

796 $= \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_n \langle \vec{w}_n \rangle | c \langle y_1, \dots, y_m \rangle \overline{[\Gamma]}] | \sigma | \gamma \rrbracket$
 797 where $\gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$
 798 $\sigma = \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n]$
 799 $\sigma' = \sigma_1[x_1 \mapsto M_1\{c/\gamma(c)\}, \dots, x_n \mapsto M_n\{c/\gamma(c)\}]$
 800
 801 Inductive Case: Application
 802 $\llbracket (st)\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket = \llbracket s\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket \llbracket t\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket$
 803 $\stackrel{\text{I.H.}}{=} \llbracket s | \sigma | \gamma \rrbracket \llbracket t | \sigma | \gamma \rrbracket = \llbracket st | \sigma | \gamma \rrbracket$
 804
 805 Inductive Case: Abstraction
 806 $\llbracket (z \langle z \rangle . t)\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket = \lambda z. \llbracket t\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda z. \llbracket t | \sigma | \gamma \rrbracket = \llbracket z \langle z \rangle . t | \sigma | \gamma \rrbracket$
 807
 808 Inductive Case: Phantom-Abstraction
 809 $\llbracket (d \langle z_1, \dots, z_m \rangle . t)\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket = \lambda d. \llbracket t\{x_1, \dots, x_n/c\}_b | \sigma' | \gamma \rrbracket$
 810 $\stackrel{\text{I.H.}}{=} \lambda d. \llbracket t | \sigma' | \gamma \rrbracket = \llbracket d \langle z_1, \dots, z_m \rangle . t | \sigma | \gamma \rrbracket$
 811
 812 Inductive Case: Sharing
 813 $\llbracket u[z_1, \dots, z_m \leftarrow t]\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket$
 814 $= \llbracket u\{x_1, \dots, x_n/c\}_b[z_1, \dots, z_m \leftarrow t\{x_1, \dots, x_n/c\}_b] | \sigma | \gamma \rrbracket$
 815 $= \llbracket u\{x_1, \dots, x_n/c\}_b | \sigma' | \gamma \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u | \sigma' | \gamma \rrbracket = \llbracket u[z_1, \dots, z_m \leftarrow t] | \sigma | \gamma \rrbracket$
 816
 817 Inductive Case: Distributor
 818 $\llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | d \langle d \rangle \overline{[\Gamma]}]\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket$
 819 $= \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | d \langle d \rangle \overline{[\Gamma]}]\{x_1, \dots, x_n/c\}_b | \sigma | \gamma \rrbracket$
 820 $= \llbracket u[\overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b | \sigma | \gamma' \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b | \sigma | \gamma' \rrbracket$
 821 $= \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | d \langle d \rangle \overline{[\Gamma]}] | \sigma | \gamma \rrbracket$ ◀

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

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

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

825 **Proof.** We prove this by induction on u

826 Base Case: Variable

$$827 \llbracket z \{c \langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket = \llbracket z | \sigma | \gamma \rrbracket = \sigma(z) = \sigma'(z) = \llbracket z | \sigma' | \gamma \rrbracket$$

828 Base Case: Phantom-Abstraction

$$829 \llbracket (c \langle x_1, \dots, x_n \rangle . t)\{c \langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket = \llbracket c \langle c \rangle . t[x_1, \dots, x_n \leftarrow c] | \sigma | \gamma \rrbracket$$

$$830 = \lambda c. \llbracket t[x_1, \dots, x_n \leftarrow c] | \sigma | \gamma \rrbracket = \lambda c. \llbracket t | \sigma' | \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n \rangle . t | \sigma' | \gamma \rrbracket$$

831 Base Case: Distributor

$$832 \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle x_1, \dots, x_n \rangle \overline{[\Gamma]}]\{c \langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket$$

$$833 = \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle c \rangle \overline{[\Gamma]}][x_1, \dots, x_n \leftarrow c] | \sigma | \gamma \rrbracket$$

$$834 = \llbracket u[\overline{[\Gamma]}][x_1, \dots, x_n \leftarrow c] | \sigma | \gamma' \rrbracket = \llbracket u[\overline{[\Gamma]}\{x_1, \dots, x_n/c\}_b | \sigma' | \gamma' \rrbracket$$

$$835 = \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] | \sigma' | \gamma \rrbracket$$

836 Inductive Case: Application

$$\begin{aligned} & \llbracket (st)\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \llbracket s\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket \llbracket t\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket \\ & \stackrel{\text{I.H.}}{=} \llbracket s \mid \sigma' \mid \gamma \rrbracket \llbracket t \mid \sigma' \mid \gamma \rrbracket = \llbracket st \mid \sigma' \mid \gamma \rrbracket \end{aligned}$$

Inductive Case: Abstraction

$$\begin{aligned} & \llbracket (z\langle z \rangle.t)\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \lambda z. \llbracket t\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket \\ & \stackrel{\text{I.H.}}{=} \lambda z. \llbracket t \mid \sigma' \mid \gamma \rrbracket = \llbracket z\langle z \rangle.t \mid \sigma' \mid \gamma \rrbracket \end{aligned}$$

Inductive Case: Phantom-Abstraction

$$\begin{aligned} & \llbracket (d\langle z_1, \dots, z_m \rangle.t)\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket = \lambda d. \llbracket t\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma'' \mid \gamma \rrbracket \\ & \stackrel{\text{I.H.}}{=} \lambda d. \llbracket t \mid \sigma''' \mid \gamma \rrbracket = \llbracket d\langle z_1, \dots, z_m \rangle.t \mid \sigma' \mid \gamma \rrbracket \end{aligned}$$

Inductive Case: Sharing

$$\begin{aligned} & \llbracket u[z_1, \dots, z_m \leftarrow t]\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket \\ & = \llbracket u\{c\langle x_1, \dots, x_n \rangle\}_e[z_1, \dots, z_m \leftarrow t\{c\langle x_1, \dots, x_n \rangle\}_e] \mid \sigma \mid \gamma \rrbracket \\ & = \llbracket u\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma'' \mid \gamma \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma''' \mid \gamma \rrbracket = \llbracket u[z_1, \dots, z_m \leftarrow t] \mid \sigma' \mid \gamma \rrbracket \end{aligned}$$

Inductive Case: Distributor

$$\begin{aligned} & \llbracket u[e_1\langle \vec{w}_1 \rangle, \dots, e_m\langle \vec{w}_m \rangle \mid d\langle d \rangle \overline{[\Gamma]}]\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket \\ & = \llbracket u[e_1\langle \vec{w}_1 \rangle, \dots, e_m\langle \vec{w}_m \rangle \mid d\langle d \rangle \overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e] \mid \sigma \mid \gamma \rrbracket \\ & = \llbracket u[\overline{[\Gamma]}\{c\langle x_1, \dots, x_n \rangle\}_e \mid \sigma \mid \gamma'] \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]} \mid \sigma' \mid \gamma'] \rrbracket \\ & = \llbracket u[e_1\langle \vec{w}_1 \rangle, \dots, e_m\langle \vec{w}_m \rangle \mid d\langle d \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma' \rrbracket \end{aligned}$$

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

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

Case: (c_1)

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

$$\llbracket u[\vec{w} \leftarrow y][\vec{x} \cdot y \leftarrow t] \mid \sigma \mid \gamma \rrbracket = \llbracket u[\vec{w} \leftarrow y] \mid \sigma' \mid \gamma \rrbracket = \llbracket u \mid \sigma'' \mid \gamma \rrbracket = \llbracket u[\vec{x} \cdot \vec{w} \leftarrow t] \mid \sigma \mid \gamma \rrbracket$$

where

$$\sigma' = \sigma[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{\forall x \in \vec{x}}[y \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]$$

$$\sigma'' = \sigma'[w \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{\forall w \in \vec{w}}$$

Case: (c_2)

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

$$\llbracket u[x \leftarrow t] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket] \mid \gamma \rrbracket = \llbracket u\{t/x\} \mid \sigma \mid \gamma \rrbracket$$

Case: (d_1)

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

$$\llbracket u[x_1 \dots x_n \leftarrow st] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma' \mid \gamma \rrbracket$$

where

$$\sigma' = \sigma[x_i \mapsto \llbracket st \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} = \sigma[x_i \mapsto \llbracket s \mid \sigma \mid \gamma \rrbracket \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n}$$

$$\llbracket u\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket$$

$$= \llbracket u\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\} \mid \sigma'' \mid \gamma \rrbracket$$

where

$$\begin{aligned}\sigma'' &= \sigma[z_i \mapsto \llbracket s \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} [y_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} \text{ since } y_i \notin (s)_{fv} \\ &= \llbracket u \mid \sigma''' \mid \gamma \rrbracket\end{aligned}$$

where

$$\begin{aligned}\sigma''' &= \sigma''[x_i \mapsto \llbracket z_i y_i \mid \sigma'' \mid \gamma \rrbracket]_{1 \leq i \leq n} = \sigma[x_i \mapsto \llbracket s \mid \sigma \mid \gamma \rrbracket \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} \\ &\text{since } z_i \text{ and } y_i \notin (u)_{fv}\end{aligned}$$

Case: (d_2)

$$\begin{aligned}u[x_1, \dots, x_n \leftarrow c\langle \vec{y} \rangle.t] &\rightsquigarrow_D \\ u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle w_1^1 \rangle \dots e_n\langle w_1^n \rangle \mid c\langle \vec{y} \rangle[w_1^1, \dots, w_1^n \leftarrow t]]\end{aligned}$$

SubCase: $\vec{y} = c$

$$\begin{aligned}\llbracket u[x_1, \dots, x_n \leftarrow c\langle c \rangle.t] \mid \sigma \mid \gamma \rrbracket &= \llbracket u \mid \sigma' \mid \gamma \rrbracket \\ \text{where } \sigma' &= \sigma[x_i \mapsto \llbracket c\langle c \rangle.t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} = \sigma[x_i \mapsto \lambda c. \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n}\end{aligned}$$

$$\begin{aligned}&\llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle w_1^1 \rangle \dots e_n\langle w_1^n \rangle \mid c\langle c \rangle[w_1^1, \dots, w_1^n \leftarrow t]] \mid \sigma \mid \gamma \rrbracket \\ &\llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t] \mid \sigma \mid \gamma' \rrbracket \\ &= \llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n} \mid \sigma' \mid \gamma' \rrbracket\end{aligned}$$

where

$$\begin{aligned}\gamma' &= \gamma[e_1 \mapsto c, \dots, e_n \mapsto c] \\ \sigma' &= \sigma[w_1^i \mapsto \llbracket t \mid \sigma \mid \gamma' \rrbracket]_{1 \leq i \leq n} = \sigma[w_1^i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} \\ &= \llbracket u \mid \sigma'' \mid \gamma' \rrbracket = \llbracket u \mid \sigma'' \mid \gamma \rrbracket\end{aligned}$$

where

$$\begin{aligned}\sigma'' &= \sigma'[x_i \mapsto \llbracket e_i\langle w_1^i \rangle.w_1^i \mid \sigma' \mid \gamma' \rrbracket]_{1 \leq i \leq n} = \sigma'[x_i \mapsto \lambda e_i. \llbracket w_1^i \mid \sigma'_i \mid \gamma' \rrbracket]_{1 \leq i \leq n} \\ &= \sigma'[x_i \mapsto \lambda e_i. \llbracket t \mid \sigma \mid \gamma \rrbracket \{e_i/c\}]_{1 \leq i \leq n} =_\alpha \sigma'[x_i \mapsto \lambda c. \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} \\ &= \sigma[x_i \mapsto \lambda c. \llbracket t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} \text{ since } w_1^i \notin (u)_{fv}\end{aligned}$$

SubCase: $\vec{y} = \{y_1, \dots, y_m\}$

$$\llbracket u[x_1, \dots, x_n \leftarrow c\langle y_1, \dots, y_m \rangle.t] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma' \mid \gamma \rrbracket$$

where

$$\begin{aligned}\sigma' &= \sigma[x_i \mapsto \llbracket c\langle y_1, \dots, y_m \rangle.t \mid \sigma \mid \gamma \rrbracket]_{1 \leq i \leq n} = \sigma[x_i \mapsto \lambda c. \llbracket t \mid \sigma'' \mid \gamma \rrbracket]_{1 \leq i \leq n} \\ \sigma &= \sigma_1[y_1 \mapsto M_1, \dots, y_m \mapsto M_m] \\ \sigma'' &= \sigma_1[y_1 \mapsto M_1\{c/\gamma(c)\}, \dots, y_m \mapsto M_m\{c/\gamma(c)\}]\end{aligned}$$

$$\begin{aligned}&\llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle w_1^1 \rangle \dots e_n\langle w_1^n \rangle \mid c\langle y_1, \dots, y_m \rangle[w_1^1, \dots, w_1^n \leftarrow t]] \mid \sigma \mid \gamma \rrbracket \\ &\llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t] \mid \sigma'' \mid \gamma' \rrbracket\end{aligned}$$

where $\gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$

$$= \llbracket u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n} \mid \sigma''' \mid \gamma' \rrbracket$$

$$\text{where } \sigma''' = \sigma''[w_1^i \mapsto \llbracket t \mid \sigma'' \mid \gamma' \rrbracket]_{1 \leq i \leq n} = \sigma''[w_1^i \mapsto \llbracket t \mid \sigma'' \mid \gamma \rrbracket]_{1 \leq i \leq n}$$

$$= \llbracket u \mid \sigma'''' \mid \gamma' \rrbracket = \llbracket u \mid \sigma'''' \mid \gamma \rrbracket = \llbracket u \mid \sigma'' \mid \gamma \rrbracket$$

$$\begin{aligned}\text{where } \sigma'''' &= \sigma'''[x_i \mapsto \llbracket e_i\langle w_1^i \rangle.w_1^i \mid \sigma''' \mid \gamma' \rrbracket]_{1 \leq i \leq n} = \sigma'''[x_i \mapsto \lambda e_i. \llbracket w_1^i \mid \sigma'''_i \mid \gamma' \rrbracket]_{1 \leq i \leq n} \\ &= \sigma'''[x_i \mapsto \lambda e_i. \llbracket t \mid \sigma'' \mid \gamma \rrbracket \{e_i/\gamma'(e_i)\}]_{1 \leq i \leq n} =_\alpha \sigma'''[x_i \mapsto \lambda c. \llbracket t \mid \sigma'' \mid \gamma \rrbracket]_{1 \leq i \leq n}\end{aligned}$$

Case: (d_3)

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

$$\begin{aligned}&\llbracket u\{e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \mid c\langle c \rangle[\vec{w}_1, \dots, \vec{w}_n \leftarrow c]\} \mid \sigma \mid \gamma \rrbracket \\ &= \llbracket u[\vec{w}_1, \dots, \vec{w}_n \leftarrow c] \mid \sigma \mid \gamma' \rrbracket = \llbracket u \mid \sigma' \mid \gamma' \rrbracket\end{aligned}$$

$$= \llbracket u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e \mid \sigma \mid \gamma' \rrbracket = \llbracket u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e \mid \sigma \mid \gamma \rrbracket$$

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

Case: (l_1)

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

$$\llbracket s[\Gamma] t \mid \sigma \mid \gamma \rrbracket = \llbracket s \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket \llbracket t \mid \sigma \mid \gamma \rrbracket = \llbracket s \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket (st)[\Gamma] \mid \sigma \mid \gamma \rrbracket$$

Case: (l_2)

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

$$\llbracket s(t[\Gamma]) \mid \sigma \mid \gamma \rrbracket = \llbracket s \mid \sigma \mid \gamma \rrbracket \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket s \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket (st)[\Gamma] \mid \sigma \mid \gamma \rrbracket$$

Case: (l_3)

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

SubCase: $\vec{x} = d$

$$\llbracket d\langle d \rangle.t[\Gamma] \mid \sigma \mid \gamma \rrbracket = \lambda d. \llbracket t[\Gamma] \mid \sigma \mid \gamma \rrbracket = \lambda d. \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket d\langle d \rangle.t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket (d\langle d \rangle.t)[\Gamma] \mid \sigma \mid \gamma \rrbracket$$

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

$$\llbracket d\langle x_1, \dots, x_n \rangle.t[\Gamma] \mid \sigma \mid \gamma \rrbracket = \lambda d. \llbracket t[\Gamma] \mid \sigma' \mid \gamma \rrbracket = \lambda d. \llbracket t \mid \sigma'_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket d\langle x_1, \dots, x_n \rangle.t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket$$

$$= \llbracket (d\langle x_1, \dots, x_n \rangle.t)[\Gamma] \mid \sigma \mid \gamma \rrbracket$$

since we know $x_1, \dots, x_n \notin ([\Gamma])_{fv}$

Case: (l_4)

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

$$\llbracket u[\vec{x} \leftarrow t[\Gamma]] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma' \mid \gamma \rrbracket = \llbracket u \mid \sigma'' \mid \gamma_\Gamma \rrbracket = \llbracket u[\vec{x} \leftarrow t] \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket = \llbracket u[\vec{x} \leftarrow t][\Gamma] \mid \sigma \mid \gamma \rrbracket$$

where

$$\sigma' = \sigma[x \mapsto \llbracket t[\Gamma] \mid \sigma \mid \gamma \rrbracket]_{\forall x \in \vec{x}} = \sigma[x \mapsto \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket]_{\forall x \in \vec{x}}$$

$$\sigma'' = \sigma_\Gamma[x \mapsto \llbracket t \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket]_{\forall x \in \vec{x}}$$

Cases: (l_5)

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

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

863 SubCase: $\vec{x} = c$

864 $\llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}[\Gamma]] \mid \sigma \mid \gamma \rrbracket = \llbracket u[\overline{[\Gamma]}[\Gamma]] \mid \sigma \mid \gamma' \rrbracket = \llbracket u[\overline{[\Gamma]}] \mid \sigma_\Gamma \mid \gamma'_\Gamma \rrbracket$

865 $= \llbracket u[\overline{[\Gamma]}\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \mid \sigma_\Gamma \mid \gamma'_\Gamma \rrbracket = \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \overline{[\Gamma]} \mid \sigma_\Gamma \mid \gamma'_\Gamma \rrbracket$

866 $= \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket$

867 $= \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] [\Gamma] \mid \sigma \mid \gamma \rrbracket$

868

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

870 $\llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle \mid c\langle x_1, \dots, x_m \rangle \overline{[\Gamma]}[\Gamma]] \mid \sigma \mid \gamma \rrbracket$

871 $= \llbracket u[\overline{[\Gamma]}[\Gamma]] \mid \sigma' \mid \gamma' \rrbracket = \llbracket u[\overline{[\Gamma]}\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \mid \sigma_\Gamma \mid \gamma'_\Gamma \rrbracket$

872 $= \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b \overline{[\Gamma]} \mid \sigma_\Gamma \mid \gamma'_\Gamma \rrbracket$

873 $= \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \mid c\langle x_1, \dots, x_m \rangle \overline{[\Gamma]}] \mid \sigma_\Gamma \mid \gamma_\Gamma \rrbracket$

874 $= \llbracket u\{\vec{z}_1/e_1\}_b \dots \{\vec{z}_n/e_n\}_b [e_1\langle \vec{z}_1 \rangle \dots e_n\langle \vec{z}_n \rangle \mid c\langle x_1, \dots, x_m \rangle \overline{[\Gamma]}] [\Gamma] \mid \sigma \mid \gamma \rrbracket$

875

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

$$877 \llbracket ts \mid \sigma \mid \gamma \rrbracket = \llbracket t \mid \sigma \mid \gamma \rrbracket \llbracket s \mid \sigma \mid \gamma \rrbracket \stackrel{\text{I.H.}}{=} \llbracket t' \mid \sigma \mid \gamma \rrbracket \llbracket s \mid \sigma \mid \gamma \rrbracket = \llbracket t's \mid \sigma \mid \gamma \rrbracket$$

878

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

$$880 \llbracket ts \mid \sigma \mid \gamma \rrbracket = \llbracket t \mid \sigma \mid \gamma \rrbracket \llbracket s \mid \sigma \mid \gamma \rrbracket \stackrel{\text{I.H.}}{=} \llbracket t \mid \sigma \mid \gamma \rrbracket \llbracket s' \mid \sigma \mid \gamma \rrbracket = \llbracket ts' \mid \sigma \mid \gamma \rrbracket$$

881

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

$$883 \llbracket x \langle x \rangle . t \mid \sigma \mid \gamma \rrbracket = \lambda x. \llbracket t \mid \sigma[x \mapsto x] \mid \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda x. \llbracket t' \mid \sigma[x \mapsto x] \mid \gamma \rrbracket = \llbracket x \langle x \rangle . t' \mid \sigma \mid \gamma \rrbracket$$

884

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

$$886 \llbracket c \langle \vec{x} \rangle . t \mid \sigma \mid \gamma \rrbracket = \lambda c. \llbracket t \mid \sigma' \mid \gamma \rrbracket \stackrel{\text{I.H.}}{=} \lambda c. \llbracket t' \mid \sigma' \mid \gamma \rrbracket = \llbracket c \langle \vec{x} \rangle . t' \mid \sigma \mid \gamma \rrbracket$$

887

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

$$889 \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i \in [n]} \mid \gamma \rrbracket$$

$$890 \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma[x_i \mapsto \llbracket t' \mid \sigma \mid \gamma \rrbracket]_{i \in [n]} \mid \gamma \rrbracket = \llbracket u[x_1, \dots, x_n \leftarrow t'] \mid \sigma \mid \gamma \rrbracket$$

891

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

$$893 \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket = \llbracket u \mid \sigma[x_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i \in [n]} \mid \gamma \rrbracket$$

$$894 \stackrel{\text{I.H.}}{=} \llbracket u' \mid \sigma[x_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i \in [n]} \mid \gamma \rrbracket = \llbracket u'[x_1, \dots, x_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket$$

895

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

$$897 \llbracket u[\overrightarrow{e \langle \vec{x} \rangle} \mid c \langle c \rangle [\overline{\Gamma}]] \mid \sigma \mid \gamma \rrbracket = \llbracket u[\overline{\Gamma}] \mid \sigma \mid \gamma' \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u'[\overline{\Gamma'}] \mid \sigma \mid \gamma' \rrbracket = \llbracket u'[\overrightarrow{e \langle \vec{x}' \rangle} \mid c \langle c \rangle [\overline{\Gamma'}]] \mid \sigma \mid \gamma \rrbracket$$

898

◀

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-deleting 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 extension 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 $\llbracket u \mid \sigma \mid \gamma \rrbracket_w$ commutes with substitution $\{M/x\}$ in the following way*

$$\llbracket u\{t/x\} \mid \sigma \mid \gamma \rrbracket_w = \llbracket u \mid \sigma[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \mid \gamma \rrbracket_w$$

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

$$\begin{aligned} \llbracket u[\leftarrow s]\{t/x\} \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u\{t/x\} \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket s\{t/x\} \mid \sigma \mid \gamma \rrbracket_w] \\ &\stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma' \mid \gamma \rrbracket_w [\leftarrow \llbracket s \mid \sigma' \mid \gamma \rrbracket_w] = \llbracket u[\leftarrow s] \mid \sigma' \mid \gamma \rrbracket_w \end{aligned}$$

Inductive Case: Distributor

$$\llbracket u[\mid c\langle \vec{x} \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w$$

SubCase: $\vec{x} = c$

$$\begin{aligned} \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w \\ &= \llbracket u[\overline{[\Gamma]}] \{t/x\} \mid \sigma'' \mid \gamma' \rrbracket_w \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] \mid \sigma''' \mid \gamma' \rrbracket_w = \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket_w \end{aligned}$$

where

$$\begin{aligned} \sigma'' &= \sigma[c \mapsto \bullet] \\ \sigma''' &= \sigma[c \mapsto \bullet][x \mapsto \llbracket t \mid \sigma'' \mid \gamma' \rrbracket_w] = \sigma[c \mapsto \bullet][x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \end{aligned}$$

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

$$\llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w$$

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

$$\begin{aligned} \llbracket u[\mid c\langle x_1, \dots, x_n, x \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w \\ \llbracket u[\mid c\langle x_1, \dots, x_n, y_1, \dots, y_m \rangle \overline{[\Gamma]}] \{t/x\} \mid \sigma \mid \gamma \rrbracket_w \end{aligned}$$

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

$$= \llbracket u[\overline{[\Gamma]}] \{t/x\} \mid \sigma'' \mid \gamma \rrbracket_w$$

944 where

$$\begin{aligned}
 945 \quad \sigma &= \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n, y_1 \mapsto N_1, \dots, y_m \mapsto N_m] \\
 946 \quad \sigma'' &= \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}, y_1 \mapsto N_1\{\bullet/\gamma(c)\}, \dots, y_m \mapsto N_m\{\bullet/\gamma(c)\}] \\
 947 \quad &\stackrel{\text{I.H.}}{=} \llbracket u[\overline{\Gamma}] \mid \sigma'' \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid c\langle x_1, \dots, x_n, x \rangle \overline{\Gamma} \mid] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} \\
 948 \quad &\text{where } \sigma''' = \sigma''[x \mapsto \llbracket t \mid \sigma'' \mid \gamma \rrbracket_{\mathcal{W}}] = \sigma''[x \mapsto \llbracket t \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}}\{\bullet/\gamma(c)\}] \\
 949 \quad &\text{since } \{y_1, \dots, y_m\} = (t)_{fv}
 \end{aligned}$$

950

$$\begin{aligned}
 951 \quad &\text{SubSubCase: } \vec{x} = x_1, \dots, x_n \\
 952 \quad &\llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{\Gamma} \mid] \{t/x\} \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{\Gamma} \mid] \{t/x\} \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 953 \quad &\llbracket u[\overline{\Gamma}] \{t/x\} \mid \sigma'' \mid \gamma \rrbracket_{\mathcal{W}} \stackrel{\text{I.H.}}{=} \llbracket u[\overline{\Gamma}] \mid \sigma''' \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{\Gamma} \mid] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} \\
 954 \quad &\sigma = \sigma_1[x_1 \mapsto M_1, \dots, x_n \mapsto M_n] \\
 955 \quad &\sigma'' = \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}] \\
 956 \quad &\sigma''' = \sigma''[x \mapsto \llbracket t \mid \sigma'' \mid \gamma \rrbracket_{\mathcal{W}}] = \sigma''[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}] \\
 957 \quad &\text{since } \{x_1, \dots, x_n\} \cap (t)_{fv} = \{\} \quad \blacktriangleleft
 \end{aligned}$$

958 ► **Proposition 42.** *Book-keeping commutes with the translation in the following way*

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

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

962 **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.

964

965 Inductive Case: Weakening

$$\begin{aligned}
 966 \quad &\llbracket u[\leftarrow t]\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}[\leftarrow \llbracket t\{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}] \\
 967 \quad &\stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}[\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}] = \llbracket u[\leftarrow t] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}
 \end{aligned}$$

968

969 Base Case: Distributor

$$\begin{aligned}
 970 \quad &\llbracket u[\mid c\langle \vec{x} \rangle \overline{\Gamma} \mid] \{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{\Gamma} \mid] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 971 \quad &\llbracket u[\overline{\Gamma}] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid c\langle \vec{x} \rangle \overline{\Gamma} \mid] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 972 \quad &\text{where } \sigma' = \sigma[x_1 \mapsto \sigma(x_1)\{\bullet/\gamma(c)\}, \dots, x_n \mapsto \sigma(x_n)\{\bullet/\gamma(c)\}] \\
 973 \quad &\text{and notice for } x_i \neq y \in \vec{x}, [y \mapsto N] = [y \mapsto N\{\bullet/\gamma(c)\}]
 \end{aligned}$$

974

975 Inductive Case: Distributor

$$\begin{aligned}
 976 \quad &\llbracket u[\mid d\langle d \rangle \overline{\Gamma} \mid] \{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid d\langle d \rangle \overline{\Gamma} \mid] \{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 977 \quad &\llbracket u[\overline{\Gamma}] \{x_1, \dots, x_n/c\}_b \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} \stackrel{\text{I.H.}}{=} \llbracket u[\overline{\Gamma}] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid d\langle d \rangle \overline{\Gamma} \mid] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 978 \quad &\text{where } \sigma' = \sigma[d \mapsto \bullet]
 \end{aligned}$$

979

$$\begin{aligned}
 980 \quad &\llbracket u[\mid d\langle z_1, \dots, z_n \rangle \overline{\Gamma} \mid] \{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid d\langle z_1, \dots, z_n \rangle \overline{\Gamma} \mid] \{x_1, \dots, x_n/c\}_b \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}} \\
 981 \quad &\llbracket u[\overline{\Gamma}] \{x_1, \dots, x_n/c\}_b \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} \stackrel{\text{I.H.}}{=} \llbracket u[\overline{\Gamma}] \mid \sigma' \mid \gamma \rrbracket_{\mathcal{W}} = \llbracket u[\mid d\langle z_1, \dots, z_n \rangle \overline{\Gamma} \mid] \mid \sigma \mid \gamma \rrbracket_{\mathcal{W}}
 \end{aligned}$$

982 where

$$983 \quad \sigma' = \sigma[z_1 \mapsto \sigma(x_1)\{\bullet/\gamma(d)\}, \dots, z_n \mapsto \sigma(x_n)\{\bullet/\gamma(d)\}] \quad \blacktriangleleft$$

984 ► **Proposition 43.** *Exorcisms commute with the translation in the following way*

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

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

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 the cases involving the three special cases defined in Definition 18.

Inductive Case: Weakening

$$\llbracket u[\leftarrow t]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w = \llbracket u\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w [\leftarrow \llbracket t\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w] \\ \stackrel{\text{I.H.}}{=} \llbracket u | \sigma' | \gamma \rrbracket_w [\leftarrow \llbracket t | \sigma' | \gamma \rrbracket_w] = \llbracket u[\leftarrow t] | \sigma' | \gamma \rrbracket_w$$

Base Case: Distributor

$$\llbracket u[\leftarrow c\langle x_1, \dots, x_n \rangle \overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w = \llbracket u[\leftarrow c\langle c \rangle \overline{\Gamma}][x_1, \dots, x_n \leftarrow c] | \sigma | \gamma \rrbracket_w \\ = \llbracket u[\overline{\Gamma}][x_1, \dots, x_n \leftarrow c] | \sigma'' | \gamma \rrbracket_w = \llbracket u[\overline{\Gamma}] | \sigma''' | \gamma \rrbracket_w = \llbracket u[\leftarrow c\langle x_1, \dots, x_n \rangle \overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma' | \gamma \rrbracket_w$$

where

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

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

Inductive Case: Distributor

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

where

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

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

$$\llbracket u[\leftarrow d\langle z_1, \dots, z_m \rangle \overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w \\ = \llbracket u[\leftarrow d\langle z_1, \dots, z_m \rangle \overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma | \gamma \rrbracket_w \\ = \llbracket u[\overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma'' | \gamma \rrbracket_w \stackrel{\text{I.H.}}{=} \llbracket u[\overline{\Gamma}] | \sigma''' | \gamma \rrbracket_w = \llbracket u[\leftarrow d\langle d \rangle \overline{\Gamma}]\{c\langle x_1, \dots, x_n \rangle\}_e | \sigma' | \gamma \rrbracket_w$$

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 $M \in \Lambda_w$ such that for all $v \in V$, $\gamma(v) \notin (M)_{fv}$ and $\sigma(x) = x$

$$\llbracket u | \sigma' | \gamma \rrbracket = \llbracket u | \sigma | \gamma \rrbracket \{M/x\}$$

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

Proof. We prove this by induction on u

Base Case: Variable

$$\llbracket x | \sigma | \gamma \rrbracket \{M/x\} = x\{M/x\} = M = \llbracket x | \sigma' | \gamma \rrbracket$$

$$\llbracket y | \sigma | \gamma \rrbracket \{M/x\} = y\{M/x\} = \llbracket y | \sigma' | \gamma \rrbracket$$

Inductive Case: Application

$$\llbracket st | \sigma | \gamma \rrbracket \{M/x\} = \llbracket s | \sigma | \gamma \rrbracket \{M/x\} \llbracket t | \sigma | \gamma \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \llbracket s | \sigma | \gamma \rrbracket \llbracket t | \sigma' | \gamma \rrbracket = \llbracket st | \sigma' | \gamma \rrbracket$$

1030 Inductive Case: Abstraction

$$1031 \llbracket c\langle c \rangle . t \mid \sigma \mid \gamma \rrbracket \{M/x\} = \lambda c. \llbracket t \mid \sigma \mid \gamma \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \lambda c. \llbracket t \mid \sigma' \mid \gamma \rrbracket = \llbracket c\langle c \rangle . t \mid \sigma' \mid \gamma \rrbracket$$

1032

1033 Inductive Case: Phantom-Abstraction

$$1034 \llbracket c\langle x_1, \dots, x_n \rangle . t \mid \sigma \mid \gamma \rrbracket \{M/x\} = (\lambda c. \llbracket t \mid \sigma'' \mid \gamma \rrbracket) \{M/x\} = \lambda c. \llbracket t \mid \sigma'' \mid \gamma \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \lambda c. \llbracket t \mid \sigma''' \mid \gamma \rrbracket$$

$$1035 = \llbracket c\langle x_1, \dots, x_n \rangle . t \mid \sigma' \mid \gamma \rrbracket$$

1036 where

$$1037 \sigma'' = \sigma[x_1 \mapsto \sigma(x_1)\{c/d\}, \dots, x_n \mapsto \sigma(x_n)\{c/d\}]$$

$$1038 \sigma''' = \sigma''\{M/x\}[x \mapsto M]$$

$$1039 \sigma''' = \sigma\{M/x\}[x_1 \mapsto \sigma(x_1)\{M/x\}\{c/d\}, \dots, x_n \mapsto \sigma(x_n)\{M/x\}\{c/d\}, x \mapsto M]$$

1040

1041 Inductive Case: Sharing

$$1042 \llbracket u[z_1, \dots, z_n \leftarrow t] \mid \sigma \mid \gamma \rrbracket \{M/x\} = \llbracket u \mid \sigma'' \mid \gamma \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma''' \mid \gamma \rrbracket = \llbracket u[z_1, \dots, z_n \leftarrow t] \mid \sigma' \mid \gamma \rrbracket$$

1043 where

$$1044 \sigma'' = \sigma[z_i \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket]_{i \in [n]}$$

$$1045 \sigma''' = \sigma\{M/x\}[z_i \mapsto \llbracket t \mid \sigma\{x/M\}[x \mapsto M] \mid \gamma \rrbracket, x \mapsto M]_{i \in [n]}$$

1046

1047 Inductive Case: Distributor 1

$$1048 \llbracket u[e_1\langle \bar{w}_1 \rangle, \dots, e_n\langle \bar{w}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket \{M/x\}$$

$$1049 = \llbracket u[\overline{[\Gamma]}] \mid \sigma \mid \gamma' \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] \mid \sigma' \mid \gamma' \rrbracket$$

$$1050 = \llbracket u[e_1\langle \bar{w}_1 \rangle, \dots, e_n\langle \bar{w}_n \rangle \mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket$$

1051 where

$$1052 \gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$$

1053

1054 Inductive Case: Distributor 2

$$1055 \llbracket u[e_1\langle \bar{w}_1 \rangle, \dots, e_n\langle \bar{w}_n \rangle \mid c\langle \bar{x} \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket \{M/x\}$$

$$1056 = \llbracket u[\overline{[\Gamma]}] \mid \sigma'' \mid \gamma' \rrbracket \{M/x\} \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] \mid \sigma''' \mid \gamma' \rrbracket$$

$$1057 = \llbracket u[e_1\langle \bar{w}_1 \rangle, \dots, e_n\langle \bar{w}_n \rangle \mid c\langle \bar{x} \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket$$

1058 where

$$1059 \gamma' = \gamma[e_1 \mapsto c, \dots, e_n \mapsto c]$$

1060

1061 Inductive Case: Weakening

$$1062 \llbracket u[\leftarrow t] \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u \mid \sigma' \mid \gamma \rrbracket [\leftarrow \llbracket t \mid \sigma' \mid \gamma \rrbracket] \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma \mid \gamma \rrbracket \{M/x\} [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket \{M/x\}]$$

$$1063 = \llbracket u \mid \sigma \mid \gamma \rrbracket [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket] \{M/x\} = \llbracket u[\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w \{M/x\}$$

1064

1065 Inductive Case: Distributor

$$1066 \llbracket u[\mid c\langle \bar{x} \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket_w$$

1067

1068 SubCase: $\bar{x} = c$

$$1069 \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u[\overline{[\Gamma]}] \mid \sigma'' \mid \gamma \rrbracket_w \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] \mid \sigma''' \mid \gamma \rrbracket_w \{M/x\}$$

$$1070 = \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket_w \{M/x\}$$

1071 where

$$1072 \sigma''' = \sigma[c \mapsto \bullet]$$

$$1073 \sigma'' = \sigma'[c \mapsto \bullet]$$

1074

1075 SubCase $\bar{x} = x_1, \dots, x_n$

$$1076 \llbracket u[\mid c\langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u[\overline{[\Gamma]}] \mid \sigma'' \mid \gamma \rrbracket_w \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] \mid \sigma''' \mid \gamma \rrbracket_w \{M/x\}$$

$$1077 = \llbracket u[\mid c\langle c \rangle \overline{[\Gamma]}] \mid \sigma \mid \gamma \rrbracket_w \{M/x\}$$

1078 where

$$\begin{aligned}
 1079 \quad \sigma' &= \sigma_1\{M/x\}[x_1 \mapsto M_1\{M/x\}, \dots, x_n \mapsto M_n\{M/x\}][x \mapsto M] \\
 1080 \quad \sigma'' &= \sigma_1\{M/x\}[x_1 \mapsto M_1\{M/x\}\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{M/x\}\{\bullet/\gamma(c)\}][x \mapsto M] \\
 1081 \quad \sigma''' &= \sigma_1[x_1 \mapsto M_1\{\bullet/\gamma(c)\}, \dots, x_n \mapsto M_n\{\bullet/\gamma(c)\}]
 \end{aligned}$$

1082 Below we repeat Proposition 20.

1083 For $N \in \Lambda$ and $t \in \Lambda_a^S$ the following properties hold

$$\begin{array}{ccc}
 \Lambda_a^S & \xrightarrow{\llbracket - \mid \sigma^w \mid \gamma \rrbracket_w} & \Lambda_w \\
 \searrow \llbracket - \mid \sigma^\Lambda \mid \gamma \rrbracket & & \swarrow \llbracket - \rrbracket \\
 & \Lambda &
 \end{array}
 \quad
 \begin{array}{ccc}
 \Lambda_a^S & \xrightarrow{\llbracket - \rrbracket^w} & \Lambda_w \\
 \searrow \llbracket - \rrbracket & & \swarrow \llbracket - \rrbracket^w \\
 & \Lambda &
 \end{array}
 \quad
 \begin{array}{ccc}
 & \Lambda_w & \\
 \swarrow \llbracket - \rrbracket^w & & \searrow \llbracket - \rrbracket \\
 \Lambda & \xrightarrow{=} & \Lambda
 \end{array}$$

$$\begin{aligned}
 & \llbracket \llbracket t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket t \mid \sigma^\Lambda \mid \gamma \rrbracket & \llbracket \llbracket N \rrbracket^w \rrbracket = \llbracket N \rrbracket^w & \llbracket \llbracket N \rrbracket^w \rrbracket = N
 \end{aligned}$$

1085 where $\sigma^\Lambda(z) = \llbracket \sigma^w(z) \rrbracket$.

1086 **Proof.** We prove $\llbracket \llbracket u \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket u \mid \sigma^\Lambda \mid \gamma \rrbracket$ by induction on u .

1088 Base Case: Variable

$$1089 \quad \llbracket \llbracket x \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket \sigma^w(x) \rrbracket = \llbracket x \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1091 Inductive Case: Application

$$1092 \quad \llbracket \llbracket st \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket \llbracket s \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket \llbracket \llbracket t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \llbracket s \mid \sigma^\Lambda \mid \gamma \rrbracket \llbracket t \mid \sigma^\Lambda \mid \gamma \rrbracket = \llbracket st \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1094 Inductive Case: Abstraction

$$1095 \quad \llbracket \llbracket x \langle x \rangle . t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \lambda x. \llbracket \llbracket t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \lambda x. \llbracket t \mid \sigma^\Lambda \mid \gamma \rrbracket = \llbracket x \langle x \rangle . t \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1097 Inductive Case: Phantom-Abstraction

$$1098 \quad \llbracket \llbracket c \langle x_1, \dots, x_n \rangle . t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \lambda c. \llbracket \llbracket t \mid \sigma_1^w \mid \gamma \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \lambda c. \llbracket x \mid \sigma_1^\Lambda \mid \gamma \rrbracket = \llbracket c \langle x_1, \dots, x_n \rangle . t \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1099 where

$$1100 \quad \sigma_1^w = \sigma[x_1 \mapsto \sigma(x_1)\{c/\gamma(c)\}, \dots, x_n \mapsto \sigma(x_n)\{c/\gamma(c)\}]$$

$$1101 \quad \sigma_1^\Lambda = \sigma[x_1 \mapsto \llbracket \sigma(x_1) \rrbracket \{c/\gamma(c)\}, \dots, x_n \mapsto \llbracket \sigma(x_n) \rrbracket \{c/\gamma(c)\}]$$

1103 Inductive Case: Weakening

$$1104 \quad \llbracket \llbracket u \langle \leftarrow t \rangle \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket \llbracket u \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma^\Lambda \mid \gamma \rrbracket = \llbracket u \langle \leftarrow t \rangle \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1106 Inductive Case: Sharing

$$1107 \quad \llbracket \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket = \llbracket \llbracket u \mid \sigma_1^w \mid \gamma \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u \mid \sigma_1^\Lambda \mid \gamma \rrbracket = \llbracket u[x_1, \dots, x_n \leftarrow t] \mid \sigma^\Lambda \mid \gamma \rrbracket$$

1108 where

$$1109 \quad \sigma_1^w = \sigma^w[x_i \mapsto \llbracket t \mid \sigma^w \mid \gamma \rrbracket_w]_{1 \leq i \leq n}$$

$$1110 \quad \sigma_1^\Lambda = \sigma^\Lambda[x_i \mapsto \llbracket \llbracket t \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket]_{1 \leq i \leq n} \stackrel{\text{I.H.}}{=} \sigma^\Lambda[x_i \mapsto \llbracket t \mid \sigma^\Lambda \mid \gamma \rrbracket]_{1 \leq i \leq n}$$

1112 Inductive Case: Distributor

$$1113 \quad \llbracket \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle \mid c \langle \vec{x} \rangle \overline{[\Gamma]}] \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket$$

1115 SubCase: $\vec{x} = c$

$$1116 \quad \llbracket \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle \mid c \langle c \rangle \overline{[\Gamma]}] \mid \sigma^w \mid \gamma \rrbracket_w \rrbracket$$

$$1117 \quad = \llbracket \llbracket u \overline{[\Gamma]} \mid \sigma \mid \gamma' \rrbracket_w \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u \overline{[\Gamma]} \mid \sigma^\Lambda \mid \gamma' \rrbracket$$

$$= \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle c \rangle \overline{[\Gamma]}] | \sigma^\Lambda | \gamma \rrbracket$$

1119

1120 SubCase: $\vec{x} = x_1, \dots, x_n$

$$\llbracket \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] | \sigma^\omega | \gamma \rrbracket_\omega \rrbracket$$

$$\llbracket \llbracket u[\overline{[\Gamma]}] | \sigma_1^\omega | \gamma' \rrbracket_\omega \rrbracket \stackrel{\text{I.H.}}{=} \llbracket u[\overline{[\Gamma]}] | \sigma_1^\Lambda | \gamma' \rrbracket$$

$$= \llbracket u[e_1 \langle \vec{w}_1 \rangle, \dots, e_m \langle \vec{w}_m \rangle | c \langle x_1, \dots, x_n \rangle \overline{[\Gamma]}] | \sigma^\Lambda | \gamma \rrbracket$$

1124 where

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

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

1127

1128 We prove $\llbracket \langle N \rangle \rrbracket^\omega = \langle N \rangle^\omega$ by induction on N . We prove this statement by first proving it for closed terms.

1130

1131 Base Case: Variable

$$\llbracket \langle x \rangle' \rrbracket^\omega = \llbracket x \rrbracket^\omega = x = \langle x \rangle^\omega$$

1133

1134 Inductive Case: Application

$$\llbracket \langle M N \rangle' \rrbracket^\omega = \llbracket \langle M \rangle' \rrbracket^\omega \llbracket \langle N \rangle' \rrbracket^\omega \stackrel{\text{I.H.}}{=} \langle M \rangle^\omega \langle N \rangle^\omega = \langle M N \rangle^\omega$$

1136

1137 Inductive Case: Abstraction

$$\llbracket \langle \lambda x. M \rangle' \rrbracket^\omega$$

1139 SubCase: $|M|_x = 0$

$$= \lambda x. \llbracket \langle M \rangle' [\leftarrow x] \rrbracket^\omega = \lambda x. \llbracket \langle M \rangle' \rrbracket^\omega [\leftarrow x] \stackrel{\text{I.H.}}{=} \lambda x. \langle M \rangle^\omega [\leftarrow x] = \langle \lambda x. M \rangle^\omega$$

1141

1142 SubCase: $|M|_x = 1$

$$= \lambda x. \llbracket \langle M \rangle' \rrbracket^\omega \stackrel{\text{I.H.}}{=} \lambda x. \langle M \rangle^\omega = \langle \lambda x. M \rangle^\omega$$

1144

1145 SubCase: $|M|_x = n > 1$

$$= \llbracket \langle M_x^n \rangle' [x^1, \dots, x^n \leftarrow x] \rrbracket^\omega = \llbracket \langle M_x^n \rangle' | \sigma | I \rrbracket_\omega \stackrel{\text{prop 44}}{=} \llbracket \langle M_x^n \rangle' \rrbracket^\omega \{x/x_i\}_{1 \leq i \leq n}$$

$$\stackrel{\text{I.H.}}{=} \langle M_x^n \rangle^\omega \{x/x_i\}_{1 \leq i \leq n} = \langle M \rangle^\omega$$

1148

1149 Now that we have proven it works for closed terms, we can show the statement $\llbracket \langle N \rangle \rrbracket^\omega = \langle N \rangle^\omega$ holds

1151

$$\llbracket \langle N \rangle \rrbracket^\omega = \llbracket \langle N \frac{n_1}{x_1} \dots \frac{n_k}{x_k} \rangle' [x_1^1, \dots, x_1^{n_1} \leftarrow x_1] \dots [x_k^1, \dots, x_k^{n_k} \leftarrow x_k] \rrbracket^\omega$$

$$\stackrel{\text{prop 44}}{=} \llbracket \langle N \frac{n_1}{x_1} \dots \frac{n_k}{x_k} \rangle' \rrbracket^\omega \{x_i/x_i^j\}_{1 \leq i \leq k, 1 \leq j \leq n_i} = \langle N \frac{n_1}{x_1} \dots \frac{n_k}{x_k} \rangle^\omega \{x_i/x_i^j\}_{1 \leq i \leq k, 1 \leq j \leq n_i} = \langle N \rangle^\omega \quad \blacktriangleleft$$

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

$$\llbracket t \rrbracket^\omega \rightarrow_\beta^+ \llbracket u \rrbracket^\omega$$

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

$$\llbracket t | \sigma | \gamma \rrbracket_\omega \rightarrow_\omega^* \llbracket u | \sigma | \gamma \rrbracket_\omega$$

1154 **Proof.** We prove this by induction. We first discuss all the case bases. $\llbracket (x \langle x \rangle. t) s \rrbracket^\omega =$

$$1155 (\lambda x. T) S = T \{S/x\} = \llbracket t \{s/x\} \rrbracket^\omega$$

where $T = \llbracket t \rrbracket^\omega$ and $S = \llbracket s \rrbracket^\omega$.

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)

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

$$\begin{aligned} \llbracket u[\leftarrow st] \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket s \mid \sigma \mid \gamma \rrbracket_w \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \\ &\rightarrow_w \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket s \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_w]] = \llbracket u[\leftarrow s][\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w \end{aligned}$$

Case: (d_2)

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

$$\llbracket u[\leftarrow c\langle \vec{x} \rangle.t] \mid \sigma \mid \gamma \rrbracket_w$$

SubCase: $\vec{x} = c$

$$\begin{aligned} \llbracket u[\leftarrow c\langle c \rangle.t] \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \lambda c. \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \rightarrow_w \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_w \{\bullet/c\}] \\ &\stackrel{\text{prop}^{44}}{=} \llbracket u[\leftarrow t] \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u[\leftarrow c\langle c \rangle][\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w \\ &\text{where } \sigma' = \sigma[c \mapsto \bullet] \end{aligned}$$

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

$$\begin{aligned} \llbracket u[\leftarrow c\langle x_1, \dots, x_n \rangle.t] \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket c\langle x_1, \dots, x_n \rangle.t \mid \sigma \mid \gamma \rrbracket_w] \\ \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \lambda c. \llbracket t \mid \sigma' \mid \gamma \rrbracket_w] &\rightarrow_w \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket t \mid \sigma' \mid \gamma \rrbracket_w \{\bullet/c\}] \\ &\stackrel{\text{prop}^{44}}{=} \llbracket u[\leftarrow t] \mid \sigma'' \mid \gamma \rrbracket_w = \llbracket u[\leftarrow c\langle x_1, \dots, x_n \rangle][\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w \end{aligned}$$

Case: (d_3)

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

$$\begin{aligned} \llbracket u[\leftarrow c\langle c \rangle][\leftarrow c] \mid \sigma \mid \gamma \rrbracket_w &= \llbracket u[\leftarrow c] \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u \mid \sigma' \mid \gamma \rrbracket_w [\leftarrow \bullet] \\ &= \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \bullet] \rightarrow_w \llbracket u \mid \sigma \mid \gamma \rrbracket_w \end{aligned}$$

Case (c_2)

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

$$\llbracket u[x \leftarrow t] \mid \sigma \mid \gamma \rrbracket_w = \llbracket u \mid \sigma' \mid \gamma \rrbracket_w = \llbracket u\{t/x\} \mid \sigma \mid \gamma \rrbracket_w$$

where

$$\sigma' = \sigma[x \mapsto \llbracket t \mid \sigma \mid \gamma \rrbracket_w]$$

For the remaining cases, we only show the cases for $\llbracket u[\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w = \llbracket u \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_w]$. The other cases are similar to those in the proof for Lemma 16.

Case: (l_1)

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

$$\begin{aligned} \llbracket s[\leftarrow t]u \mid \sigma \mid \gamma \rrbracket_w &= \llbracket s \mid \sigma \mid \gamma \rrbracket_w [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \llbracket u \mid \sigma \mid \gamma \rrbracket_w \rightarrow_w (\llbracket s \mid \sigma \mid \gamma \rrbracket_w \llbracket u \mid \sigma \mid \gamma \rrbracket_w) [\leftarrow \llbracket t \mid \sigma \mid \gamma \rrbracket_w] \\ &= \llbracket (su)[\leftarrow t] \mid \sigma \mid \gamma \rrbracket_w \end{aligned}$$

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)

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

$$\begin{aligned} \text{SubCase: } \vec{x} = d \\ \llbracket d\langle d \rangle.u[\leftarrow t]|\sigma|\gamma\rrbracket_w = \lambda d.(\llbracket u|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma\rrbracket_w]) \rightarrow_w \lambda d.\llbracket u|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma\rrbracket_w] \\ = \llbracket (d\langle \vec{x} \rangle.u)[\leftarrow t]|\sigma|\gamma\rrbracket_w \end{aligned}$$

$$\begin{aligned} \text{SubCase: } \vec{x} = x_1, \dots, x_n \\ \llbracket d\langle x_1, \dots, x_n \rangle.u[\leftarrow t]|\sigma|\gamma\rrbracket_w = \lambda d.(\llbracket u|\sigma'|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma'|\gamma\rrbracket_w]) \\ \rightarrow_w \lambda d.\llbracket u|\sigma'|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma'|\gamma\rrbracket_w] = \llbracket (d\langle x_1, \dots, x_n \rangle.u)[\leftarrow t]|\sigma|\gamma\rrbracket_w \end{aligned}$$

Case: (l_5)

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

1156 iff all $\vec{x} \notin (t)_{fv}$

1157

$$\llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle [\overline{\Gamma}][\leftarrow t]]|\sigma|\gamma\rrbracket_w$$

1159 Case: $\vec{x} = c$

$$\begin{aligned} 1160 &= \llbracket u[\overline{\Gamma}][\leftarrow t]|\sigma|\gamma'\rrbracket_w = \llbracket u[\overline{\Gamma}]|\sigma|\gamma'\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma'\rrbracket_w] \\ 1161 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\overline{\Gamma}]]|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma'\rrbracket_w] \\ 1162 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\overline{\Gamma}]]|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma\rrbracket_w] \\ 1163 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle c \rangle [\overline{\Gamma}]][\leftarrow t]|\sigma|\gamma\rrbracket_w \end{aligned}$$

1164

1165 Case: $\vec{x} = x_1, \dots, x_n$

$$\begin{aligned} 1166 &= \llbracket u[\overline{\Gamma}][\leftarrow t]|\sigma'|\gamma'\rrbracket_w = \llbracket u[\overline{\Gamma}]|\sigma'|\gamma'\rrbracket_w[\leftarrow \llbracket t|\sigma'|\gamma'\rrbracket_w] \\ 1167 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle x_1, \dots, x_n \rangle [\overline{\Gamma}]]|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma'|\gamma'\rrbracket_w] \\ 1168 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle x_1, \dots, x_n \rangle [\overline{\Gamma}]]|\sigma|\gamma\rrbracket_w[\leftarrow \llbracket t|\sigma|\gamma\rrbracket_w] \\ 1169 &= \llbracket u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle x_1, \dots, x_n \rangle [\overline{\Gamma}]][\leftarrow t]|\sigma|\gamma\rrbracket_w \end{aligned}$$

◀

1170 B.1 Sharing Measure

1171 We prove strong normalisation of sharing reductions through the use of *multisets*. Intuitively,
1172 a multiset can be interpreted as a set where elements can be repeated, or equivalently as lists
1173 that are considered equal up to the permutation of elements. We use multisets to measure
1174 aspects of a term, and show that these aspects strictly decrease via $\rightsquigarrow_{(R,D,L)}$ reduction.

1175 ► **Definition 45** (Multisets). A multiset m is a pair (A, f) where A is a set and $f : A \rightarrow \mathcal{N}$
1176 is a function that maps elements of A to a natural number.

1177 The formal definition of multisets in Definition 45 follows intuition when we consider the
1178 function f to tell us the number of occurrences of an element $x \in A$ in the multiset m .

1179 ► **Example 46.** Let $m = (\{x, y, z\}, f)$ and $f(x) = 2$, $f(y) = 1$ and $f(z) = 3$. Then this
1180 multiset can also be written as $\{x, x, y, z, z, z\}$ or equivalently as $\{x^2, y^1, z^3\}$

1181 ► **Remark 47.** The empty multiset is written as $\{\}$

1182 We will need to be able to reason about multisets in order to use them as part of our
1183 reasoning for strong normalisation. First we discuss the union of multisets, which will be
1184 needed when measuring a term recursively, e.g. in an application st we will need to measure
1185 aspects of s and unionise them with the multiset corresponding to the measure of the same
1186 of t , to obtain the overall measure of the application.

1187 ► **Definition 48** (Union of Multisets). The union (or sum) of two multisets $m = (A, f)$ and
1188 $n = (B, g)$ is the multiset $m \cup n = (A \cup B, h)$ such that for all $x \in A \cup B$, $h(x) = f(x) + g(x)$.

1189 ► **Example 49.** Let $m = \{a^1, b^3, c^2\}$ and $n = \{c^3, d^1\}$, then $m \cup n = \{a^1, b^3, c^5, d^1\}$

1190 ► **Remark 50.** The notion $A \cup B$ is the union of the sets and *not* a disjoint union.

1191 To show strong normalisation of sharing reductions, we need to show that aspects of
 1192 terms that can be represented as multisets strictly decrease during reduction. In order to
 1193 show this, we need to be able determine when a multiset is larger/smaller than another i.e.
 1194 we need to be able to apply an ordering.

1195 ► **Definition 51 (Ordering of Multisets).** *Given a totally ordered set A and two multisets*
 1196 *$m = (A, f)$ and $n = (A, g)$, we say m is strictly larger than n , $m > n$, if the following*
 1197 *conditions hold*

1198 • $m \neq n$

1199 • $\forall x \in A. (g(x) > f(x) \rightarrow \exists y \in A. [(y > x) \wedge (f(y) > g(y))])$
 1200

1201 ► **Example 52.** $\{1^5, 2^2, 3^1\} < \{1^3, 2^4, 3^3\}$

1202 The *height* of a term is intuitively a multiset of integers that record the scope of each
 1203 sharing. The scope is measured by the number of constructors from the sharing node to the
 1204 root of the term in its graphical notation. The formal definition of the height is given in
 1205 Definition 32. First we prove Lemma 27 on a case-by-case basis.

1206 If $t \rightsquigarrow_{(L)} u$ then $\mathcal{H}^i(t) > \mathcal{H}^i(u)$

Proof.

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

$$\begin{aligned} \mathcal{H}^i((s[\Gamma]) t) &= \mathcal{H}^{i+1}(s[\Gamma]) \cup \mathcal{H}^{i+1}(t) = \mathcal{H}^{i+1}(s) \cup \mathcal{H}^{i+1}(t) \cup \mathcal{H}^{i+1}([\Gamma]) \cup \{i+1\} \\ \mathcal{H}^i((st)[\Gamma]) &= \mathcal{H}^i(st) \cup \mathcal{H}^i([\Gamma]) = \mathcal{H}^{i+1}(s) \cup \mathcal{H}^{i+1}(t) \cup \mathcal{H}^i([\Gamma]) \cup \{i\} \end{aligned}$$

$$st[\Gamma] \rightsquigarrow_L (st)[\Gamma]$$

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

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

$$\begin{aligned} \mathcal{H}^i(c\langle \vec{x} \rangle. t[\Gamma]) &= \mathcal{H}^{i+1}(t[\Gamma]) = \mathcal{H}^{i+1}(t) \cup \mathcal{H}^{i+1}([\Gamma]) \cup \{i+1\} \\ \mathcal{H}^i((c\langle \vec{x} \rangle. t)[\Gamma]) &= \mathcal{H}^i(c\langle \vec{x} \rangle. t) \cup \mathcal{H}^i([\Gamma]) \cup \{i\} = \mathcal{H}^{i+1}(t) \cup \mathcal{H}^i([\Gamma]) \cup \{i\} \end{aligned}$$

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

$$\begin{aligned} \mathcal{H}^i(u[\vec{x} \leftarrow t[\Gamma]]) &= \mathcal{H}^i(u) \cup \mathcal{H}^i([\vec{x} \leftarrow t[\Gamma]]) \cup \{i\} = \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(t) \cup \mathcal{H}^{i+1}([\Gamma]) \cup \{i, i+1\} \\ \mathcal{H}^i(u[\vec{x} \leftarrow t][\Gamma]) &= \mathcal{H}^i(u[\vec{x} \leftarrow t]) \cup \mathcal{H}^i([\Gamma]) \cup \{i\} = \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(t) \cup \mathcal{H}^{i+1}([\Gamma]) \cup \{i, i\} \end{aligned}$$

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

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

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

$$\begin{aligned} &\mathcal{H}^i(u[e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\vec{y} \leftarrow t]]) \\ &= \mathcal{H}^i(u) \cup \mathcal{H}^i([e_1\langle \vec{w}_1 \rangle \dots e_n\langle \vec{w}_n \rangle | c\langle \vec{x} \rangle \overline{[\Gamma]}[\vec{y} \leftarrow t]]) \cup \{i\} \\ &= \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(\overline{[\Gamma]}) \cup \mathcal{H}^{i+1}([\vec{y} \leftarrow t]) \cup \{i, (i+1)^{n+1}\} \end{aligned}$$

where n is the number of closures in the environment $\overline{[\Gamma]}$

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

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

$$\begin{aligned}
&u[e_1\langle \bar{w}_1 \rangle \dots e_n\langle \bar{w}_n \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}] \overrightarrow{[f\langle \bar{z} \rangle | d\langle \bar{a} \rangle \overline{[\Gamma']}]} \rightsquigarrow_L \\
&u\{\langle \bar{w}_1/\bar{z} \rangle / e_1\}_b \dots \{\langle \bar{w}_n/\bar{z} \rangle / e_n\}_b [e_1\langle \bar{w}_1/\bar{z} \rangle \dots e_n\langle \bar{w}_n/\bar{z} \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}] \overrightarrow{[f\langle \bar{z} \rangle | d\langle \bar{a} \rangle \overline{[\Gamma']}]} \\
1207 &\text{iff all } \bar{x} \in (u[e_1\langle \bar{w}_1 \rangle \dots e_n\langle \bar{w}_n \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}])_{fv} \\
1208 &\mathcal{H}^i(u[e_1\langle \bar{w}_1 \rangle \dots e_n\langle \bar{w}_n \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}] \overrightarrow{[f\langle \bar{z} \rangle | d\langle \bar{a} \rangle \overline{[\Gamma']}]}]) \\
1209 &= \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(\overline{[\Gamma]}) \cup \mathcal{H}^{i+1}(\overrightarrow{[f\langle \bar{z} \rangle | d\langle \bar{a} \rangle \overline{[\Gamma']}]}]) \cup \{i, (i+1)^{n+1}\} \\
1210 &\text{where } n \text{ is the number of closures in } \overline{[\Gamma]} \\
1211 &= \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(\overline{[\Gamma]}) \cup \mathcal{H}^{i+2}(\overline{[\Gamma']}) \cup \{i, (i+1)^{n+1}, (i+2)^m\} \\
1212 &\text{where } m \text{ is the number of closures in } \overline{[\Gamma']} \\
1213 &\mathcal{H}^i(u\{\langle \bar{w}_1/\bar{z} \rangle / e_1\}_b \dots \{\langle \bar{w}_n/\bar{z} \rangle / e_n\}_b [e_1\langle \bar{w}_1/\bar{z} \rangle \dots e_n\langle \bar{w}_n/\bar{z} \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}] \overrightarrow{[f\langle \bar{z} \rangle | d\langle \bar{a} \rangle \overline{[\Gamma']}]}]) \\
1214 &\mathcal{H}^i(u\{\langle \bar{w}_1/\bar{z} \rangle / e_1\}_b \dots \{\langle \bar{w}_n/\bar{z} \rangle / e_n\}_b [e_1\langle \bar{w}_1/\bar{z} \rangle \dots e_n\langle \bar{w}_n/\bar{z} \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}]) \\
1215 &\cup \mathcal{H}^{i+1}(\overline{[\Gamma']}) \cup \{i, (i+1)^m\} \\
1216 &= \mathcal{H}^i(u\{\langle \bar{w}_1/\bar{z} \rangle / e_1\}_b \dots \{\langle \bar{w}_n/\bar{z} \rangle / e_n\}_b) \cup \mathcal{H}^{i+1}(\overline{[\Gamma]}) \cup \mathcal{H}^{i+1}(\overline{[\Gamma']}) \cup \{i, (i+1)^{n+m}\} \\
1217 &= \mathcal{H}^i(u) \cup \mathcal{H}^{i+1}(\overline{[\Gamma]}) \cup \mathcal{H}^{i+1}(\overline{[\Gamma']}) \cup \{i, (i+1)^{n+m}\} \quad \blacktriangleleft
\end{aligned}$$

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

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

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

1218 and its equivalent in the $\Lambda_{\mathcal{W}}$ -calculus obtained by $\llbracket - \rrbracket^{\mathcal{W}}$. Each red line shows the connection
 1219 between the abstraction and application constructors in both calculi. The weight of a
 1220 constructor is then the number of red lines associated with it, e.g. the weight of the example
 1221 is the multiset $\{1^6, 2^4, 4^1\}$.

1222 ► **Proposition 53.** For $e \notin \bar{w}$, $\mathcal{W}^i(t) = \mathcal{W}^i(t\{\bar{w}/e\}_b)$

1223 **Proof.** To prove this, first we need to prove that book-keeping does not affect the function
 1224 $\mathcal{V}^i(t)$. We prove this by induction on t .

1225 Base Case: Variable

1226 Vacuously True

1227

1228 Base Case: Abstraction

$$1229 \mathcal{V}^i(e\langle \bar{y} \rangle . t\{\bar{w}/e\}_b) = \mathcal{V}^i(e\langle \bar{w} \rangle . t) = \mathcal{V}^i(t) \cup \{e \mapsto i\} = \mathcal{V}^i(e\langle \bar{y} \rangle . t)$$

1230

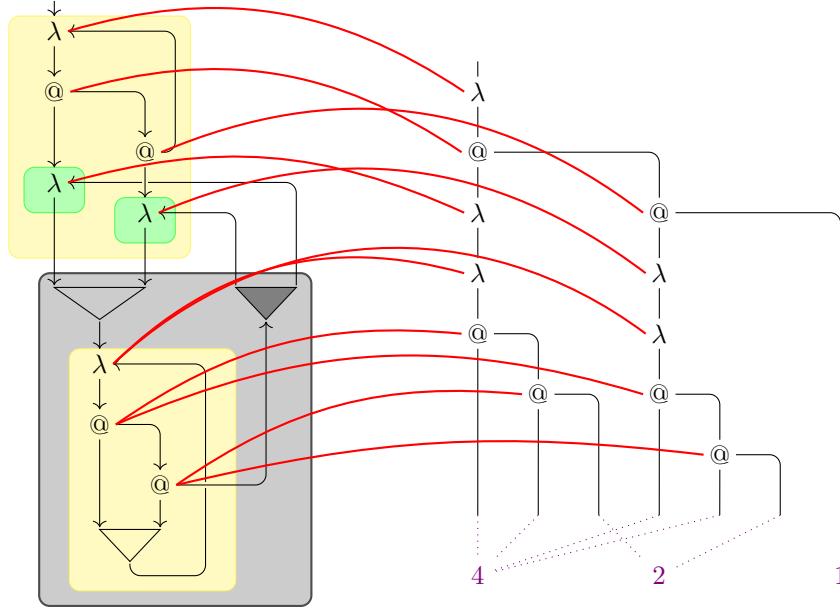
1231 Base Case: Distributor

$$\begin{aligned}
1232 &\mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} | e\langle \bar{y} \rangle \overline{[\Gamma]}]\{\bar{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} | e\langle \bar{w} \rangle \overline{[\Gamma]}]) \\
1233 &= \mathcal{V}^i(u\overline{[\Gamma]}) \cup \{\bar{e}\} = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} | e\langle \bar{y} \rangle \overline{[\Gamma]}])
\end{aligned}$$

1234

1235 Inductive Case: Application

$$\begin{aligned}
1236 &\mathcal{V}^i(st\{\bar{w}/e\}_b) = \mathcal{V}^i((s\{\bar{w}/e\}_b)t\{\bar{w}/e\}_b) = \mathcal{V}^i(s\{\bar{w}/e\}_b) \cup \mathcal{V}^i(t\{\bar{w}/e\}_b) \stackrel{1.H.2}{=} \mathcal{V}^i(s) \cup \mathcal{V}^i(t) = \\
1237 &\mathcal{V}^i(st)
\end{aligned}$$



■ **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 $\{1^6, 2^4, 4\}$

1238

1239 Inductive Case: Abstraction

1240 Case 1

$$1241 \mathcal{V}^i((c\langle c \rangle.t)\{\bar{w}/e\}_b) = \mathcal{V}^i(c\langle c \rangle.t\{\bar{w}/e\}_b) = \mathcal{V}^i(t\{\bar{w}/e\}_b)/\{c\} \stackrel{\text{I.H.}}{=} \mathcal{V}^i(t)/\{c\} = \mathcal{V}^i(c\langle c \rangle.t)$$

1242 Case 2

$$1243 \mathcal{V}^i((c\langle \bar{x} \rangle.t)\{\bar{w}/e\}_b) = \mathcal{V}^i(c\langle \bar{x} \rangle.t\{\bar{w}/e\}_b) = \mathcal{V}^i(t\{\bar{w}/e\}_b) \cup \{c \mapsto i\} \stackrel{\text{I.H.}}{=} \mathcal{V}^i(t) \cup \{c \mapsto i\} =$$

$$1244 \mathcal{V}^i(c\langle \bar{x} \rangle.t)$$

1245

1246 Inductive Case: Weakening

$$1247 \mathcal{V}^i(u[\leftarrow t]\{\bar{w}/e\}_b) = \mathcal{V}^i(u\{\bar{w}/e\}_b[\leftarrow t\{\bar{w}/e\}_b]) = \mathcal{V}^i(u\{\bar{w}/e\}_b) \cup \mathcal{V}^1(t\{\bar{w}/e\}_b)$$

$$1248 \stackrel{\text{I.H.}^2}{=} \mathcal{V}^i(u) \cup \mathcal{V}^1(t) = \mathcal{V}^i(u[\leftarrow t])$$

1249

1250 Inductive Case: Sharing

$$1251 \mathcal{V}^i(u[x_1 \dots x_n \leftarrow t]\{\bar{w}/e\}_b) = \mathcal{V}^i(u\{\bar{w}/e\}_b[x_1 \dots x_n \leftarrow t\{\bar{w}/e\}_b])$$

$$1252 = (\mathcal{V}^i(u\{\bar{w}/e\}_b)/\{x_1, \dots, x_n\}) \cup \mathcal{V}(tj\{\bar{w}/e\}_b) \text{ where } j = \mathcal{V}^i(t\{\bar{w}/e\}_b) + \dots + \mathcal{V}^i(t\{\bar{w}/e\}_b)$$

$$1253 \stackrel{\text{I.H.}^{n+2}}{=} (\mathcal{V}^i(u)/\{x_1, \dots, x_n\}) \cup \mathcal{V}(t) \text{ where } j = \mathcal{V}^i(t) + \dots + \mathcal{V}^i(t) = \mathcal{V}^i(u[x_1, \dots, x_n \leftarrow t])$$

1254

1255 Inductive Case: Distributor

1256 Case 1

$$1257 \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\overline{\Gamma}]]\{\bar{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\overline{\Gamma}]\{\bar{w}/e\}_b]) = \mathcal{V}^i(u[\overline{\Gamma}]\{\bar{w}/e\}_b)/\{c, \vec{f}\}$$

$$1258 \stackrel{\text{I.H.}}{=} \mathcal{V}^i(u[\overline{\Gamma}])/\{c, \vec{f}\} = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\overline{\Gamma}]])$$

1259 Case 2

$$1260 \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]]\{\bar{w}/e\}_b) = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]\{\bar{w}/e\}_b])$$

$$1261 = \mathcal{V}^i(u[\overline{\Gamma}]\{\bar{w}/e\}_b)/\{\vec{f}\} \cup \{c \mapsto i\}$$

$$1262 \stackrel{\text{I.H.}}{=} \mathcal{V}^i(u[\overline{\Gamma}])/\{\vec{f}\} \cup \{c \mapsto i\} = \mathcal{V}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]])$$

1263

 1264 We now prove this proposition by induction on t

1265

Base Case: Variable

1266

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

1267

Base Case: Abstraction

1268

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

1270

Base Case: Distributor

1271

$$\begin{aligned} \mathcal{W}^i(u[\overrightarrow{e\langle \bar{z} \rangle} \mid e\langle \bar{y} \rangle [\Gamma]]\{\bar{w}/e\}_b) &= \mathcal{W}^i(u[\overrightarrow{e\langle \bar{z} \rangle} \mid e\langle \bar{w} \rangle [\Gamma]]) = \mathcal{W}^i(u[\overline{\Gamma}]) \\ &= \mathcal{W}^i(u[\overrightarrow{e\langle \bar{z} \rangle} \mid e\langle \bar{y} \rangle [\Gamma]]) \end{aligned}$$

1274

Inductive Case: Application

1275

$$\begin{aligned} \mathcal{W}^i(st\{\bar{w}/e\}_b) &= \mathcal{W}^i((s\{\bar{w}/e\}_b)t\{\bar{w}/e\}_b) = \mathcal{W}^i(s\{\bar{w}/e\}_b) \cup \mathcal{W}^i(t\{\bar{w}/e\}_b) \cup \{i\} \\ &\stackrel{\text{I.H.}^2}{=} \mathcal{W}^i(s) \cup \mathcal{W}^i(t) \cup \{i\} = \mathcal{W}^i(st) \end{aligned}$$

1278

Inductive Case: Abstraction

1280

Case 1

1281

$$\begin{aligned} \mathcal{W}^i((c\langle c \rangle.t)\{\bar{w}/e\}_b) &= \mathcal{W}^i(c\langle c \rangle.t\{\bar{w}/e\}_b) = \mathcal{W}^i(t\{\bar{w}/e\}_b) \cup \{i, \mathcal{V}^i(t\{\bar{w}/e\}_b)(c)\} \\ &\stackrel{\text{I.H.}}{=} \mathcal{W}^i(t) \cup \{i, \mathcal{V}^i(t)(c)\} = \mathcal{W}^i(c\langle c \rangle.t) \end{aligned}$$

1283

Case 2

1284

$$\begin{aligned} \mathcal{W}^i((c\langle \bar{x} \rangle.t)\{\bar{w}/e\}_b) &= \mathcal{W}^i(c\langle \bar{x} \rangle.t\{\bar{w}/e\}_b) = \mathcal{W}^i(t\{\bar{w}/e\}_b) \cup \{i\} \stackrel{\text{I.H.}}{=} \mathcal{W}^i(t) \cup \{i\} \\ &= \mathcal{W}^i(c\langle \bar{x} \rangle.t) \end{aligned}$$

1286

Inductive Case: Weakening

1287

$$\begin{aligned} \mathcal{W}^i(u[\leftarrow t]\{\bar{w}/e\}_b) &= \mathcal{W}^i(u\{\bar{w}/e\}_b[\leftarrow t\{\bar{w}/e\}_b]) = \mathcal{W}^i(u\{\bar{w}/e\}_b) \cup \mathcal{W}^1(t\{\bar{w}/e\}_b) \\ &\stackrel{\text{I.H.}^2}{=} \mathcal{W}^i(u) \cup \mathcal{W}^1(t) = \mathcal{W}^i(u[\leftarrow t]) \end{aligned}$$

1290

Inductive Case: Sharing

1291

$$\begin{aligned} \mathcal{W}^i(u[x_1, \dots, x_n \leftarrow t]\{\bar{w}/e\}_b) &= \mathcal{W}^i(u\{\bar{w}/e\}_b[x_1, \dots, x_n \leftarrow t\{\bar{w}/e\}_b]) \\ &= \mathcal{W}^i(u\{\bar{w}/e\}_b) \cup \mathcal{W}^j(t\{\bar{w}/e\}_b) \text{ where } j = \mathcal{V}^i(u\{\bar{w}/e\}_b)(x_1) + \dots + \mathcal{V}^i(u\{\bar{w}/e\}_b)(x_n) \\ &\stackrel{\text{I.H.}^{n+2}}{=} \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \text{ where } j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) = \mathcal{W}^i(u[x_1, \dots, x_n \leftarrow t]) \end{aligned}$$

1295

Inductive Case: Distributor

1296

Case 1

1297

$$\begin{aligned} \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\Gamma]]\{\bar{w}/e\}_b) &= \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\overline{\Gamma}]]\{\bar{w}/e\}_b) \\ &= \mathcal{W}^i(u[\overline{\Gamma}]\{\bar{w}/e\}_b) \cup \{\mathcal{V}^i(u[\overline{\Gamma}]\{\bar{w}/e\}_b)(c)\} \stackrel{\text{I.H.}}{=} \mathcal{W}^i(u[\overline{\Gamma}]) \cup \{\mathcal{V}^i(u[\overline{\Gamma}])\}(c)\} \\ &= \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle c \rangle [\overline{\Gamma}]]) \end{aligned}$$

1301

Case 2

1302

$$\begin{aligned} \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]]\{\bar{w}/e\}_b) &= \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]]\{\bar{w}/e\}_b) \\ &= \mathcal{W}^i(u[\overline{\Gamma}]\{\bar{w}/e\}_b) \stackrel{\text{I.H.}}{=} \mathcal{W}^i(u[\overline{\Gamma}]) = \mathcal{W}^i(u[\overrightarrow{f\langle \bar{z} \rangle} \mid c\langle \bar{x} \rangle [\overline{\Gamma}]]) \end{aligned}$$

1303

1304

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

1305

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

1306

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

Proof. Duplication Rules

$$\begin{aligned}
& u^*[x_1 \dots x_n \leftarrow s t] \rightsquigarrow_D u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t] \\
& \mathcal{W}^i(u^*[x_1 \dots x_n \leftarrow s t]) = \mathcal{W}^i(u) \cup \mathcal{W}^j(st) = \mathcal{W}^i(u) \cup \mathcal{W}^j(s) \cup \mathcal{W}^j(t) \cup \{j\} \\
& \text{where } j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) \\
& \mathcal{W}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s][y_1 \dots y_n \leftarrow t]) \\
& = \mathcal{W}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s]) \cup \mathcal{W}^k(t) \\
& = \mathcal{W}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}) \cup \mathcal{W}^l(s) \cup \mathcal{W}^k(t) \\
& \text{where } k = \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s])(y_1) + \dots \\
& \quad \dots + \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}[z_1 \dots z_n \leftarrow s])(y_n) \\
& = \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\})(y_1) + \dots + \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\})(y_n) \\
& = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) = j \\
& \text{and where } l = \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\})(z_1) + \dots \\
& \quad \dots + \mathcal{V}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\})(z_n) \\
& = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) = j \\
& \text{Therefore} \\
& = \mathcal{W}^i(u^*\{z_1 y_1/x_1\} \dots \{z_n y_n/x_n\}) \cup \mathcal{W}^j(s) \cup \mathcal{W}^j(t) \\
& = \mathcal{W}^i(u) \cup \mathcal{W}^j(s) \cup \mathcal{W}^j(t) \cup \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n)\}
\end{aligned}$$

$$u[x_1, \dots, x_n \leftarrow c(\vec{y}).t] \rightsquigarrow_D$$

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

Case 1:

$$\begin{aligned}
& \mathcal{W}^i(u[x_1, \dots, x_n \leftarrow c(\vec{y}).t]) = \mathcal{W}^i(u) \cup \mathcal{W}^j(c(\vec{y}).t) = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup \{j, \mathcal{V}^j(t)(c)\} \\
& \text{where } j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) \\
& \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle w_1^1 \rangle \dots e_n\langle w_1^n \rangle | c(\vec{y})[w_1^1, \dots, w_1^n \leftarrow t]]) \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t]) \cup \\
& \quad \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t])(c) \\
& \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t])(c) = \mathcal{V}^k(t)(c) = \mathcal{V}^j(t)(c) \\
& \text{where } k = \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t])(w_1^1) + \dots \\
& \quad \dots + \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t])(w_1^n) = j \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t]) \cup \mathcal{V}^j(t)(c) \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}) \cup \mathcal{W}^k(t) \cup \{\mathcal{V}^j(t)(c)\} \\
& = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n), \mathcal{V}^j(t)(c)\}
\end{aligned}$$

Case: 2

$$\begin{aligned}
& \mathcal{W}^i(u[x_1, \dots, x_n \leftarrow c(\vec{y}).t]) = \mathcal{W}^i(u) \cup \mathcal{W}^j(c(\vec{y}).t) = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup \{j\} \\
& \text{where } j = \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) \\
& \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[e_1\langle w_1^1 \rangle \dots e_n\langle w_1^n \rangle | c(\vec{y})[w_1^1, \dots, w_1^n \leftarrow t]]) \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}[w_1^1, \dots, w_1^n \leftarrow t]) \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}) \cup \mathcal{W}^k(t) \\
& \text{where } k = \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n})(w_1^1) + \dots + \mathcal{V}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n})(w_1^n) = j \\
& = \mathcal{W}^i(u\{e_i\langle w_1^i \rangle.w_1^i/x_i\}_{1 \leq i \leq n}) \cup \mathcal{W}^j(t) \\
& = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup \{\mathcal{V}^i(u)(x_1), \dots, \mathcal{V}^i(u)(x_n)\}
\end{aligned}$$

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

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

where $j = \mathcal{V}^i(u)(\vec{w}_1) + \dots + \mathcal{V}^i(u)(\vec{w}_n)$
 $\mathcal{W}^i(u\{e_1\langle \vec{w}_1 \rangle\}_e \dots \{e_n\langle \vec{w}_n \rangle\}_e) = \mathcal{W}^i(u) \cup \{\mathcal{V}^i(u)(\vec{w}_1), \dots, \mathcal{V}^i(u)(\vec{w}_n)\}$
 where $\mathcal{V}^i(u)(\vec{w}) = \mathcal{V}^i(u)(w_1) + \dots + \mathcal{V}^i(u)(w_n)$ and $\vec{w} = \{w_1, \dots, w_n\}$

Lifting and Compound

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

$$\begin{aligned} \mathcal{W}^i(u[\vec{w} \leftarrow y][\vec{x} \cdot y \leftarrow t]) &= \mathcal{W}^i(u[\vec{w} \leftarrow y]) \cup \mathcal{W}^j(t) \\ \text{where } j &= \mathcal{V}^i(u[\vec{w} \leftarrow y])(\vec{x}) + \mathcal{V}^i(u[\vec{w} \leftarrow y])(y) = \mathcal{V}^i(u[\vec{w} \leftarrow y])(\vec{x}) + \mathcal{V}^i(u)(\vec{w}) \\ &= \mathcal{W}^i(u) \cup \mathcal{W}^j(t) = \mathcal{W}^i(u[\vec{x} \cdot \vec{w} \leftarrow t]) \end{aligned}$$

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

$$1307 \quad \mathcal{W}^i(u[x \leftarrow t]) = \mathcal{W}^i(u) \cup \mathcal{W}^j(t)$$

$$1308 \quad \text{where } j = \mathcal{V}^i(u)(x)$$

$$1309 \quad \mathcal{W}^i(u\{t/x\}) = \mathcal{W}^i(u) \cup \mathcal{W}^{\mathcal{V}^i(u)(x)}(t)$$

1310

1311 For the other lifting rules, we show that $\mathcal{V}^i(u[\Gamma])$ outputs the same integers before and after
 1312 lifting for each variable bounded by $[\Gamma]$. Then we can know it produces some multiset M .

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

$$\begin{aligned} \mathcal{W}^i((s[\Gamma])t) &= \mathcal{W}^i(s[\Gamma]) \cup \mathcal{W}^i(t) = \mathcal{W}^i(s) \cup \mathcal{W}^i(t) \cup M_1 \\ \mathcal{W}^i((st)[\Gamma]) &= \mathcal{W}^i(st) \cup M_2 = \mathcal{W}^i(s) \cup \mathcal{W}^i(t) \cup M_2 \\ M_1 &= M_2 \text{ since } \mathcal{V}^i(s)(x) = \mathcal{V}^i(st)(x) \text{ for } x \in (s)_{fv} \text{ and } [\Gamma] \text{ only binds variables in } s. \end{aligned}$$

$$st[\Gamma] \rightsquigarrow_L (st)[\Gamma]$$

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:

$$\begin{aligned} \mathcal{W}^i(d\langle d \rangle.(t[\Gamma])) &= \mathcal{W}^i(t[\Gamma]) \cup \{i, \mathcal{V}^i(t[\Gamma])(d)\} = \mathcal{W}^i(t) \cup M_1 \cup \{i, \mathcal{V}^i(t)(d)\} \\ \mathcal{W}^i((d\langle d \rangle.t)[\Gamma]) &= \mathcal{W}^i(d\langle d \rangle.t) \cup M_2 = \mathcal{W}^i(t) \cup M_2 \cup \{i, \mathcal{V}^i(t)(d)\} \\ M_1 &= M_2 \text{ since } \mathcal{V}^i(t)(x) = \mathcal{W}^i(d\langle d \rangle.t)(x) \text{ where } x \neq d \text{ and } d \text{ is not bound by } [\Gamma] \end{aligned}$$

Case 2:

$$\begin{aligned} \mathcal{W}^i(d\langle \vec{x} \rangle.(t[\sigma])) &= \mathcal{W}^i(t[\sigma]) \cup \{i\} = \mathcal{W}^i(t) \cup M_1 \cup \{i\} \\ \mathcal{W}^i((d\langle \vec{x} \rangle.t)[\sigma]) &= \mathcal{W}^i(d\langle \vec{x} \rangle.t) \cup M_2 = \mathcal{W}^i(t) \cup M_2 \cup \{i\} \\ M_1 &= M_2 \text{ since } \mathcal{V}^i(t)(x) = \mathcal{W}^i(d\langle \vec{x} \rangle.t)(x) \text{ where } x \neq d \text{ and } d \text{ is not bound by } [\Gamma] \end{aligned}$$

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

Case 1:

$$\begin{aligned} \mathcal{W}^i(u[\vec{x} \leftarrow t[\Gamma]]) &= \mathcal{W}^i(u) \cup \mathcal{W}^j(t[\Gamma]) = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup M_1 \\ \text{where } j &= \mathcal{V}^i(u)(x_1) + \dots + \mathcal{V}^i(u)(x_n) \\ \mathcal{W}^i(u[\vec{x} \leftarrow t][\Gamma]) &= \mathcal{W}^i(u[\vec{x} \leftarrow t]) \cup M_2 = \mathcal{W}^i(u) \cup \mathcal{W}^j(t) \cup M_2 \\ M_1 &= M_2 \text{ since } \mathcal{V}^j(t)(x) = \mathcal{V}^i(u[\vec{x} \leftarrow t])(x) \text{ for } x \in (t)_{fv} \text{ and } [\Gamma] \text{ only binds variables in } t \end{aligned}$$

Case 2:

$$\begin{aligned} \mathcal{W}^i(u[\leftarrow t[\Gamma]]) &= \mathcal{W}^i(u) \cup \mathcal{W}^1(t[\Gamma]) = \mathcal{W}^i(u) \cup \mathcal{W}^1(t) \cup M_1 \\ \mathcal{W}^i(u[\leftarrow t][\Gamma]) &= \mathcal{W}^i(u[\leftarrow t]) \cup M_2 = \mathcal{W}^i(u) \cup \mathcal{W}^1(t) \cup M_2 \end{aligned}$$

$M_1 = M_2$ 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

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

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

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.

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

1313 Case 1:

$$\begin{aligned} 1314 \quad \mathcal{W}^i(u[e_1\langle \bar{w}_1 \rangle \dots e_n\langle \bar{w}_n \rangle | c\langle c \rangle \overline{[\Gamma]}[\Gamma]]) &= \mathcal{W}^i(u\overline{[\Gamma]}[\Gamma]) \cup \{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma](c))\} \\ 1315 \quad &= \mathcal{W}^i(u\overline{[\Gamma]}) \cup M_1 \cup \{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma](c))\} \\ 1316 \quad \mathcal{W}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}][\Gamma]) &= \mathcal{W}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}]) \cup M_2 \\ 1317 \quad &= \mathcal{W}^i(u'\overline{[\Gamma]}) \cup M_2 \cup \{\mathcal{V}^i(u'\overline{[\Gamma]})(c)\} \\ 1318 \quad M_1 = M_2 \text{ since } \mathcal{V}^i(u\overline{[\Gamma]})(x) &= \mathcal{V}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}])(x) \\ 1319 \quad \text{for } x \in (u\overline{[\Gamma]})/\{c, e_1, \dots, e_n\}_{fv} \text{ and the variables } c, e_1, \dots, e_n &\text{ are not bound by } [\Gamma] \\ 1320 \quad \{\mathcal{V}^i(u\overline{[\Gamma]}[\Gamma])(c)\} = \{\mathcal{V}^i(u\overline{[\Gamma]})(c)\} \text{ since } c \in (\overline{[\Gamma]})_{fv} \text{ and } \mathcal{V}^i(u\overline{[\Gamma]}[\Gamma]) &= \mathcal{V}^i(u\overline{[\Gamma]}) \cup \mathcal{V}^j([\Gamma]). \end{aligned}$$

1321 Case 2:

$$\begin{aligned} 1322 \quad \mathcal{W}^i(u[e_1\langle \bar{w}_1 \rangle \dots e_n\langle \bar{w}_n \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}[\Gamma]]) &= \mathcal{W}^i(u\overline{[\Gamma]}[\Gamma]) = \mathcal{W}^i(u\overline{[\Gamma]}) \cup M_1 \\ 1323 \quad \mathcal{W}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}][\Gamma]) &= \mathcal{W}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle \bar{x} \rangle \overline{[\Gamma]}]) \cup M_2 \\ 1324 \quad &= \mathcal{W}^i(u'\overline{[\Gamma]}) \cup M_2 \\ 1325 \quad M_1 = M_2 \text{ since } \mathcal{V}^i(u\overline{[\Gamma]})(x) &= \mathcal{V}^i(u'[e_1\langle \bar{z}_1 \rangle \dots e_n\langle \bar{z}_1 \rangle | c\langle c \rangle \overline{[\Gamma]}])(x) \\ 1326 \quad \text{for } x \in (u\overline{[\Gamma]})/\{c, e_1, \dots, e_n\}_{fv} \text{ and the variables } c, e_1, \dots, e_n &\text{ are not bound by } [\Gamma] \end{aligned}$$