Destination calculus

A linear λ−calculus for pure, functional memory updates

ARNAUD SPIWACK, Modus Create, France
THOMAS BAGREL, LORIA/Inria, France and Modus Create, France

We present the destination calculus, a linear λ−calculus for pure, functional memory updates. We introduce the syntax, type system, and operational semantics of the destination calculus, and prove type safety formally in the Coq proof assistant.

We show how the principles of the destination calculus can form a theoretical ground for destination-passing style programming in functional languages. In particular, we detail how the present work can be applied to Linear Haskell to lift the main restriction of DPS programming in Haskell as developed in [1]. We illustrate this with a range of pseudo-Haskell examples.

ACM Reference Format:
Arnaud Spiwack and Thomas Bagrel. 2024. Destination calculus: A linear λ−calculus for pure, functional memory updates. In . ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Contents

POPL'25, January 19 – 25, 2025, Denver, Colorado

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

2 SYSTEM IN ACTION ON SIMPLE EXAMPLES

Build up to DList.

3 LIMITIONS OF THE PREVIOUS APPROACH

3.1 Breadth-first tree traversal

3.2 Storing linear data in destination-based data structures

3.3 Need for scope control

4 UPDATED BREADTH-FIRST TREE TRAVERSAL

5 LANGUAGE SYNTAX

5.1 Names and variables

var, x, y, d, un, ex, st Variable names

$\mathsf{hvar}, \mathsf{h}$	$::=$			Hole (or destination) name, represented by a natural number)
	$\mid$	$\mathsf{h}{+}\mathsf{h}'$	M	
	$\mid$	$\mathsf{h}[\mathsf{H}{\pm}\mathsf{h}']$	M	Shift by h' if $\mathsf{h} \in \mathsf{H}$
	$\mid$	$\mathit{max}(\mathsf{H})$	M	Maximum of a set of hole names

$\mathsf{hvars}, \mathsf{H}$	$::=$			Set of hole names
	$\mid$	$\{\mathsf{h}_1, .., \mathsf{h}_k\}$		
	$\mid$	$\mathsf{H}_1 \cup \mathsf{H}_2$	M	Union of sets
	$\mid$	$\mathsf{H}{\pm}\mathsf{h}'$	M	Shift all names from H by h'.
	$\mid$	$\mathit{hvars}(\Gamma)$	M	Hole names bound by the typing context Γ
	$\mid$	$\mathit{hvars}(C)$	M	Hole names bound by the evaluation context C

5.2 Term and value core syntax

term, t, u	$::=$			Term
	$\mid$	v		Value
	$\mid$	x		Variable
	$\mid$	$t \triangleright t'$		Application
	$\mid$	$t \mathbin{;} u$		Pattern-match on unit
	$\mid$	$t \triangleright \mathtt{case}_{\mathsf{m}} \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1, \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}$		Pattern-match on sum
	$\mid$	$t \triangleright \mathtt{case}_{\mathsf{m}} (\mathsf{x}_1, \mathsf{x}_2) \mapsto u$		Pattern-match on product
	$\mid$	$t \triangleright \mathtt{case}_{\mathsf{m}}\ \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u$		Pattern-match on exponential
	$\mid$	$t \triangleright \mathtt{map}\ \mathsf{x} \mapsto t'$		Map over the right side of ampar
	$\mid$	$\mathtt{to}_{\ltimes}\ u$		Wrap into a trivial ampar
	$\mid$	$\mathtt{from}_{\ltimes}\ t$		Convert ampar to a pair
	$\mid$	$t \triangleleft ()$		Fill destination with unit
	$\mid$	$t \triangleleft \mathtt{Inl}$		Fill destination with left variant
	$\mid$	$t \triangleleft \mathtt{Inr}$		Fill destination with right variant
	$\mid$	$t \triangleleft \mathsf{E}_{\mathsf{m}}$		Fill destination with exponential const
	$\mid$	$t \triangleleft (,)$		Fill destination with product construct
	$\mid$	$t \triangleleft (\lambda \mathsf{x}\ {}_{\mathsf{m}} \mapsto u)$		Fill destination with function
	$\mid$	$t \triangleleft\!\bullet\ t'$		Fill destination with root of other amp
	$\mid$	$t[\mathsf{x} \coloneqq v]$	M	

val, v	$::=$		Value
	$\mid$	$-\mathsf{h}$	Hole
	$\mid$	$+\mathsf{h}$	Destination
	$\mid$	$()$	Unit
	$\mid$	${}^{\mathrm{v}}\lambda \mathsf{x}\ {}_{\mathsf{m}} \mapsto u$	Function with no free variable
	$\mid$	$\mathtt{Inl}\ v$	Left variant for sum
	$\mid$	$\mathtt{Inr}\ v$	Right variant for sum
	$\mid$	$\mathsf{E}_{\mathsf{m}}\ v$	Exponential

$\mid$	$(v_1 \,,\, v_2)$		Product
$\mid$	${}_{\mathsf{H}}\langle v_2 \,{}_{\wedge}\, v_1 \rangle$		Ampar
$\mid$	$v[\mathsf{H}{\pm}\mathsf{h}']$	M	Shift hole names inside v by h' if they belong to H.

5.3 Syntactic sugar for constructors and commonly used operations

sterm	::=			Syntactic sugar for terms
	$\mid$	$\mathtt{alloc}$	M	Evaluate to a fresh new ampar
	$\mid$	$t \triangleleft t'$	M	Fill destination with supplied term
	$\mid$	$\mathtt{from}'_{\ltimes}\ t$	M	Extract left side of ampar when right side is unit
	$\mid$	${}^{s}\lambda \mathsf{x}\ {}_{\mathsf{m}}\!\mapsto u$	M	Allocate function
	$\mid$	${}^{s}\mathtt{Inl}\ t$	M	Allocate left variant
	$\mid$	${}^{s}\mathtt{Inr}\ t$	M	Allocate right variant
	$\mid$	${}^{s}\mathsf{E}_{\mathsf{m}}\ t$	M	Allocate exponential
	$\mid$	${}^{s}(t_1 \,,\, t_2)$	M	Allocate product

$\mathtt{alloc}$	$\triangleq$	${}_{\mathsf{H}}\langle -1 \,{}_{\wedge}\, +1 \rangle$	$t \triangleleft t'$	$\triangleq$	$t \triangleleft\!\bullet (\mathtt{to}_{\ltimes}\ t')$
$\mathtt{from}'_{\ltimes}\ t$	$\triangleq$	$(\mathtt{from}_{\ltimes}\ (t \rhd \mathtt{map}\ \mathsf{un} \mapsto \mathsf{un}\ ; \mathsf{E}_{1\infty}\ ())) \rhd \mathtt{case}_{1\nu}$ $(\mathsf{st} \,,\, \mathsf{ex}) \mapsto \mathsf{ex} \rhd \mathtt{case}_{1\nu}$ $\mathsf{E}_{1\infty}\ \mathsf{un} \mapsto \mathsf{un}\ ; \mathsf{st}$	${}^{s}\lambda \mathsf{x}\ {}_{\mathsf{m}}\!\mapsto u$	$\triangleq$	$\mathtt{from}'_{\ltimes}($ $\mathtt{alloc} \rhd \mathtt{map}\ \mathsf{d} \mapsto$ $\mathsf{d} \triangleleft (\lambda \mathsf{x}\ {}_{\mathsf{m}}\!\mapsto u)$ $)$
${}^{s}\mathtt{Inl}\ t$	$\triangleq$	$\mathtt{from}'_{\ltimes}($ $\mathtt{alloc} \rhd \mathtt{map}\ \mathsf{d} \mapsto$ $\mathsf{d} \triangleleft \mathtt{Inl} \triangleleft t$ $)$	${}^{s}\mathtt{Inr}\ t$	$\triangleq$	$\mathtt{from}'_{\ltimes}($ $\mathtt{alloc} \rhd \mathtt{map}\ \mathsf{d} \mapsto$ $\mathsf{d} \triangleleft \mathtt{Inr} \triangleleft t$ $)$
${}^{s}\mathsf{E}_{\mathsf{m}}\ t$	$\triangleq$	$\mathtt{from}'_{\ltimes}($ $\mathtt{alloc} \rhd \mathtt{map}\ \mathsf{d} \mapsto$ $\mathsf{d} \triangleleft \mathsf{E}_{\mathsf{m}} \triangleleft t$ $)$	${}^{s}(t_1 \,,\, t_2)$	$\triangleq$	$\mathtt{from}'_{\ltimes}($ $\mathtt{alloc} \rhd \mathtt{map}\ \mathsf{d} \mapsto$ $\mathsf{d} \triangleleft (,) \rhd \mathtt{case}_{1\nu}$ $(\mathsf{d}_1 \,,\, \mathsf{d}_2) \mapsto \mathsf{d}_1 \triangleleft t_1\ ; \mathsf{d}_2 \triangleleft t_2$ $)$

Table 1. Desugaring of syntactic sugar forms for terms

6 TYPE SYSTEM

6.1 Syntax for types, modes, and typing contexts

type, T, U	$::=$			Type
	$\mid$	1		Unit
	$\mid$	$T_1 \oplus T_2$		Sum
	$\mid$	$T_1 \otimes T_2$		Product
	$\mid$	$!_{m}\, T$		Exponential
	$\mid$	$U \ltimes T$		Ampar
	$\mid$	$T\ {}_{m}\!\!\rightarrow U$		Function
	$\mid$	$\lfloor T \rfloor^{m}$		Destination
mode, m, n	$::=$			Mode (Semiring)
	$\mid$	pa		Pair of a multiplicity and age
	$\mid$	☠		Error case (incompatible types, multiplicities, or ages)
mul, p	$::=$			Multiplicity (Semiring, first component of modality)
	$\mid$	1		Linear use
	$\mid$	ω		Non-linear use
age, a	$::=$			Age (Semiring, second component of modality)
	$\mid$	ν		Born now
	$\mid$	$\uparrow$		One scope older
	$\mid$	∞		Infinitely old / static
ctx, Γ, Δ, Π	$::=$			Typing context
	$\mid$	$x :_{m} T$		Variable typing binding
	$\mid$	$+h :_{m} \lfloor T \rfloor^{n}$		Destination typing binding
	$\mid$	$-h : T^{n}$		Hole typing binding
	$\mid$	$m \cdot \Gamma$	M	Multiply the left-most mode of each binding by m
	$\mid$	$\Gamma_1 + \Gamma_2$	M	Sum (incompatible bindings get tagged with ☠)
	$\mid$	$\Gamma_1,\ \Gamma_2$	M	Disjoint sum
	$\mid$	$-\Gamma$	M	Transforms dest bindings into a hole bindings
	$\mid$	$-^{-1}\Gamma$	M	Transforms hole bindings into dest bindings
	$\mid$	$\Gamma[H \pm h']$	M	Shift hole/dest names by h' if they belong to H

6.2 Typing of terms and values

$\boxed{\Gamma \Vdash v : \mathsf{T}}$ *(Typing judgment for values)*

$$\textsc{Ty-val-Hole}\quad \frac{}{-h : \mathsf{T}^{1\nu} \Vdash -h : \mathsf{T}}$$

$$\textsc{Ty-val-Dest}\quad \frac{}{+h :_{1\nu} \lfloor \mathsf{T} \rfloor^{n} \Vdash +h : \lfloor \mathsf{T} \rfloor^{n}}$$

$$\textsc{Ty-val-Unit}\quad \frac{}{\Vdash () : 1}$$

$$\textsc{Ty-val-Fun}\quad \frac{\Delta,\ x :_{m} \mathsf{T} \vdash u : \mathsf{U}}{\Delta \Vdash {}^{\mathrm{v}}\lambda x_{m} \mapsto u : \mathsf{T}\ {}_{m}\!\!\to \mathsf{U}}$$

$$\textsc{Ty-val-Left}\quad \frac{\Gamma \Vdash v_1 : \mathsf{T}_1}{\Gamma \Vdash \mathtt{Inl}\ v_1 : \mathsf{T}_1 \oplus \mathsf{T}_2}$$

$$\textsc{Ty-val-Right}\quad \frac{\Gamma \Vdash v_2 : \mathsf{T}_2}{\Gamma \Vdash \mathtt{Inr}\ v_2 : \mathsf{T}_1 \oplus \mathsf{T}_2}$$

$$\textsc{Ty-val-Prod}\quad \frac{\Gamma_1 \Vdash v_1 : \mathsf{T}_1 \qquad \Gamma_2 \Vdash v_2 : \mathsf{T}_2}{\Gamma_1 + \Gamma_2 \Vdash (v_1\ ,\ v_2) : \mathsf{T}_1 \otimes \mathsf{T}_2}$$

$$\textsc{Ty-val-Exp}\quad \frac{\Gamma \Vdash v' : \mathsf{T}}{n \cdot \Gamma \Vdash \mathsf{E}_{n}\ v' : !_{n}\, \mathsf{T}}$$

$$\textsc{Ty-val-Ampar}\quad \frac{\mathtt{LinOnly}\ \Delta_3 \qquad \mathtt{FinAgeOnly}\ \Delta_3 \qquad 1{\uparrow}\cdot\Delta_1,\ \Delta_3 \Vdash v_1 : \mathsf{T} \qquad \Delta_2,\ (-\Delta_3) \Vdash v_2 : \mathsf{U}}{\Delta_1,\ \Delta_2 \Vdash {}_{hvars(-\Delta_3)}\langle v_2\ {}_{\wedge}\ v_1 \rangle : \mathsf{U} \ltimes \mathsf{T}}$$

$\boxed{\Pi \vdash t : \mathsf{T}}$ *(Typing judgment for terms)*

$$\textsc{Ty-term-Val}\quad \frac{\mathtt{DisposableOnly}\ \Pi \qquad \Delta \Vdash v : \mathsf{T}}{\Pi,\ \Delta \vdash v : \mathsf{T}}$$

$$\textsc{Ty-term-Var}\quad \frac{\mathtt{DisposableOnly}\ \Pi \qquad 1\nu <: m}{\Pi,\ x :_{m} \mathsf{T} \vdash x : \mathsf{T}}$$

$$\textsc{Ty-term-App}\quad \frac{\Pi_1 \vdash t : \mathsf{T} \qquad \Pi_2 \vdash t' : \mathsf{T}\ {}_{m}\!\!\to \mathsf{U}}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright t' : \mathsf{U}}$$

$$\textsc{Ty-term-PatU}\quad \frac{\Pi_1 \vdash t : 1 \qquad \Pi_2 \vdash u : \mathsf{U}}{\Pi_1 + \Pi_2 \vdash t\ ;\ u : \mathsf{U}}$$

$$\textsc{Ty-term-PatS}\quad \frac{\Pi_1 \vdash t : \mathsf{T}_1 \oplus \mathsf{T}_2 \qquad \Pi_2,\ x_1 :_{m} \mathsf{T}_1 \vdash u_1 : \mathsf{U} \qquad \Pi_2,\ x_2 :_{m} \mathsf{T}_2 \vdash u_2 : \mathsf{U}}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \mathtt{case}_{m}\ \{\mathtt{Inl}\ x_1 \mapsto u_1\ ,\ \mathtt{Inr}\ x_2 \mapsto u_2\} : \mathsf{U}}$$

$$\textsc{Ty-term-PatP}\quad \frac{\Pi_1 \vdash t : \mathsf{T}_1 \otimes \mathsf{T}_2 \qquad \Pi_2,\ x_1 :_{m} \mathsf{T}_1,\ x_2 :_{m} \mathsf{T}_2 \vdash u : \mathsf{U}}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \mathtt{case}_{m}\ (x_1\ ,\ x_2) \mapsto u : \mathsf{U}}$$

$$\textsc{Ty-term-PatE}\quad \frac{\Pi_1 \vdash t : !_{n}\, \mathsf{T} \qquad \Pi_2,\ x :_{m \cdot n} \mathsf{T} \vdash u : \mathsf{U}}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \mathtt{case}_{m}\ \mathsf{E}_{n}\ x \mapsto u : \mathsf{U}}$$

$$\textsc{Ty-term-Map}\quad \frac{\Pi_1 \vdash t : \mathsf{U} \ltimes \mathsf{T} \qquad 1{\uparrow}\cdot\Pi_2,\ x :_{1\nu} \mathsf{T} \vdash t' : \mathsf{T}'}{\Pi_1 + \Pi_2 \vdash t \triangleright \mathtt{map}\ x \mapsto t' : \mathsf{U} \ltimes \mathsf{T}'}$$

$$\textsc{Ty-term-ToA}\quad \frac{\Pi \vdash u : \mathsf{U}}{\Pi \vdash \mathtt{to}_{\ltimes}\ u : \mathsf{U} \ltimes 1}$$

$$\textsc{Ty-term-FromA}\quad \frac{\Pi \vdash t : \mathsf{U} \ltimes (!_{1\infty}\, \mathsf{T})}{\Pi \vdash \mathtt{from}_{\ltimes}\ t : \mathsf{U} \otimes (!_{1\infty}\, \mathsf{T})}$$

$$\frac{\Pi \vdash t : \lfloor 1 \rfloor^{n}}{\Pi \vdash t \triangleleft () : 1}\ \textsc{Ty-term-FillU} \qquad \frac{\Pi \vdash t : \lfloor T_1 \oplus T_2 \rfloor^{n}}{\Pi \vdash t \triangleleft \texttt{Inl} : \lfloor T_1 \rfloor^{n}}\ \textsc{Ty-term-FillL} \qquad \frac{\Pi \vdash t : \lfloor T_1 \oplus T_2 \rfloor^{n}}{\Pi \vdash t \triangleleft \texttt{Inr} : \lfloor T_2 \rfloor^{n}}\ \textsc{Ty-term-FillR} \qquad \frac{\Pi \vdash t : \lfloor T_1 \otimes T_2 \rfloor^{n}}{\Pi \vdash t \triangleleft (,) : \lfloor T_1 \rfloor^{n} \otimes \lfloor T_2 \rfloor^{n}}\ \textsc{Ty-term-FillP}$$

$$\frac{\Pi \vdash t : \lfloor !_{n'}\, T \rfloor^{n}}{\Pi \vdash t \triangleleft \mathsf{E}_{n'} : \lfloor T \rfloor^{n' \cdot n}}\ \textsc{Ty-term-FillE} \qquad \frac{\Pi_1 \vdash t : \lfloor T\ {}_{m}\!\!\to U \rfloor^{n} \qquad \Pi_2,\ x :_{m} T \vdash u : U}{\Pi_1 + (1 \uparrow \cdot n) \cdot \Pi_2 \vdash t \triangleleft (\lambda x\ {}_{m}\!\!\mapsto\ u) : 1}\ \textsc{Ty-term-FillF} \qquad \frac{\Pi_1 \vdash t : \lfloor U \rfloor^{n} \qquad \Pi_2 \vdash t' : U \ltimes T}{\Pi_1 + (1 \uparrow \cdot n) \cdot \Pi_2 \vdash t \triangleleft\!\bullet\ t' : T}\ \textsc{Ty-term-FillComp}$$

6.3 Derived typing rules for syntactic sugar forms

$\boxed{\Pi \ {}^{s}\!\!\vdash t : T}$ *(Derived typing judgment for syntactic sugar forms)*

$$\frac{\texttt{DisposableOnly}\ \Pi}{\Pi \ {}^{s}\!\!\vdash \texttt{alloc} : T \ltimes (\lfloor T \rfloor^{1\nu})}\ \textsc{Ty-sterm-Alloc} \qquad \frac{\Pi \vdash t : T \ltimes 1}{\Pi \ {}^{s}\!\!\vdash \texttt{from}'_{\ltimes}\ t : T}\ \textsc{Ty-sterm-FromA'} \qquad \frac{\Pi_1 \vdash t : \lfloor T \rfloor^{n} \qquad \Pi_2 \vdash t' : T}{\Pi_1 + (1 \uparrow \cdot n) \cdot \Pi_2 \ {}^{s}\!\!\vdash t \triangleleft t' : 1}\ \textsc{Ty-sterm-FillLeaf}$$

$$\frac{\Pi_2,\ x :_{m} T \vdash u : U}{\Pi_2 \ {}^{s}\!\!\vdash {}^{s}\!\lambda x\ {}_{m}\!\!\mapsto\ u : T\ {}_{m}\!\!\to U}\ \textsc{Ty-sterm-Fun} \qquad \frac{\Pi_2 \vdash t : T_1}{\Pi_2 \ {}^{s}\!\!\vdash {}^{s}\texttt{Inl}\ t : T_1 \oplus T_2}\ \textsc{Ty-sterm-Left} \qquad \frac{\Pi_2 \vdash t : T_2}{\Pi_2 \ {}^{s}\!\!\vdash {}^{s}\texttt{Inr}\ t : T_1 \oplus T_2}\ \textsc{Ty-sterm-Right}$$

$$\frac{\Pi_2 \vdash t : T}{m \cdot \Pi_2 \ {}^{s}\!\!\vdash {}^{s}\mathsf{E}_{m}\ t : !_{m}\, T}\ \textsc{Ty-sterm-Exp} \qquad \frac{\Pi_{21} \vdash t_1 : T_1 \qquad \Pi_{22} \vdash t_2 : T_2}{\Pi_{21} + \Pi_{22} \ {}^{s}\!\!\vdash {}^{s}(t_1\,,\, t_2) : T_1 \otimes T_2}\ \textsc{Ty-sterm-Prod}$$

7 EVALUATION CONTEXTS AND SEMANTICS

7.1 Evaluation contexts forms

$$
\begin{array}{lcll}
\mathit{ectx},\ c & ::= & & \text{Evaluation context component} \\
& | & \square \rhd t' & \\
& | & v \rhd \square & \\
& | & \square\ ;\ u & \\
& | & \square \rhd \texttt{case}_{m}\ \{\texttt{Inl}\ x_1 \mapsto u_1\ ,\ \texttt{Inr}\ x_2 \mapsto u_2\} & \\
& | & \square \rhd \texttt{case}_{m}\ (x_1\ ,\ x_2) \mapsto u & \\
& | & \square \rhd \texttt{case}_{m}\ \mathsf{E}_{n}\ x \mapsto u & \\
& | & \square \rhd \texttt{map}\ x \mapsto t' & \\
& | & \texttt{to}_{\ltimes}\ \square & \\
& | & \texttt{from}_{\ltimes}\ \square & \\
& | & \square \triangleleft () & \\
& | & \square \triangleleft \texttt{Inl} & \\
& | & \square \triangleleft \texttt{Inr} & \\
& | & \square \triangleleft \mathsf{E}_{m} & \\
& | & \square \triangleleft (,) & \\
& | & \square \triangleleft (\lambda x\ {}_{m}\!\!\mapsto\ u) & \\
& | & \square \triangleleft\!\bullet\ t' & \\
& | & v \triangleleft\!\bullet\ \square &
\end{array}
$$

	$\mid$	${}^{\mathrm{op}}_{\mathsf{H}}\langle v_2 \,{}_{\wedge} \square$	Open ampar, binding hole names in the next components
ectxs, C	$::=$		Evaluation context stack
	$\mid$	$\square$	Represent the empty stack / "identity" evaluation context
	$\mid$	$C \circ c$	Push c on top of C
	$\mid$	$C[\mathsf{h} :=_{\mathsf{H}} v]$ $\quad$ M	Fill h in C with value v (that may contain holes)

7.2 Typing of evaluation contexts and commands

$\boxed{\Delta \dashv C : \mathsf{T} \rightarrowtail \mathsf{U}_0}$ *(Typing judgment for evaluation contexts)*

$$\textsc{Ty-ectxs-Id}\quad \frac{}{\dashv \square : \mathsf{U}_0 \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-App-Foc1}\quad \frac{\mathsf{m}\cdot\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2 \vdash t' : \mathsf{T}\ {}_{\mathsf{m}}\!\!\to \mathsf{U}}{\Delta_1 \dashv C \circ (\square \triangleright t') : \mathsf{T} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-App-Foc2}\quad \frac{\mathsf{m}\cdot\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_1 \vdash v : \mathsf{T}}{\Delta_2 \dashv C \circ (v \triangleright \square) : (\mathsf{T}\ {}_{\mathsf{m}}\!\!\to \mathsf{U}) \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-PatU-Foc}\quad \frac{\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2 \vdash u : \mathsf{U}}{\Delta_1 \dashv C \circ (\square\ ;\, u) : 1 \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-PatS-Foc}\quad \frac{\mathsf{m}\cdot\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2,\ \mathsf{x}_1 :_{\mathsf{m}} \mathsf{T}_1 \vdash u_1 : \mathsf{U} \qquad \Delta_2,\ \mathsf{x}_2 :_{\mathsf{m}} \mathsf{T}_2 \vdash u_2 : \mathsf{U}}{\Delta_1 \dashv C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}) : (\mathsf{T}_1 \oplus \mathsf{T}_2) \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-PatP-Foc}\quad \frac{\mathsf{m}\cdot\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2,\ \mathsf{x}_1 :_{\mathsf{m}} \mathsf{T}_1,\ \mathsf{x}_2 :_{\mathsf{m}} \mathsf{T}_2 \vdash u : \mathsf{U}}{\Delta_1 \dashv C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\, \mathsf{x}_2) \mapsto u) : (\mathsf{T}_1 \otimes \mathsf{T}_2) \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-PatE-Foc}\quad \frac{\mathsf{m}\cdot\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2,\ \mathsf{x} :_{\mathsf{m}\cdot\mathsf{m}'} \mathsf{T} \vdash u : \mathsf{U}}{\Delta_1 \dashv C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, \mathtt{E}_{\mathsf{m}'}\ \mathsf{x} \mapsto u) : !_{\mathsf{m}'}\, \mathsf{T} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-Map-Foc}\quad \frac{\Delta_1,\ \Delta_2 \dashv C : \mathsf{U} \ltimes \mathsf{T}' \rightarrowtail \mathsf{U}_0 \qquad 1\!\uparrow\!\cdot\Delta_2,\ \mathsf{x} :_{1\nu} \mathsf{T} \vdash t' : \mathsf{T}'}{\Delta_1 \dashv C \circ (\square \triangleright \mathtt{map}\ \mathsf{x} \mapsto t') : (\mathsf{U} \ltimes \mathsf{T}) \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-ToA-Foc}\quad \frac{\Delta \dashv C : (\mathsf{U} \ltimes 1) \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\mathtt{to}_{\ltimes}\ \square) : \mathsf{U} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-FromA-Foc}\quad \frac{\Delta \dashv C : (\mathsf{U} \otimes (!_{1\infty}\, \mathsf{T})) \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\mathtt{from}_{\ltimes}\ \square) : (\mathsf{U} \ltimes (!_{1\infty}\, \mathsf{T})) \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-FillU-Foc}\quad \frac{\Delta \dashv C : 1 \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\square \triangleleft ()) : \lfloor 1 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-FillL-Foc}\quad \frac{\Delta \dashv C : \lfloor \mathsf{T}_1 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\square \triangleleft \mathtt{Inl}) : \lfloor \mathsf{T}_1 \oplus \mathsf{T}_2 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-FillR-Foc}\quad \frac{\Delta \dashv C : \lfloor \mathsf{T}_2 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\square \triangleleft \mathtt{Inr}) : \lfloor \mathsf{T}_1 \oplus \mathsf{T}_2 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

$$\textsc{Ty-ectxs-FillP-Foc}\quad \frac{\Delta \dashv C : (\lfloor \mathsf{T}_1 \rfloor^{\mathsf{n}} \otimes \lfloor \mathsf{T}_2 \rfloor^{\mathsf{n}}) \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\square \triangleleft (,)) : \lfloor \mathsf{T}_1 \otimes \mathsf{T}_2 \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

TY-ECTXS-FILLE-FOC
$$\frac{\Delta \dashv C : \lfloor \mathsf{T} \rfloor^{\mathsf{m}\cdot\mathsf{n}} \rightarrowtail \mathsf{U}_0}{\Delta \dashv C \circ (\square \triangleleft \mathsf{E}_{\mathsf{m}}) : \lfloor !_{\mathsf{m}}\, \mathsf{T} \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

TY-ECTXS-FILLF-FOC
$$\frac{\Delta_1,\ (1\!\uparrow\!\cdot\mathsf{n})\cdot\Delta_2 \dashv C : 1 \rightarrowtail \mathsf{U}_0 \qquad \Delta_2,\ \mathsf{x} :_{\mathsf{m}} \mathsf{T} \vdash u : \mathsf{U}}{\Delta_1 \dashv C \circ (\square \triangleleft (\lambda \mathsf{x}\,_{\mathsf{m}}\!\mapsto\ u)) : \lfloor \mathsf{T}\ _{\mathsf{m}}\!\!\to \mathsf{U} \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

TY-ECTXS-FILLCOMP-FOC1
$$\frac{\Delta_1,\ (1\!\uparrow\!\cdot\mathsf{n})\cdot\Delta_2 \dashv C : \mathsf{T} \rightarrowtail \mathsf{U}_0 \qquad \Delta_2 \vdash t' : \mathsf{U} \ltimes \mathsf{T}}{\Delta_1 \dashv C \circ (\square \triangleleft\!\bullet\ t') : \lfloor \mathsf{U} \rfloor^{\mathsf{n}} \rightarrowtail \mathsf{U}_0}$$

TY-ECTXS-FILLCOMP-FOC2
$$\frac{\Delta_1,\ (1\!\uparrow\!\cdot\mathsf{n})\cdot\Delta_2 \dashv C : \mathsf{T} \rightarrowtail \mathsf{U}_0 \qquad \Delta_1 \vdash v : \lfloor \mathsf{U} \rfloor^{\mathsf{n}}}{\Delta_2 \dashv C \circ (v \triangleleft\!\bullet\ \square) : \mathsf{U} \ltimes \mathsf{T} \rightarrowtail \mathsf{U}_0}$$

TY-ECTXS-OPENAMPAR-FOC
$$\frac{\begin{array}{c} \mathit{hvars}(C)\ \#\#\ \mathit{hvars}(-\Delta_3) \\ \mathtt{LinOnly}\ \Delta_3 \\ \mathtt{FinAgeOnly}\ \Delta_3 \\ \Delta_1,\ \Delta_2 \dashv C : (\mathsf{U} \ltimes \mathsf{T}') \rightarrowtail \mathsf{U}_0 \\ \Delta_2,\ -\Delta_3 \Vdash v_2 : \mathsf{U} \end{array}}{1\!\uparrow\!\cdot\Delta_1,\ \Delta_3 \dashv C \circ ({}^{\mathrm{op}}_{\mathit{hvars}(-\Delta_3)}\langle v_2\ {}_{\wedge} \square) : \mathsf{T}' \rightarrowtail \mathsf{U}_0}$$

$\boxed{\vdash C[t] : \mathsf{T}}$ *(Typing judgment for commands)*

TY-CMD
$$\frac{\Delta \dashv C : \mathsf{T} \rightarrowtail \mathsf{U}_0 \qquad \Delta \vdash t : \mathsf{T}}{\vdash C[t] : \mathsf{U}_0}$$

7.3 Small-step semantics

$\boxed{C[t] \longrightarrow C'[t']}$ *(Small-step evaluation of commands)*

SEM-APP-FOC1
$$\frac{\mathtt{NotVal}\ \ t}{C[t \rhd t'] \longrightarrow (C \circ (\square \rhd t'))[t]}$$

SEM-APP-UNFOC1
$$\frac{}{(C \circ (\square \rhd t'))[v] \longrightarrow C[v \rhd t']}$$

SEM-APP-FOC2
$$\frac{\mathtt{NotVal}\ \ t'}{C[v \rhd t'] \longrightarrow (C \circ (v \rhd \square))[t']}$$

SEM-APP-UNFOC2
$$\frac{}{(C \circ (v \rhd \square))[v'] \longrightarrow C[v \rhd v']}$$

SEM-APP-RED
$$\frac{}{C[v \rhd ({}^{\mathsf{v}}\lambda \mathsf{x}\,_{\mathsf{m}}\!\mapsto\ u)] \longrightarrow C[u[\mathsf{x} := v]]}$$

SEM-PATU-FOC
$$\frac{\mathtt{NotVal}\ \ t}{C[t\ ;u] \longrightarrow (C \circ (\square\ ;u))[t]}$$

SEM-PATU-UNFOC
$$\frac{}{(C \circ (\square\ ;u))[v] \longrightarrow C[v\ ;u]}$$

SEM-PATU-RED
$$\frac{}{C[()\ ;u] \longrightarrow C[u]}$$

SEM-PATS-FOC
$$\frac{\mathtt{NotVal}\ \ t}{C[t \rhd \mathtt{case}_{\mathsf{m}}\,\{\mathtt{Inl}\,\mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\,\mathsf{x}_2 \mapsto u_2\}] \longrightarrow (C \circ (\square \rhd \mathtt{case}_{\mathsf{m}}\,\{\mathtt{Inl}\,\mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\,\mathsf{x}_2 \mapsto u_2\}))[t]}$$

Sem-PatS-Unfoc

$$\dfrac{}{(C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}))[v] \longrightarrow C[v \triangleright \mathtt{case}_{\mathsf{m}}\, \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}]}$$

Sem-PatL-Red

$$\dfrac{}{C[(\mathtt{Inl}\ v_1) \triangleright \mathtt{case}_{\mathsf{m}}\, \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}] \longrightarrow C[u_1[\mathsf{x}_1 \coloneqq v_1]]}$$

Sem-PatR-Red

$$\dfrac{}{C[(\mathtt{Inr}\ v_2) \triangleright \mathtt{case}_{\mathsf{m}}\, \{\mathtt{Inl}\ \mathsf{x}_1 \mapsto u_1\,,\ \mathtt{Inr}\ \mathsf{x}_2 \mapsto u_2\}] \longrightarrow C[u_2[\mathsf{x}_2 \coloneqq v_2]]}$$

Sem-PatP-Foc

$$\dfrac{\mathtt{NotVal}\ \ t}{C[t \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\ \mathsf{x}_2) \mapsto u] \longrightarrow (C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\ \mathsf{x}_2) \mapsto u))[t]}$$

Sem-PatP-Unfoc

$$\dfrac{}{(C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\ \mathsf{x}_2) \mapsto u))[v] \longrightarrow C[v \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\ \mathsf{x}_2) \mapsto u]}$$

Sem-PatP-Red

$$\dfrac{}{C[(v_1\,,\ v_2) \triangleright \mathtt{case}_{\mathsf{m}}\, (\mathsf{x}_1\,,\ \mathsf{x}_2) \mapsto u] \longrightarrow C[u[\mathsf{x}_1 \coloneqq v_1][\mathsf{x}_2 \coloneqq v_2]]}$$

Sem-PatE-Foc

$$\dfrac{\mathtt{NotVal}\ \ t}{C[t \triangleright \mathtt{case}_{\mathsf{m}}\, \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u] \longrightarrow (C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u))[t]}$$

Sem-PatE-Unfoc

$$\dfrac{}{(C \circ (\square \triangleright \mathtt{case}_{\mathsf{m}}\, \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u))[v] \longrightarrow C[v \triangleright \mathtt{case}_{\mathsf{m}}\, \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u]}$$

Sem-PatE-Red

$$\dfrac{}{C[\mathsf{E}_{\mathsf{n}}\ v' \triangleright \mathtt{case}_{\mathsf{m}}\, \mathsf{E}_{\mathsf{n}}\ \mathsf{x} \mapsto u] \longrightarrow C[u[\mathsf{x} \coloneqq v']]}$$

Sem-Map-Foc

$$\dfrac{\mathtt{NotVal}\ \ t}{C[t \triangleright \mathtt{map}\ \mathsf{x} \mapsto t'] \longrightarrow (C \circ (\square \triangleright \mathtt{map}\ \mathsf{x} \mapsto t'))[t]}$$

Sem-Map-Unfoc

$$\dfrac{}{(C \circ (\square \triangleright \mathtt{map}\ \mathsf{x} \mapsto t'))[v] \longrightarrow C[v \triangleright \mathtt{map}\ \mathsf{x} \mapsto t']}$$

Sem-Map-Red-OpenAmpar-Foc

$$\dfrac{\mathsf{h}' \ = \ max(hvars(C))+1}{C[{}_{\mathsf{H}}\langle v_2 \,{}_{\wedge}\, v_1\rangle \triangleright \mathtt{map}\ \mathsf{x} \mapsto t'] \longrightarrow (C \circ ({}^{\mathrm{op}}_{\mathsf{H}\pm\mathsf{h}'}\langle v_2[\mathsf{H}\pm\mathsf{h}'] \,{}_{\wedge}\, \square))[t'[\mathsf{x} \coloneqq v_1[\mathsf{H}\pm\mathsf{h}']]]}$$

Sem-OpenAmpar-Unfoc

$$\dfrac{}{(C \circ^{\mathrm{op}}_{\mathsf{H}} \langle v_2 \,{}_{\wedge}\, \square)[v_1] \longrightarrow C[{}_{\mathsf{H}}\langle v_2 \,{}_{\wedge}\, v_1\rangle]}$$

Sem-ToA-Foc

$$\dfrac{\mathtt{NotVal}\ \ u}{C[\mathtt{to}_{\ltimes}\ u] \longrightarrow (C \circ (\mathtt{to}_{\ltimes}\ \square))[u]}$$

$$\textsc{Sem-ToA-Unfoc}\quad \frac{}{(C \circ (\mathtt{to}_{\ltimes}\ \square))[v_2] \longrightarrow C[\mathtt{to}_{\ltimes}\ v_2]}$$

$$\textsc{Sem-ToA-Red}\quad \frac{}{C[\mathtt{to}_{\ltimes}\ v_2] \longrightarrow C[{}_{\{\,\}}\langle v_2\ {}_{\wedge}()\rangle]}$$

$$\textsc{Sem-FromA-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[\mathtt{from}_{\ltimes}\ t] \longrightarrow (C \circ (\mathtt{from}_{\ltimes}\ \square))[t]}$$

$$\textsc{Sem-FromA-Unfoc}\quad \frac{}{(C \circ (\mathtt{from}_{\ltimes}\ \square))[v] \longrightarrow C[\mathtt{from}_{\ltimes}\ v]}$$

$$\textsc{Sem-FromA-Red}\quad \frac{}{C[\mathtt{from}_{\ltimes}\ {}_{\{\,\}}\langle v_2\ {}_{\wedge}\mathsf{E}_{1\infty}\ v_1\rangle] \longrightarrow C[(v_2\,,\ \mathsf{E}_{1\infty}\ v_1)]}$$

$$\textsc{Sem-FillU-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft ()] \longrightarrow (C \circ (\square \triangleleft ()))[t]}$$

$$\textsc{Sem-FillU-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft ()))[v] \longrightarrow C[v \triangleleft ()]}$$

$$\textsc{Sem-FillU-Red}\quad \frac{}{C[+\mathsf{h} \triangleleft ()] \longrightarrow C[\mathsf{h} \coloneqq_{\{\,\}}\ ()][()]}$$

$$\textsc{Sem-FillL-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft \mathtt{Inl}] \longrightarrow (C \circ (\square \triangleleft \mathtt{Inl}))[t]}$$

$$\textsc{Sem-FillL-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft \mathtt{Inl}))[v] \longrightarrow C[v \triangleleft \mathtt{Inl}]}$$

$$\textsc{Sem-FillL-Red}\quad \frac{\mathsf{h}' \ = \ \mathit{max}(\mathit{hvars}(C) \cup \{\mathsf{h}\})+1}{C[+\mathsf{h} \triangleleft \mathtt{Inl}] \longrightarrow C[\mathsf{h} \coloneqq_{\{\mathsf{h}'+1\}}\ \mathtt{Inl}\ -(\mathsf{h}'+1)][+(\mathsf{h}'+1)]}$$

$$\textsc{Sem-FillR-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft \mathtt{Inr}] \longrightarrow (C \circ (\square \triangleleft \mathtt{Inr}))[t]}$$

$$\textsc{Sem-FillR-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft \mathtt{Inr}))[v] \longrightarrow C[v \triangleleft \mathtt{Inr}]}$$

$$\textsc{Sem-FillR-Red}\quad \frac{\mathsf{h}' \ = \ \mathit{max}(\mathit{hvars}(C) \cup \{\mathsf{h}\})+1}{C[+\mathsf{h} \triangleleft \mathtt{Inr}] \longrightarrow C[\mathsf{h} \coloneqq_{\{\mathsf{h}'+1\}}\ \mathtt{Inr}\ -(\mathsf{h}'+1)][+(\mathsf{h}'+1)]}$$

$$\textsc{Sem-FillE-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft \mathsf{E}_{\mathsf{m}}] \longrightarrow (C \circ (\square \triangleleft \mathsf{E}_{\mathsf{m}}))[t]}$$

$$\textsc{Sem-FillE-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft \mathsf{E}_{\mathsf{m}}))[v] \longrightarrow C[v \triangleleft \mathsf{E}_{\mathsf{m}}]}$$

$$\textsc{Sem-FillE-Red}\quad \frac{\mathsf{h}' \ = \ \mathit{max}(\mathit{hvars}(C) \cup \{\mathsf{h}\})+1}{C[+\mathsf{h} \triangleleft \mathsf{E}_{\mathsf{m}}] \longrightarrow C[\mathsf{h} \coloneqq_{\{\mathsf{h}'+1\}}\ \mathsf{E}_{\mathsf{m}}\ -(\mathsf{h}'+1)][+(\mathsf{h}'+1)]}$$

$$\textsc{Sem-FillP-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft (,)] \longrightarrow (C \circ (\square \triangleleft (,)))[t]}$$

$$\textsc{Sem-FillP-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft (,)))[v] \longrightarrow C[v \triangleleft (,)]}$$

$$\textsc{Sem-FillP-Red}\quad \frac{\mathsf{h}' \ = \ \mathit{max}(\mathit{hvars}(C) \cup \{\mathsf{h}\})+1}{C[+\mathsf{h} \triangleleft (,)] \longrightarrow C[\mathsf{h} \coloneqq_{\{\mathsf{h}'+1,\mathsf{h}'+2\}}\ (-(\mathsf{h}'+1)\,,\ -(\mathsf{h}'+2))][(+(\mathsf{h}'+1)\,,\ +(\mathsf{h}'+2))]}$$

$$\textsc{Sem-FillF-Foc}\quad \frac{\mathtt{NotVal}\ t}{C[t \triangleleft (\lambda \mathsf{x}\ {}_{\mathsf{m}}\!\mapsto\ u)] \longrightarrow (C \circ (\square \triangleleft (\lambda \mathsf{x}\ {}_{\mathsf{m}}\!\mapsto\ u)))[t]}$$

$$\textsc{Sem-FillF-Unfoc}\quad \frac{}{(C \circ (\square \triangleleft (\lambda \mathsf{x}\ _{\mathsf{m}}\!\mapsto\ u)))[v] \longrightarrow C[v \triangleleft (\lambda \mathsf{x}\ _{\mathsf{m}}\!\mapsto\ u)]}$$

$$\textsc{Sem-FillF-Red}\quad \frac{}{C[+\mathsf{h} \triangleleft (\lambda \mathsf{x}\ _{\mathsf{m}}\!\mapsto\ u)] \longrightarrow C[\mathsf{h} :=_{\{\}}\ {}^{\mathrm{v}}\!\lambda \mathsf{x}\ _{\mathsf{m}}\!\mapsto\ u][()]}$$

$$\textsc{Sem-FillComp-Foc1}\quad \frac{\mathtt{NotVal}\ \ t}{C[t \triangleleft\!\bullet\, t'] \longrightarrow (C \circ (\square \triangleleft\!\bullet\, t'))[t]}$$

$$\textsc{Sem-FillComp-Unfoc1}\quad \frac{}{(C \circ (\square \triangleleft\!\bullet\, t'))[v] \longrightarrow C[v \triangleleft\!\bullet\, t']}$$

$$\textsc{Sem-FillComp-Foc2}\quad \frac{\mathtt{NotVal}\ \ t'}{C[v \triangleleft\!\bullet\, t'] \longrightarrow (C \circ (v \triangleleft\!\bullet\, \square))[t']}$$

$$\textsc{Sem-FillComp-Unfoc2}\quad \frac{}{(C \circ (v \triangleleft\!\bullet\, \square))[v'] \longrightarrow C[v \triangleleft\!\bullet\, v']}$$

$$\textsc{Sem-FillComp-Red}\quad \frac{\mathsf{h}' \;=\; \mathit{max}(\mathit{hvars}(C) \cup \{\mathsf{h}\}) + 1}{C[+\mathsf{h} \triangleleft\!\bullet\, _{\mathsf{H}}\langle v_2\ _{\wedge}\ v_1\rangle] \longrightarrow C[\mathsf{h} :=_{(\mathsf{H} \pm \mathsf{h}')}\ v_2[\mathsf{H} \pm \mathsf{h}']][v_1[\mathsf{H} \pm \mathsf{h}']]}$$

8 REMARKS ON THE COQ PROOFS

- Not particularly elegant. Max number of goals observed 232 (solved by a single call to the `congruence` tactic). When you have a computer, brute force is a viable strategy. (in particular, no semiring formalisation, it was quicker to do directly)
- Rules generated by ott, same as in the article (up to some notational difference). Contexts are not generated purely by syntax, and are interpreted in a semantic domain (finite functions).
- Reasoning on closed terms avoids almost all complications on binder manipulation. Makes proofs tractable.
- Finite functions: making a custom library was less headache than using existing libraries (including `MMap`). Existing libraries don't provide some of the tools that we needed, but the most important factor ended up being the need for a modicum of dependency between key and value. There wasn't really that out there. Backed by actual functions for simplicity; cost: equality is complicated.
- Most of the proofs done by author with very little prior experience to Coq.
- Did proofs in Coq because context manipulations are tricky.
- Context sum made total by adding an extra invalid *mode* (rather than an extra context). It seems to be much simpler this way.
- It might be a good idea to provide statistics on the number of lemmas and size of Coq codebase.
- (possibly) renaming as permutation, inspired by nominal sets, make more lemmas don't require a condition (but some lemmas that wouldn't in a straight renaming do in exchange).
- (possibly) methodology: assume a lot of lemmas, prove main theorem, prove assumptions, some wrong, fix. A number of wrong lemma initially assumed, but replacing them by correct variant was always easy to fix in proofs.
- Axioms that we use and why (in particular setoid equality not very natural with ott-generated typing rules).
- Talk about the use and benefits of Copilot.

REFERENCES

[1] Thomas Bagrel. 2024. Destination-passing style programming: a Haskell implementation. https://inria.hal.science/hal-04406360