

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>

1 TERM AND VALUE SYNTAX

`var, x, y, d, un` Term-level variable name
`k` Index for ranges

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

hvar, h	$::=$		Hole or destination name (\mathbb{N})
		$h+h'$	M
		$h[H\pm h']$	M Shift by h' if $h \in H$
		$\max(H)$	M Maximum of a set of holes
hvars, H	$::=$		Set of hole names
		$\{h_1, \dots, h_k\}$	
		$H_1 \cup H_2$	M Union of sets
		$H\pm h'$	M Shift all names from H by h' .
		$\text{hvars}(\Gamma)$	M Hole names of a context (requires ctx .)
		$\text{hvars}(C)$	M Hole names of an evaluation context
term, t, u	$::=$		Term
		v	Value
		x	Variable
		$t \triangleright t'$	Application
		$t ; u$	Pattern-match on unit
		$t \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}$	Pattern-match on sum
		$t \triangleright \text{case}_m (x_1, x_2) \mapsto u$	Pattern-match on product
		$t \triangleright \text{case}_m E_n x \mapsto u$	Pattern-match on exponential
		$t \triangleright \text{map } x \mapsto t'$	Map over the right side of ampar t
		$\text{to}_x u$	Wrap u into a trivial ampar
		$\text{from}_x t$	Convert ampar with no dest remaining
		$t \triangleleft ()$	Fill destination with unit
		$t \triangleleft \text{Inl}$	Fill destination with left variant
		$t \triangleleft \text{Inr}$	Fill destination with right variant
		$t \triangleleft E_m$	Fill destination with exponential const

	$t \triangleleft (,)$		Fill destination with product constructor
	$t \triangleleft (\lambda x_m \mapsto u)$		Fill destination with function
	$t \triangleleft \bullet t'$		Fill destination with root of ampar t'
	$t[x := v]$	M	
	alloc	M	
	from $'_x t$	M	
	$t \triangleleft t'$	M	
	$\lambda x_m \mapsto u$	M	Lambda abstraction
	$\text{Inl } t$	M	Left variant for sum
	$\text{Inr } t$	M	Right variant for sum
	$\text{E}_m t$	M	Exponential
	$\text{E}(t_1, t_2)$	M	Product
val, v	$::=$		Term value
	$-h$		Hole
	$+h$		Destination
	$()$		Unit
	$\lambda x_m \mapsto u$		Lambda abstraction
	$\text{Inl } v$		Left variant for sum
	$\text{Inr } v$		Right variant for sum
	$\text{E}_m v$		Exponential
	(v_1, v_2)		Product
	$h\langle v_2, v_1 \rangle$		Ampar
	$v[H \vdash h']$	M	Shift hole names inside v by h' if they belong to H .
ctx, c	$::=$		Evaluation context component
	$\square \triangleright t'$		Application
	$v \triangleright \square$		Application
	$\square ; u$		Pattern-match on unit
	$\square \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}$		Pattern-match on sum
	$\square \triangleright \text{case}_m (x_1, x_2) \mapsto u$		Pattern-match on product
	$\square \triangleright \text{case}_m \text{E}_n x \mapsto u$		Pattern-match on exponential
	$\square \triangleright \text{map } x \mapsto t'$		Map over the right side of ampar
	to $_x \square$		Wrap into a trivial ampar
	from $_x \square$		Convert ampar with no dest remaining into
	$\square \triangleleft ()$		Fill destination with unit
	$\square \triangleleft \text{Inl}$		Fill destination with left variant
	$\square \triangleleft \text{Inr}$		Fill destination with right variant
	$\square \triangleleft \text{E}_m$		Fill destination with exponential constructor
	$\square \triangleleft (,)$		Fill destination with product constructor
	$\square \triangleleft (\lambda x_m \mapsto u)$		Fill destination with function
	$\square \triangleleft \bullet t'$		Fill destination with root of ampar
	$v \triangleleft \bullet \square$		Fill destination with root of ampar
	$h^{\text{op}} \langle v_2, \square \rangle$		Open ampar. Only new addition to term sh

$ectxs, C$	$::=$	Evaluation context stack
	\square	Represent the empty stack / "identity" evaluation context
	$C \circ c$	Push c on top of C
	$C[h :=_H v] \quad M$	Fill h in C with value v (that may contain holes)

Desugaring

$\text{alloc} \triangleq {}_H(-1, +1)$
 $\text{from}'_{\mathbb{K}} t \triangleq (\text{from}_{\mathbb{K}} (t \triangleright \text{map } \text{un} \mapsto \text{un} ; E_{1\infty} ())) \triangleright \text{case}_{1v}$
 $(x_1, x_2) \mapsto x_2 \triangleright \text{case}_{1v}$
 $E_{1\infty} x_3 \mapsto x_3 ; x_1$
 $t \triangleleft t' \triangleq t \triangleleft (\text{to}_{\mathbb{K}} t')$
 ${}^s\lambda_{x_m} \mapsto u \triangleq \text{from}'_{\mathbb{K}} ($
 $\quad \text{alloc} \triangleright \text{map } d \mapsto$
 $\quad d \triangleleft (\lambda_{x_m} \mapsto u)$
 $\quad)$
 ${}^s\text{Inl } t \triangleq \text{from}'_{\mathbb{K}} ($
 $\quad \text{alloc} \triangleright \text{map } d \mapsto$
 $\quad d \triangleleft \text{Inl} \triangleleft t$
 $\quad)$
 ${}^s\text{Inr } t \triangleq \text{from}'_{\mathbb{K}} ($
 $\quad \text{alloc} \triangleright \text{map } d \mapsto$
 $\quad d \triangleleft \text{Inr} \triangleleft t$
 $\quad)$
 ${}^sE_m t \triangleq \text{from}'_{\mathbb{K}} ($
 $\quad \text{alloc} \triangleright \text{map } d \mapsto$
 $\quad d \triangleleft E_m \triangleleft t$
 $\quad)$
 ${}^s(t_1, t_2) \triangleq \text{from}'_{\mathbb{K}} ($
 $\quad \text{alloc} \triangleright \text{map } d \mapsto$
 $\quad d \triangleleft (,) \triangleright \text{case}_{1v}$
 $\quad (d_1, d_2) \mapsto d_1 \triangleleft t_1 ; d_2 \triangleleft t_2$
 $\quad)$

2 TYPE SYSTEM

type, T, U	$::=$	Type
	1	Unit
	$T_1 \oplus T_2$	Sum
	$T_1 \otimes T_2$	Product
	$!_m T$	Exponential
	$U \ltimes T$	Ampar type (consuming T yields U)
	$T \multimap U$	Function
	$[T]^m$	Destination

$\text{mode}, m, n \quad ::= \quad \text{Mode (Semiring)}$

		pa		Pair of a multiplicity and age
		ω		Error case (incompatible types, multiplicities, or ages)
		$\text{m}_1 \cdot \dots \cdot \text{m}_k$	M	Semiring product
mul, p	::=			Multiplicity (first component of modality)
		1		Linear. Neutral element of the product
		ω		Non-linear. Absorbing for the product
		$\text{p}_1 \cdot \dots \cdot \text{p}_k$	M	Semiring product
age, a	::=			Age (second component of modality)
		ν		Born now. Neutral element of the product
		\uparrow		One scope older
		∞		Infinitely old / static. Absorbing for the product
		$\text{a}_1 \cdot \dots \cdot \text{a}_k$	M	Semiring product
$\text{ctx}, \Gamma, \Delta, \Pi$::=			Typing context
		$\text{x} :_{\text{m}} \text{T}$		
		$+\text{h} :_{\text{m}} [\text{T}]^n$		
		$-\text{h} : \text{T}^n$		
		$\text{m}\Gamma$	M	Multiply each binding by m
		$\Gamma_1 + \Gamma_2$	M	Sum contexts Γ_1 and Γ_2 . Duplicate keys with incompatible values
		Γ_1, Γ_2	M	Disjoint sum/union of contexts Γ_1 and Γ_2 .
		$-\Gamma$	M	Transforms dest bindings into a hole bindings (requires ctx_Dest)
		$^{-1}\Gamma$	M	Transforms hole bindings into dest bindings with left mode 1ν (requires ctx_Dest)
		$\Gamma[\text{H} \Leftarrow \text{h}']$	M	Shift hole/dest names by h' if they belong to H

$\Gamma \Vdash \nu : \text{T}$

(Typing of values (raw))

$\frac{\text{TyR-VAL-H}}{-h : T^{1\nu} \Vdash -h : T}$	$\frac{\text{TyR-VAL-D}}{+h :_{1\nu} [T]^n \Vdash +h : [T]^n}$	$\frac{\text{TyR-VAL-U}}{\Vdash () : 1}$	$\frac{\text{TyR-VAL-F}}{\Delta, x :_m T \vdash u : U}$ $\Delta \Vdash \lambda x_m \mapsto u : T_m \multimap U$	
$\frac{\text{TyR-VAL-L}}{\Gamma \Vdash v_1 : T_1}$ $\Gamma \Vdash \text{Inl } v_1 : T_1 \oplus T_2$	$\frac{\text{TyR-VAL-R}}{\Gamma \Vdash v_2 : T_2}$ $\Gamma \Vdash \text{Inr } v_2 : T_1 \oplus T_2$	$\frac{\text{TyR-VAL-P}}{\Gamma_1 \Vdash v_1 : T_1 \quad \Gamma_2 \Vdash v_2 : T_2}$ $\Gamma_1 + \Gamma_2 \Vdash (v_1, v_2) : T_1 \otimes T_2$		
$\frac{\text{TyR-VAL-E}}{\Gamma \Vdash v' : T}$ $n \cdot \Gamma \Vdash e_n v' : !_n T$				$\frac{\text{TyR-VAL-A}}{\Delta_1, \Delta_2 \Vdash \text{hvars}(-\Delta_3) \langle v_2, v_1 \rangle : U \ltimes T}$

$\boxed{\Pi \vdash t : T}$		<i>(Typing of terms)</i>	
$\frac{\text{TY-TERM-VAL} \quad \text{DisposableOnly } \Pi \quad \Delta \Vdash v : T}{\Pi, \Delta \vdash v : T}$		$\frac{\text{TY-TERM-VAR} \quad \text{DisposableOnly } \Pi \quad 1v <: m}{\Pi, x :_m T \vdash x : T}$	
		$\frac{\text{TY-TERM-APP} \quad \Pi_1 \vdash t : T \quad \Pi_2 \vdash t' : T_m \rightarrow U}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright t' : U}$	
		$\text{TY-TERM-PATS} \quad \frac{\Pi_1 \vdash t : T_1 \oplus T_2 \quad \Pi_2, x_1 :_m T_1 \vdash u_1 : U \quad \Pi_2, x_2 :_m T_2 \vdash u_2 : U}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \} : U}$	
$\text{TY-TERM-PATU} \quad \frac{\Pi_1 \vdash t : 1 \quad \Pi_2 \vdash u : U}{\Pi_1 + \Pi_2 \vdash t ; u : U}$			
$\text{TY-TERM-PATP} \quad \frac{\Pi_1 \vdash t : T_1 \otimes T_2 \quad \Pi_2, x_1 :_m T_1, x_2 :_m T_2 \vdash u : U}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \text{case}_m (x_1, x_2) \mapsto u : U}$		$\text{TY-TERM-PATE} \quad \frac{\Pi_1 \vdash t : !_n T \quad \Pi_2, x :_{m \cdot n} T \vdash u : U}{m \cdot \Pi_1 + \Pi_2 \vdash t \triangleright \text{case}_m \varepsilon_n x \mapsto u : U}$	
$\text{TY-TERM-MAP} \quad \frac{\Pi_1 \vdash t : U \ltimes T \quad 1\uparrow \cdot \Pi_2, x :_{1v} T \vdash t' : T'}{\Pi_1 + \Pi_2 \vdash t \triangleright \text{map } x \mapsto t' : U \ltimes T'}$		$\text{TY-TERM-TOA} \quad \frac{\Pi \vdash u : U}{\Pi \vdash \text{to}_\ltimes u : U \ltimes 1}$	
		$\text{TY-TERM-FROMA} \quad \frac{\Pi \vdash t : U \ltimes (!_{1\infty} T)}{\Pi \vdash \text{from}_\ltimes t : U \otimes (!_{1\infty} T)}$	
$\text{TY-TERM-FILLU} \quad \frac{\Pi \vdash t : [1]^n}{\Pi \vdash t \triangleleft () : 1}$	$\text{TY-TERM-FILLL} \quad \frac{\Pi \vdash t : [T_1 \oplus T_2]^n}{\Pi \vdash t \triangleleft \text{Inl} : [T_1]^n}$	$\text{TY-TERM-FILLR} \quad \frac{\Pi \vdash t : [T_1 \oplus T_2]^n}{\Pi \vdash t \triangleleft \text{Inr} : [T_2]^n}$	$\text{TY-TERM-FILLP} \quad \frac{\Pi \vdash t : [T_1 \otimes T_2]^n}{\Pi \vdash t \triangleleft (,) : [T_1]^n \otimes [T_2]^n}$
$\text{TY-TERM-FILLE} \quad \frac{\Pi \vdash t : [!_{n'} T]^n}{\Pi \vdash t \triangleleft \varepsilon_{n'} : [T]^{n \cdot n'}}$	$\text{TY-TERM-FILLF} \quad \frac{\Pi_1 \vdash t : [T_m \rightarrow U]^n \quad \Pi_2, x :_m T \vdash u : U}{\Pi_1 + (1\uparrow \cdot n) \cdot \Pi_2 \vdash t \triangleleft (\lambda x_m \mapsto u) : 1}$	$\text{TY-TERM-FILLC} \quad \frac{\Pi_1 \vdash t : [U]^n \quad \Pi_2 \vdash t' : U \ltimes T}{\Pi_1 + (1\uparrow \cdot n) \cdot \Pi_2 \vdash t \triangleleft t' : T}$	
$\boxed{\Pi \text{ }^s\vdash t : T}$		<i>(Typing of syntactic sugar forms)</i>	
$\text{TY-TERMS-ALLOC} \quad \frac{\text{DisposableOnly } \Pi}{\Pi \text{ }^s\vdash \text{alloc} : T \ltimes ([T]^{1v})}$		$\text{TY-TERMS-FROMA'} \quad \frac{\Pi \vdash t : T \ltimes 1}{\Pi \text{ }^s\vdash \text{from}'_\ltimes t : T}$	
		$\text{TY-TERMS-FILLLEAF} \quad \frac{\Pi_1 \vdash t : [T]^n \quad \Pi_2 \vdash t' : T}{\Pi_1 + (1\uparrow \cdot n) \cdot \Pi_2 \text{ }^s\vdash t \triangleleft t' : 1}$	
$\text{TY-TERMS-F} \quad \frac{\Pi_2, x :_m T \vdash u : U}{\Pi_2 \text{ }^s\vdash \text{ }^s\lambda x_m \mapsto u : T_m \rightarrow U}$		$\text{TY-TERMS-L} \quad \frac{\Pi_2 \vdash t : T_1}{\Pi_2 \text{ }^s\vdash \text{ }^s\text{Inl } t : T_1 \oplus T_2}$	
		$\text{TY-TERMS-R} \quad \frac{\Pi_2 \vdash t : T_2}{\Pi_2 \text{ }^s\vdash \text{ }^s\text{Inr } t : T_1 \oplus T_2}$	
$\text{TY-TERMS-E} \quad \frac{\Pi_2 \vdash t : T}{m \cdot \Pi_2 \text{ }^s\vdash \text{ }^s\varepsilon_m t : !_m T}$		$\text{TY-TERMS-P} \quad \frac{\Pi_{21} \vdash t_1 : T_1 \quad \Pi_{22} \vdash t_2 : T_2}{\Pi_{21} + \Pi_{22} \text{ }^s\vdash \text{ }^s(t_1, t_2) : T_1 \otimes T_2}$	

$$\boxed{\Delta \vdash C : \mathbb{T} \multimap \mathbb{U}_0}$$

(Typing of evaluation contexts)

$$\begin{array}{c}
\text{TY-ECTXS-ID} \\
\hline
\vdash \square : \mathbb{U}_0 \multimap \mathbb{U}_0
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-APPFOC1} \\
\frac{\mathbb{m}\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_2 \vdash t' : \mathbb{T} \multimap \mathbb{U}}{\Delta_1 \vdash C \circ (\square \triangleright t') : \mathbb{T} \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-APPFOC2} \\
\frac{\mathbb{m}\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_1 \vdash v : \mathbb{T}}{\Delta_2 \vdash C \circ (v \triangleright \square) : (\mathbb{T} \multimap \mathbb{U}) \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-PATUFoc} \\
\frac{\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_2 \vdash u : \mathbb{U}}{\Delta_1 \vdash C \circ (\square ; u) : \mathbb{1} \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-PATSFoc} \\
\frac{\mathbb{m}\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_2, x_1 : \mathbb{m}\mathbb{T}_1 \vdash u_1 : \mathbb{U} \quad \Delta_2, x_2 : \mathbb{m}\mathbb{T}_2 \vdash u_2 : \mathbb{U}}{\Delta_1 \vdash C \circ (\square \triangleright \text{case}_{\mathbb{m}} \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}) : (\mathbb{T}_1 \oplus \mathbb{T}_2) \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-PATPFoc} \\
\frac{\mathbb{m}\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_2, x_1 : \mathbb{m}\mathbb{T}_1, x_2 : \mathbb{m}\mathbb{T}_2 \vdash u : \mathbb{U}}{\Delta_1 \vdash C \circ (\square \triangleright \text{case}_{\mathbb{m}} (x_1, x_2) \mapsto u) : (\mathbb{T}_1 \otimes \mathbb{T}_2) \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-PATEFoc} \\
\frac{\mathbb{m}\Delta_1, \Delta_2 \vdash C : \mathbb{U} \multimap \mathbb{U}_0 \quad \Delta_2, x : \mathbb{m}\mathbb{m}'\mathbb{T} \vdash u : \mathbb{U}}{\Delta_1 \vdash C \circ (\square \triangleright \text{case}_{\mathbb{m}} \mathbb{E}_{\mathbb{m}'} x \mapsto u) : \mathbb{!}_{\mathbb{m}'}\mathbb{T} \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-MAPFoc} \\
\frac{\Delta_1, \Delta_2 \vdash C : \mathbb{U} \times \mathbb{T}' \multimap \mathbb{U}_0 \quad \mathbb{1}\uparrow\Delta_2, x : \mathbb{!}_v\mathbb{T} \vdash t' : \mathbb{T}'}{\Delta_1 \vdash C \circ (\square \triangleright \text{map } x \mapsto t') : (\mathbb{U} \times \mathbb{T}) \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-TOAFoc} \\
\frac{\Delta \vdash C : (\mathbb{U} \times \mathbb{1}) \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\text{to}_{\times} \square) : \mathbb{U} \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-FROMAFoc} \\
\frac{\Delta \vdash C : (\mathbb{U} \otimes (\mathbb{!}_{\infty} \mathbb{T})) \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\text{from}_{\times} \square) : (\mathbb{U} \times (\mathbb{!}_{\infty} \mathbb{T})) \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-FILLUFoc} \\
\frac{\Delta \vdash C : \mathbb{1} \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\square \triangleleft ()) : [\mathbb{1}]^n \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-FILLLFoc} \\
\frac{\Delta \vdash C : [\mathbb{T}_1]^n \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\square \triangleleft \text{Inl}) : [\mathbb{T}_1 \oplus \mathbb{T}_2]^n \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-FILLRFoc} \\
\frac{\Delta \vdash C : [\mathbb{T}_2]^n \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\square \triangleleft \text{Inr}) : [\mathbb{T}_1 \oplus \mathbb{T}_2]^n \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-FILLPFoc} \\
\frac{\Delta \vdash C : ([\mathbb{T}_1]^n \otimes [\mathbb{T}_2]^n) \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\square \triangleleft (,)) : [\mathbb{T}_1 \otimes \mathbb{T}_2]^n \multimap \mathbb{U}_0}
\end{array}$$

$$\begin{array}{c}
\text{TY-ECTXS-FILLEFoc} \\
\frac{\Delta \vdash C : [\mathbb{T}]^{\mathbb{m}\cdot n} \multimap \mathbb{U}_0}{\Delta \vdash C \circ (\square \triangleleft \mathbb{E}_{\mathbb{m}}) : [\mathbb{!}_{\mathbb{m}}\mathbb{T}]^n \multimap \mathbb{U}_0}
\end{array}
\quad
\begin{array}{c}
\text{TY-ECTXS-FILLFFoc} \\
\frac{\Delta_1, (\mathbb{1}\uparrow n) \cdot \Delta_2 \vdash C : \mathbb{1} \multimap \mathbb{U}_0 \quad \Delta_2, x : \mathbb{m}\mathbb{T} \vdash u : \mathbb{U}}{\Delta_1 \vdash C \circ (\square \triangleleft (\lambda x_{\mathbb{m}} \mapsto u)) : [\mathbb{T}_{\mathbb{m}} \multimap \mathbb{U}]^n \multimap \mathbb{U}_0}
\end{array}$$

TY-ECTXS-FILLCFoc1

$$\frac{\Delta_1, (1\uparrow \cdot n) \cdot \Delta_2 \vdash C : \mathbb{T} \multimap \mathbb{U}_0 \quad \Delta_2 \vdash t' : \mathbb{U} \ltimes \mathbb{T}}{\Delta_1 \vdash C \circ (\Box \blacktriangleleft t') : [\mathbb{U}]^n \multimap \mathbb{U}_0}$$

TY-ECTXS-FILLCFoc2

$$\frac{\Delta_1, (1\uparrow \cdot n) \cdot \Delta_2 \vdash C : \mathbb{T} \multimap \mathbb{U}_0 \quad \Delta_1 \vdash v : [\mathbb{U}]^n}{\Delta_2 \vdash C \circ (v \blacktriangleleft \Box) : \mathbb{U} \ltimes \mathbb{T} \multimap \mathbb{U}_0}$$

TY-ECTXS-AOPENFOC

$$\frac{\begin{array}{c} \textcolor{red}{hvars}(C) \ \#\# \ \textcolor{red}{hvars}(-\Delta_3) \\ \text{LinOnly } \Delta_3 \\ \text{FinAgeOnly } \Delta_3 \\ \Delta_1, \Delta_2 \vdash C : (\mathbb{U} \ltimes \mathbb{T}') \multimap \mathbb{U}_0 \\ \Delta_2, -\Delta_3 \Vdash v_2 : \mathbb{U} \end{array}}{1\uparrow \Delta_1, \Delta_3 \vdash C \circ (\overset{\text{op}}{\textcolor{red}{hvars}}(-\Delta_3) \langle v_2, \Box \rangle) : \mathbb{T}' \multimap \mathbb{U}_0}$$

$$\boxed{\vdash C[t] : \mathbb{T}}$$

(Typing of extended terms (pair of evaluation context and term))

TY-ETERM-CLOSEDETERM

$$\frac{\Delta \vdash C : \mathbb{T} \multimap \mathbb{U}_0 \quad \Delta \vdash t : \mathbb{T}}{\vdash C[t] : \mathbb{U}_0}$$

3 SMALL-STEP SEMANTICS

$$\boxed{C[t] \longrightarrow C'[t']}$$

(Small-step evaluation of terms using evaluation contexts)

$$\frac{\text{SEM-ETERM-APPFOC1} \quad \text{NotVal } t}{C[t \triangleright t'] \longrightarrow (C \circ (\Box \triangleright t'))[t]}$$

$$\frac{\text{SEM-ETERM-APPUNFOC1}}{(C \circ (\Box \triangleright t'))[v] \longrightarrow C[v \triangleright t']}$$

$$\frac{\text{SEM-ETERM-APPFOC2} \quad \text{NotVal } t'}{C[v \triangleright t'] \longrightarrow (C \circ (v \triangleright \Box))[t']}$$

$$\frac{\text{SEM-ETERM-APPUNFOC2}}{(C \circ (v \triangleright \Box))[v'] \longrightarrow C[v \triangleright v']}$$

$$\frac{\text{SEM-ETERM-APPRED}}{C[v \triangleright (\lambda x_m \mapsto u)] \longrightarrow C[u[x := v]]}$$

$$\frac{\text{SEM-ETERM-PATUFoc} \quad \text{NotVal } t}{C[t; u] \longrightarrow (C \circ (\Box; u))[t]}$$

$$\frac{\text{SEM-ETERM-PATUNFOC}}{(C \circ (\Box; u))[v] \longrightarrow C[v; u]}$$

$$\frac{\text{SEM-ETERM-PATURED}}{C[(); u] \longrightarrow C[u]}$$

SEM-ETERM-PATSFOC

$$\frac{\text{NotVal } t}{C[t \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}] \longrightarrow (C \circ (\Box \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}))[t]}$$

SEM-ETERM-PATSUNFOC

$$(C \circ (\Box \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}))[v] \longrightarrow C[v \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}]$$

SEM-ETERM-PATLRED

$$C[(\text{Inl } v_1) \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}] \longrightarrow C[u_1[x_1 := v_1]]$$

SEM-ETERM-PATRRED

$$C[(\text{Inr } v_2) \triangleright \text{case}_m \{ \text{Inl } x_1 \mapsto u_1, \text{Inr } x_2 \mapsto u_2 \}] \longrightarrow C[u_2[x_2 := v_2]]$$

SEM-ETERM-PATPFoc

$$\frac{\text{NotVal } t}{C[t \triangleright \text{case}_m (x_1, x_2) \mapsto u] \longrightarrow (C \circ (\Box \triangleright \text{case}_m (x_1, x_2) \mapsto u))[t]}$$

SEM-ETERM-PATPUNFOC

$$(C \circ (\Box \triangleright \text{case}_m (x_1, x_2) \mapsto u))[v] \longrightarrow C[v \triangleright \text{case}_m (x_1, x_2) \mapsto u]$$

SEM-ETERM-PATPREd

$$C[(v_1, v_2) \triangleright \text{case}_m (x_1, x_2) \mapsto u] \longrightarrow C[u[x_1 := v_1][x_2 := v_2]]$$

SEM-ETERM-PATEFoc

$$\frac{\text{NotVal } t}{C[t \triangleright \text{case}_m E_n x \mapsto u] \longrightarrow (C \circ (\Box \triangleright \text{case}_m E_n x \mapsto u))[t]}$$

SEM-ETERM-PATeUNFOC

$$\frac{}{(C \circ (\Box \triangleright \text{case}_{\mathfrak{m}} \mathfrak{E}_{\mathfrak{n}} \mathfrak{x} \mapsto u)) [v] \longrightarrow C[v \triangleright \text{case}_{\mathfrak{m}} \mathfrak{E}_{\mathfrak{n}} \mathfrak{x} \mapsto u]}$$

SEM-ETERM-PATeRED

$$\frac{}{C[\mathfrak{E}_{\mathfrak{n}} v' \triangleright \text{case}_{\mathfrak{m}} \mathfrak{E}_{\mathfrak{n}} \mathfrak{x} \mapsto u] \longrightarrow C[u[\mathfrak{x} := v']]}$$

SEM-ETERM-MAPFOC

NotVal t

$$\frac{}{C[t \triangleright \text{map } \mathfrak{x} \mapsto t'] \longrightarrow (C \circ (\Box \triangleright \text{map } \mathfrak{x} \mapsto t')) [t]}$$

SEM-ETERM-MAPUNFOC

$$\frac{}{(C \circ (\Box \triangleright \text{map } \mathfrak{x} \mapsto t')) [v] \longrightarrow C[v \triangleright \text{map } \mathfrak{x} \mapsto t']}$$

SEM-ETERM-MAPREDaOPENFOC

$$h' = \text{max}(hvars(C)) + 1$$

$$\frac{}{C[\mathfrak{H} \langle v_2, v_1 \rangle \triangleright \text{map } \mathfrak{x} \mapsto t'] \longrightarrow (C \circ (\mathfrak{H}_{\mathfrak{H}h'}^{\text{op}} \langle v_2 [\mathfrak{H}h'], \Box \rangle)) [t'[\mathfrak{x} := v_1 [\mathfrak{H}h']]]}$$

SEM-ETERM-AOPENUNFOC

$$\frac{}{(C \circ \mathfrak{H}_{\mathfrak{H}}^{\text{op}} \langle v_2, \Box \rangle) [v_1] \longrightarrow C[\mathfrak{H} \langle v_2, v_1 \rangle]}$$

SEM-ETERM-ToAFoc

NotVal u

$$\frac{}{C[\text{to}_{\mathfrak{x}} u] \longrightarrow (C \circ (\text{to}_{\mathfrak{x}} \Box)) [u]}$$

SEM-ETERM-ToAUNFOC

$$\frac{}{(C \circ (\text{to}_{\mathfrak{x}} \Box)) [v_2] \longrightarrow C[\text{to}_{\mathfrak{x}} v_2]}$$

SEM-ETERM-ToAREd

$$\frac{}{C[\text{to}_{\mathfrak{x}} v_2] \longrightarrow C[\{\} \langle v_2, () \rangle]}$$

SEM-ETERM-FROMaFOC

NotVal t

$$\frac{}{C[\text{from}_{\mathfrak{x}} t] \longrightarrow (C \circ (\text{from}_{\mathfrak{x}} \Box)) [t]}$$

SEM-ETERM-FROMaUNFOC

$$\frac{}{(C \circ (\text{from}_{\mathfrak{x}} \Box)) [v] \longrightarrow C[\text{from}_{\mathfrak{x}} v]}$$

SEM-ETERM-FROMAREd

$$\frac{}{C[\text{from}_{\mathfrak{x}} \{\} \langle v_2, \mathfrak{E}_{\text{Inf}} v_1 \rangle] \longrightarrow C[(v_2, \mathfrak{E}_{\text{Inf}} v_1)]}$$

SEM-ETERM-FILLUFOC

NotVal t

$$\frac{}{C[t \triangleleft ()] \longrightarrow (C \circ (\Box \triangleleft ())) [t]}$$

SEM-ETERM-FILLUUNFOC

$$\frac{}{(C \circ (\Box \triangleleft ())) [v] \longrightarrow C[v \triangleleft ()]}$$

SEM-ETERM-FILLURED

$$\frac{}{C[+h \triangleleft ()] \longrightarrow C[h :=_{\{\}} ()][()]}$$

SEM-ETERM-FILLFOC

NotVal t

$$\frac{}{C[t \triangleleft \text{Inl}] \longrightarrow (C \circ (\Box \triangleleft \text{Inl})) [t]}$$

SEM-ETERM-FILLUNFOC

$$\frac{}{(C \circ (\Box \triangleleft \text{Inl})) [v] \longrightarrow C[v \triangleleft \text{Inl}]}$$

SEM-ETERM-FILLRED

$$h' = \text{max}(hvars(C) \cup \{h\}) + 1$$

$$\frac{}{C[+h \triangleleft \text{Inl}] \longrightarrow C[h :=_{\{h'+1\}} \text{Inl } -(h'+1)][+(h'+1)]}$$

SEM-ETERM-FILLRFoc

NotVal t

$$\frac{}{C[t \triangleleft \text{Inr}] \longrightarrow (C \circ (\Box \triangleleft \text{Inr})) [t]}$$

SEM-ETERM-FILLRUNFOC

$$\frac{}{(C \circ (\Box \triangleleft \text{Inr})) [v] \longrightarrow C[v \triangleleft \text{Inr}]}$$

SEM-ETERM-FILLRRED

$$h' = \text{max}(hvars(C) \cup \{h\}) + 1$$

$$\frac{}{C[+h \triangleleft \text{Inr}] \longrightarrow C[h :=_{\{h'+1\}} \text{Inr } -(h'+1)][+(h'+1)]}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLEFOC} \\
\text{NotVal } t \\
\hline
C[t \triangleleft E_m] \longrightarrow (C \circ (\Box \triangleleft E_m))[t]
\end{array}
\qquad
\begin{array}{c}
\text{SEM-ETERM-FILLEUNFOC} \\
\hline
(C \circ (\Box \triangleleft E_m))[v] \longrightarrow C[v \triangleleft E_m]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLERED} \\
h' = \text{max}(hvars(C) \cup \{h\}) + 1 \\
\hline
C[+h \triangleleft E_m] \longrightarrow C[h :=_{\{h'+1\}} E_m - (h'+1)][+(h'+1)]
\end{array}
\qquad
\begin{array}{c}
\text{SEM-ETERM-FILLPFoc} \\
\text{NotVal } t \\
\hline
C[t \triangleleft (,)] \longrightarrow (C \circ (\Box \triangleleft (,)))[t]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLPUNFOC} \\
\hline
(C \circ (\Box \triangleleft (,)))[v] \longrightarrow C[v \triangleleft (,)]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLPREd} \\
h' = \text{max}(hvars(C) \cup \{h\}) + 1 \\
\hline
C[+h \triangleleft (,)] \longrightarrow C[h :=_{\{h'+1, h'+2\}} (- (h'+1), - (h'+2))][+(h'+1), +(h'+2)]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLFFoc} \\
\text{NotVal } t \\
\hline
C[t \triangleleft (\lambda x_m \mapsto u)] \longrightarrow (C \circ (\Box \triangleleft (\lambda x_m \mapsto u)))[t]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLFUNFOC} \\
\hline
(C \circ (\Box \triangleleft (\lambda x_m \mapsto u)))[v] \longrightarrow C[v \triangleleft (\lambda x_m \mapsto u)]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLFRED} \\
\hline
C[+h \triangleleft (\lambda x_m \mapsto u)] \longrightarrow C[h :=_{\{\}} \lambda x_m \mapsto u][()]
\end{array}
\qquad
\begin{array}{c}
\text{SEM-ETERM-FILLCFoc1} \\
\text{NotVal } t \\
\hline
C[t \triangleleft t'] \longrightarrow (C \circ (\Box \triangleleft t'))[t]
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLCUNFOC1} \\
\hline
(C \circ (\Box \triangleleft t'))[v] \longrightarrow C[v \triangleleft t']
\end{array}
\qquad
\begin{array}{c}
\text{SEM-ETERM-FILLCFoc2} \\
\text{NotVal } t' \\
\hline
C[v \triangleleft t'] \longrightarrow (C \circ (v \triangleleft \Box))[t']
\end{array}$$

$$\begin{array}{c}
\text{SEM-ETERM-FILLCUNFOC2} \\
\hline
(C \circ (v \triangleleft \Box))[v'] \longrightarrow C[v \triangleleft v']
\end{array}
\qquad
\begin{array}{c}
\text{SEM-ETERM-FILLCREd} \\
h' = \text{max}(hvars(C) \cup \{h\}) + 1 \\
\hline
C[+h \triangleleft_{\text{H}} \langle v_2, v_1 \rangle] \longrightarrow C[h :=_{(\text{H} \triangleleft h')} v_2 [H \triangleleft h']][v_1 [H \triangleleft h']]
\end{array}$$

4 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>