Destination λ-calculus

Thomas Bagrel

December 8, 2023

1 Term and value syntax

termvar, x, y, d	Term-level variable
holevar, h	Hole

term_value, v	::=		Term value
	$\mid$	$\langle v_1 , \overline{v_2} \rangle_H$	Ampar
	$\mid$	$@h$	Destination
	$\mid$	$()$	Unit
	$\mid$	$\mathsf{Inl}\, v$	Left variant for sum
	$\mid$	$\mathsf{Inr}\, v$	Right variant for sum
	$\mid$	(v_1 , v_2)	Product
	$\mid$	$\}^m v$	Exponential
	$\mid$	$\lambda x . t$	Linear function

$\overline{\text{extended\_value}}$, $\overline{v}$	::=		Pseudo-value that may contain holes
	$\mid$	v	Term value
	$\mid$	h	Hole
	$\mid$	$\mathsf{Inl}\, \overline{v}$	Left variant with val or hole
	$\mid$	$\mathsf{Inr}\, \overline{v}$	Right variant with val or hole
	$\mid$	$(\overline{v_1} , \overline{v_2})$	Product with val or hole
	$\mid$	$\}^m \overline{v}$	Exponential with val or hole

term, t, u	::=		Term
	$\mid$	v	Term value
	$\mid$	x	Variable
	$\mid$	$t \succ u$	Application
	$\mid$	$t \succ \mathsf{case}\, () \mapsto u$	Pattern-match on unit
	$\mid$	$t \succ \mathsf{case}\, \{ \mathsf{Inl}\, x_1 \mapsto u_1 , \mathsf{Inr}\, x_2 \mapsto u_2 \}$	Pattern-match on sum
	$\mid$	$t \succ \mathsf{case}\, (x_1 , x_2) \mapsto u$	Pattern-match on product
	$\mid$	$t \succ \mathsf{case}\, \}^m x \mapsto u$	Pattern-match on exponential
	$\mid$	$t \succ \mathsf{mapL}\, x \mapsto u$	Map over the left side of the ampar
	$\mid$	$\mathsf{to}_{\bowtie}\, t$	Wrap t into a trivial ampar
	$\mid$	$\mathsf{from}_{\bowtie}\, t$	Extract value from trivial ampar
	$\mid$	alloc_A	Return a fresh "identity" ampar object
	$\mid$	$t \triangleleft ()$	Fill destination with unit
	$\mid$	$t \triangleleft \mathsf{Inl}$	Fill destination with left variant
	$\mid$	$t \triangleleft \mathsf{Inr}$	Fill destination with right variant
	$\mid$	$t \triangleleft (,)$	Fill destination with product constructor
	$\mid$	$t \triangleleft \}^m$	Fill destination with exponential constructor
	$\mid$	$t \triangleleft\bullet u$	Fill destination with root of ampar u

2 Type system

type, $\mathbf{A}$, $\mathbf{B}$	::=		Type
	|	$\mathbf{1}$	Unit
	|	$\mathbf{A}_1 \oplus \mathbf{A}_2$	Sum
	|	$\mathbf{A}_1 \otimes \mathbf{A}_2$	Product
	|	$!^m \mathbf{A}$	Exponential
	|	$\mathbf{A}_1 \rtimes \mathbf{A}_2$	Ampar type (consuming $\mathbf{A}_1$ yields $\mathbf{A}_2$)
	|	$\mathbf{A}_1 \ {}_m\!\!\rightarrow \mathbf{A}_2$	Linear function
	|	${}^m\lfloor \mathbf{A} \rfloor$	Destination
multiplicity, m, n	::=		Multiplicity (Semiring with product $\cdot$)
	|	ν	Born now. Identity of the product
	|	$\uparrow$	One scope older
	|	∞	Infinitely old / static. Absorbing for product
	|	$m_1 \cdot m_2$	Semiring product
typing_context, Δ	::=		Typing context
	|	Γ	
	|	H	
	|	$\Gamma \uplus \mathrm{H}$	
	|	$m \cdot \Delta$	Increase age of bindings by m
pos_context, Γ	::=		Positive typing context
	|	{pos_assign∗}	
	|	$\Gamma_1 \sqcup \Gamma_2$	
	|	$\Gamma^{@}$	Positive context restricted to destinations only
	|	$m \cdot \Gamma$	Increase age of bindings by m
pos_assign	::=		Positive type assignment
	|	$\mathtt{x} :_m \mathbf{A}$	Variable
	|	$@\mathtt{h} :_m {}^n\lfloor \mathbf{A} \rfloor$	Destination (m is its own age; n is the age of values it accepts)
neg_context, H	::=		Negative typing context
	|	{neg_assign∗}	
	|	$\mathrm{H}_1 \sqcup \mathrm{H}_2$	
	|	$@^{-1}\Gamma$	Invert the sign of the context
	|	$m \cdot \mathrm{H}$	Increase age of bindings by m
neg_assign	::=		Negative type assignment
	|	$\mathtt{h} :^n \mathbf{A}$	Hole (n is the age of values it accepts, its own age is undefined)

$\boxed{\mathrm{H}_1 = \mathrm{H}_2}$ *($@^{-1}$: "Inverse sign of context" operation)*

AtApp-Empty

$$\frac{}{@^{-1}\emptyset = \emptyset}$$

AtApp-Rec

$$\frac{}{@^{-1}(\{@\mathtt{h} :_m {}^n\lfloor \mathbf{A} \rfloor\} \sqcup \Gamma) = \{\mathtt{h} :^{m \cdot n} \mathbf{A}\} \sqcup @^{-1}\Gamma}$$

$\boxed{\Delta \Vdash \mathsf{e}}$ *(Typing of effects (require both positive and negative contexts))*

TyEff-NoEff

$$\frac{}{\emptyset \uplus \emptyset \Vdash \varepsilon}$$

TyEff-Single

$$\frac{\Gamma \uplus \mathrm{H} \Vdash \overline{\mathsf{v}} : \mathbf{A} \qquad \mathtt{h} \notin \mathbf{names}(\Gamma)}{m \cdot ((n \cdot \uparrow) \cdot \Gamma \sqcup \{@\mathtt{h} :_\nu {}^n\lfloor \mathbf{A} \rfloor\} \uplus {}^{n \cdot}\mathrm{H}) \Vdash \mathtt{h} := \overline{\mathsf{v}}}$$

TyEff-Union

$$\frac{\begin{array}{c}\Gamma_1 \uplus \mathrm{H}_1 \sqcup @^{-1}\Gamma_{22} \Vdash \mathsf{e}_1 \\ \Gamma_{21} \sqcup \Gamma_{22} \uplus \mathrm{H}_2 \Vdash \mathsf{e}_2 \\ \mathbf{names}(\Gamma_1 \uplus \mathrm{H}_1) \cap \mathbf{names}(\Gamma_{21} \uplus \mathrm{H}_2) = \emptyset\end{array}}{\Gamma_1 \sqcup \Gamma_{21} \uplus \mathrm{H}_1 \sqcup \mathrm{H}_2 \Vdash \mathsf{e}_1 \cdot \mathsf{e}_2}$$

$\boxed{\Gamma \vdash \mathsf{v} \mid \mathsf{e} : \mathbf{A}}$ *(Typing of commands (only a positive context is needed))*

TyCmd-Cmd

$$\frac{\begin{array}{c}\Gamma_{11} \sqcup \Gamma_{12} \vdash \mathsf{v} : \mathbf{A} \\ \Gamma_2 \uplus @^{-1}\Gamma_{12} \Vdash \mathsf{e} \\ \mathbf{names}(\Gamma_{11}) \cap \mathbf{names}(\Gamma_2) = \emptyset\end{array}}{\Gamma_{11} \sqcup \Gamma_2 \vdash \mathsf{v} \mid \mathsf{e} : \mathbf{A}}$$

$\boxed{\Delta \Vdash \overline{v} : A}$ *(Typing of extended values (require both positive and negative contexts))*

TyValExt-Hole
$$\frac{}{\emptyset \uplus \{h :^{\nu} A\} \Vdash h : A}$$

TyValExt-Dest
$$\frac{}{\{@h :_{\nu} {}^{n}\lfloor A \rfloor\} \uplus \emptyset \Vdash @h : {}^{n}\lfloor A \rfloor}$$

TyValExt-Unit
$$\frac{}{\emptyset \uplus \emptyset \Vdash () : 1}$$

TyValExt-Inl
$$\frac{\Gamma \uplus H \Vdash \overline{v} : A_1}{\Gamma \uplus H \Vdash \mathsf{Inl}\,\overline{v} : A_1 \oplus A_2}$$

TyValExt-Inr
$$\frac{\Gamma \uplus H \Vdash \overline{v} : A_2}{\Gamma \uplus H \Vdash \mathsf{Inr}\,\overline{v} : A_1 \oplus A_2}$$

TyValExt-Prod
$$\frac{\Gamma_1 \uplus H_1 \Vdash \overline{v_1} : A_1 \quad \Gamma_2 \uplus H_2 \Vdash \overline{v_2} : A_2 \quad \mathsf{names}(\Gamma_1 \uplus H_1) \cap \mathsf{names}(\Gamma_2 \uplus H_2) = \emptyset}{\Gamma_1 \sqcup \Gamma_2 \uplus H_1 \sqcup H_2 \Vdash (\overline{v_1}, \overline{v_2}) : A_1 \otimes A_2}$$

TyValExt-Exp
$$\frac{\Gamma \uplus H \Vdash \overline{v} : A}{m\cdot\Gamma \uplus {}^{m}\cdot H \Vdash \rangle^{m}\,\overline{v} : !^{m}\,A}$$

TyValExt-Ampar
$$\frac{\Gamma_1 \uplus \emptyset \Vdash v_1 : A_1 \quad \Gamma_2 \uplus @^{-1}\Gamma_1 \Vdash \overline{v_2} : A_2}{\Gamma_2 \uplus \emptyset \Vdash \langle v_1 \,,\, \overline{v_2} \rangle_H : A_1 \rtimes A_2}$$

TyValExt-Lambda
$$\frac{\Gamma \sqcup \{x :_m A_1\} \vdash t : A_2}{\Gamma \uplus \emptyset \Vdash \lambda x.\, t : A_1 \xrightarrow{m} A_2}$$

$\boxed{\Gamma \vdash t : A}$ *(Typing of terms (only a positive context is needed))*

TyTerm-Val
$$\frac{\Gamma \uplus \emptyset \Vdash v : A}{\Gamma \vdash v : A}$$

TyTerm-VarNow
$$\frac{}{\{x :_{\nu} A\} \vdash x : A}$$

TyTerm-VarInf
$$\frac{}{\{x :_{\infty} A\} \vdash x : A}$$

TyTerm-App
$$\frac{\Gamma_1 \vdash t : A_1 \quad \Gamma_2 \vdash u : A_1 \xrightarrow{m} A_2 \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{m\cdot\Gamma_1 \sqcup \Gamma_2 \vdash t \succ u : A_2}$$

TyTerm-PatUnit
$$\frac{\Gamma_1 \vdash t : 1 \quad \Gamma_2 \vdash u : B \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{\Gamma_1 \sqcup \Gamma_2 \vdash t \succ \mathsf{case}\,() \mapsto u : B}$$

TyTerm-PatSum
$$\frac{\Gamma_1 \vdash t : A_1 \oplus A_2 \quad \Gamma_2 \sqcup \{x_1 :_m A_1\} \vdash u_1 : B \quad \Gamma_2 \sqcup \{x_2 :_m A_2\} \vdash u_2 : B \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{m\cdot\Gamma_1 \sqcup \Gamma_2 \vdash t \succ \mathsf{case}\,\{\,\mathsf{Inl}\,x_1 \mapsto u_1\,,\ \mathsf{Inr}\,x_2 \mapsto u_2\,\} : B}$$

TyTerm-PatProd
$$\frac{\Gamma_1 \vdash t : A_1 \otimes A_2 \quad \Gamma_2 \sqcup \{x_1 :_m A_1, x_2 :_m A_2\} \vdash u : B \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{m\cdot\Gamma_1 \sqcup \Gamma_2 \vdash t \succ \mathsf{case}\,(x_1\,,\,x_2) \mapsto u : B}$$

TyTerm-PatExp
$$\frac{\Gamma_1 \vdash t : !^{m'}\,A \quad \Gamma_2 \sqcup \{x :_{m\cdot m'} A_1\} \vdash u : B \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{m\cdot\Gamma_1 \sqcup \Gamma_2 \vdash t \succ \mathsf{case}\,\rangle^{m'}\,x \mapsto u : B}$$

TyTerm-MapAmpar
$$\frac{\Gamma_1 \vdash t : A_1 \rtimes A_2 \quad \uparrow\cdot\Gamma_2 \sqcup \{x :_{\nu} A_1\} \vdash u : B \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{\Gamma_1 \sqcup \Gamma_2 \vdash t \succ \mathsf{mapL}\,x \mapsto u : B \rtimes A_2}$$

TyTerm-FillComp
$$\frac{\Gamma_1 \vdash t : {}^{n}\lfloor A_2 \rfloor \quad \Gamma_2 \vdash u : A_1 \rtimes A_2 \quad \mathsf{names}(\Gamma_1) \cap \mathsf{names}(\Gamma_2) = \emptyset}{\Gamma_1 \sqcup (n\cdot\uparrow)\cdot\Gamma_2 \vdash t \triangleleft_{\bullet} u : A_1}$$

TyTerm-FillUnit
$$\frac{\Gamma \vdash t : {}^{n}\lfloor 1 \rfloor}{\Gamma \vdash t \triangleleft () : 1}$$

TyTerm-FillInl
$$\frac{\Gamma \vdash t : {}^{n}\lfloor A_1 \oplus A_2 \rfloor}{\Gamma \vdash t \triangleleft \mathsf{Inl} : {}^{n}\lfloor A_1 \rfloor}$$

TyTerm-FillInr
$$\frac{\Gamma \vdash t : {}^{n}\lfloor A_1 \oplus A_2 \rfloor}{\Gamma \vdash t \triangleleft \mathsf{Inr} : {}^{n}\lfloor A_2 \rfloor}$$

TyTerm-FillProd
$$\frac{\Gamma \vdash t : {}^{n}\lfloor A_1 \otimes A_2 \rfloor}{\Gamma \vdash t \triangleleft (,) : {}^{n}\lfloor A_1 \rfloor \otimes {}^{n}\lfloor A_2 \rfloor}$$

TyTerm-FillExp
$$\frac{\Gamma \vdash t : {}^{n}\lfloor !^{n'}\,A \rfloor}{\Gamma \vdash t \triangleleft \rangle^{n'} : {}^{n\cdot n'}\lfloor A \rfloor}$$

TyTerm-Alloc
$$\frac{}{\emptyset \vdash \mathsf{alloc}_A : {}^{\nu}\lfloor A \rfloor \rtimes A}$$

TyTerm-ToAmpar
$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{to}_{\rtimes}\, t : 1 \rtimes A}$$

TyTerm-FromAmpar
$$\frac{\Gamma \vdash t : 1 \rtimes A}{\Gamma \vdash \mathsf{from}_{\rtimes}\, t : A}$$

3 Effects and big-step semantics

$$
\begin{array}{llll}
\text{effect},\ e & ::= & & \text{Effect} \\
& \mid & \varepsilon & \text{No effect} \\
& \mid & h := \overline{v} & \\
& \mid & e_1 \cdot e_2 &
\end{array}
$$

$\boxed{\mathsf{eff\_app}_1 = \mathsf{eff\_app}_2}$ *(**apply**: how effects are applied locally or winded up (we assume effect lists are ε-terminated))*

$$\textsc{EffApp-NoEff}\quad \frac{}{\mathbf{apply}\,(\varepsilon,\ \overline{v}_H) = \varepsilon,\ \overline{v}_H}$$

$$\textsc{EffApp-WindUp}\quad \frac{h \notin \mathbf{names}\,(H)}{\mathbf{apply}\,(h := \overline{v_2} \cdot e,\ \overline{v_1}_H) = h := \overline{v_2} \mathbin{\hat{\cdot}} \mathbf{apply}\,(e,\ \overline{v_1}_H)}$$

$$\textsc{EffApp-Fill}\quad \frac{\_ \sqcup H' \Vdash \overline{v_2} : A \qquad \mathbf{names}(H \sqcup \{h :^n A\}) \cap \mathbf{names}(H') = \emptyset}{\mathbf{apply}\,(h := \overline{v_2} \cdot e,\ \overline{v_1}_{H \sqcup \{h:^n A\}}) = \mathbf{apply}\,(e,\ \overline{v_1[h := \overline{v_2}]}_{H \sqcup^n H'})}$$

$\boxed{t \Downarrow v \mid e}$ *(Big-step evaluation into commands)*

$$\textsc{BigStep-Val}\quad \frac{}{v \Downarrow v \mid \varepsilon}$$

$$\textsc{BigStep-App}\quad \frac{t_1 \Downarrow v_1 \mid e_1 \qquad t_2 \Downarrow \lambda x.\,u \mid e_2 \qquad u[x := v_1] \Downarrow v_3 \mid e_3}{t_1 \succ t_2 \Downarrow v_3 \mid e_1 \cdot e_2 \cdot e_3}$$

$$\textsc{BigStep-PatUnit}\quad \frac{t_1 \Downarrow () \mid e_1 \qquad t_2 \Downarrow v_2 \mid e_2}{t_1 \succ \mathbf{case}\,() \mapsto t_2 \Downarrow v_2 \mid e_1 \cdot e_2}$$

$$\textsc{BigStep-PatInl}\quad \frac{t \Downarrow \mathsf{Inl}\,v_1 \mid e_1 \qquad u_1[x_1 := v_1] \Downarrow v_2 \mid e_2}{t \succ \mathbf{case}\,\{\,\mathsf{Inl}\,x_1 \mapsto u_1\,,\ \mathsf{Inr}\,x_2 \mapsto u_2\,\} \Downarrow v_2 \mid e_1 \cdot e_2}$$

$$\textsc{BigStep-PatInr}\quad \frac{t \Downarrow \mathsf{Inr}\,v_1 \mid e_1 \qquad u_2[x_2 := v_1] \Downarrow v_2 \mid e_2}{t \succ \mathbf{case}\,\{\,\mathsf{Inl}\,x_1 \mapsto u_1\,,\ \mathsf{Inr}\,x_2 \mapsto u_2\,\} \Downarrow v_2 \mid e_1 \cdot e_2}$$

$$\textsc{BigStep-PatProd}\quad \frac{t \Downarrow (v_1\,,\ v_2) \mid e_1 \qquad u[x_1 := v_1, x_2 := v_2] \Downarrow v_2 \mid e_2}{t \succ \mathbf{case}\,(x_1\,,\ x_2) \mapsto u \Downarrow v_2 \mid e_1 \cdot e_2}$$

$$\textsc{BigStep-MapAmpar}\quad \frac{t \Downarrow \langle v_1\,,\ \overline{v_2}\rangle_H \mid e_1 \qquad u[x := v_1] \Downarrow v_3 \mid e_2 \qquad e_3,\ \overline{v_4}_{H'} = \mathbf{apply}\,(e_2,\ \overline{v_2}_H)}{t \succ \mathbf{mapL}\,x \mapsto u \Downarrow \langle v_3\,,\ \overline{v_4}\rangle_{H'} \mid e_1 \cdot e_3}$$

$$\textsc{BigStep-Alloc}\quad \frac{\mathbf{fresh}\,h}{\mathbf{alloc}_A \Downarrow \langle @h\,,\ h\rangle_{\{h:^\nu A\}} \mid \varepsilon}$$

$$\textsc{BigStep-ToAmpar}\quad \frac{t \Downarrow v \mid e}{\mathbf{to}_{\bowtie}\,t \Downarrow \langle (),\ v\rangle_\emptyset \mid e}$$

$$\textsc{BigStep-FromAmpar}\quad \frac{t \Downarrow \langle (),\ v\rangle_\emptyset \mid e}{\mathbf{from}_{\bowtie}\,t \Downarrow v \mid e}$$

$$\textsc{BigStep-FillUnit}\quad \frac{t \Downarrow @h \mid e}{t \triangleleft () \Downarrow () \mid e \cdot h := ()}$$

$$\textsc{BigStep-FillInl}\quad \frac{t \Downarrow @h \mid e \qquad \mathbf{fresh}\,h'}{t \triangleleft \mathsf{Inl} \Downarrow @h' \mid e \cdot h := \mathsf{Inl}\,h'}$$

$$\textsc{BigStep-FillInr}\quad \frac{t \Downarrow @h \mid e}{t \triangleleft \mathsf{Inr} \Downarrow @h' \mid e \cdot h := \mathsf{Inr}\,h'}$$

$$\textsc{BigStep-FillProd}\quad \frac{t \Downarrow @h \mid e \qquad \mathbf{fresh}\,h_1 \qquad \mathbf{fresh}\,h_2}{t \triangleleft (,) \Downarrow (@h_1\,,\ @h_2) \mid e \cdot h := (h_1\,,\ h_2)}$$

$$\textsc{BigStep-FillComp}\quad \frac{t \Downarrow @h \mid e_1 \qquad u \Downarrow \langle v_1\,,\ \overline{v_2}\rangle_H \mid e_2}{t \triangleleft\bullet\, u \Downarrow v_1 \mid e_1 \cdot e_2 \cdot h := \overline{v_2}}$$

4 Type safety

Theorem 1 (Type safety)**.** *If* $\Gamma^{@} \vdash \mathtt{t} : \mathbf{A}$ *then* $\mathtt{t} \Downarrow \mathtt{v} \mid \mathtt{e}$ *and* $\Gamma^{@} \vdash \mathtt{v} \mid \mathtt{e} : \mathbf{A}$.

Proof. By induction on the typing derivation.

- TyTerm_Val: (0) $\Gamma^{@} \vdash \mathtt{v} : \mathbf{A}$
  (0) gives (1) $\mathtt{v} \Downarrow \mathtt{v} \mid \varepsilon$ immediately. From TyEff_NoEff and TyCmd_Cmd we conclude (2) $\Gamma^{@} \vdash \mathtt{v} \mid \mathtt{e} : \mathbf{A}$.

- TyTerm_App: (0) $m{\cdot}\Gamma_1^{@} \sqcup \Gamma_2^{@} \vdash \mathtt{t} \succ \mathtt{u} : \mathbf{A}_2$
  We have
  (1) $\Gamma_1^{@} \vdash \mathtt{t} : \mathbf{A}_1$
  (2) $\Gamma_2^{@} \vdash \mathtt{u} : \mathbf{A}_1 \xrightarrow{m} \mathbf{A}_2$
  (3) $\mathbf{names}(\Gamma_1^{@}) \cap \mathbf{names}(\Gamma_2^{@}) = \emptyset$
  Using recursion hypothesis on (1) we get (4) $\mathtt{t} \Downarrow \mathtt{v}_1 \mid \mathtt{e}_1$ where (5) $\Gamma_1^{@} \vdash \mathtt{v}_1 \mid \mathtt{e}_1 : \mathbf{A}_1$.
  Inverting TyCmd_Cmd we get (5) $\Gamma_{11}^{@} \sqcup \Gamma_{13}^{@} \vdash \mathtt{v}_1 : \mathbf{A}_1$ and (6) $\Gamma_{12} \uplus @^{-1}\Gamma_{13}^{@} \Vdash \mathtt{e}_1$ where (7) $\Gamma_1^{@} = \Gamma_{11}^{@} \sqcup \Gamma_{12}^{@}$.
  Using recursion hypothesis on (2) we get (8) $\mathtt{u} \Downarrow \mathtt{v}_2 \mid \mathtt{e}_2$ where (9) $\Gamma_2^{@} \vdash \mathtt{v}_2 \mid \mathtt{e}_2 : \mathbf{A}_1 \xrightarrow{m} \mathbf{A}_2$.
  Inverting TyCmd_Cmd we get (10) $\Gamma_{21}^{@} \sqcup \Gamma_{23}^{@} \vdash \mathtt{v}_2 : \mathbf{A}_1 \xrightarrow{m} \mathbf{A}_2$ and (11) $\Gamma_{22} \uplus @^{-1}\Gamma_{23}^{@} \Vdash \mathtt{e}_2$ where (12) $\Gamma_2^{@} = \Gamma_{21}^{@} \sqcup \Gamma_{22}^{@}$.
  Using Lemma **??** on (9) we get (13) $\mathtt{v}_2 = \lambda \mathtt{x}.\mathtt{t}'$ and (14) $\Gamma_{21}^{@} \sqcup \Gamma_{23}^{@} \sqcup \{\mathtt{x} :_m \mathbf{A}_1\} \vdash \mathtt{t}' : \mathbf{A}_2$.

  Typing value part of the result

  Using Lemma **??** on (14) and (5) we get (15) $m{\cdot}(\Gamma_{11}^{@} \sqcup \Gamma_{13}^{@}) \sqcup (\Gamma_{21}^{@} \sqcup \Gamma_{23}^{@}) \vdash \mathtt{t}'[\mathtt{x} := \mathtt{v}_1] : \mathbf{A}_2$.
  Using recursion hypothesis on (15) we get (16) $\mathtt{t}'[\mathtt{x} := \mathtt{v}_1] \Downarrow \mathtt{v}_3 \mid \mathtt{e}_3$ where (17) $m{\cdot}(\Gamma_{11}^{@} \sqcup \Gamma_{13}^{@}) \sqcup (\Gamma_{21}^{@} \sqcup \Gamma_{23}^{@}) \vdash \mathtt{v}_3 \mid \mathtt{e}_3 : \mathbf{A}_2$.

  Typing effect part of the result

  We have
  (6) $\Gamma_{12}^{@} \uplus @^{-1}\Gamma_{13}^{@} \Vdash \mathtt{e}_1$
  (11) $\Gamma_{22}^{@} \uplus @^{-1}\Gamma_{23}^{@} \Vdash \mathtt{e}_2$

  $\mathbf{names}(\Gamma_{12}^{@}) \cap \mathbf{names}(\Gamma_{22}^{@}) = \emptyset$ comes naturally from (3), (7) and (12).

  We must show:
  $\mathbf{names}(\Gamma_{12}^{@}) \cap \mathbf{names}(\Gamma_{23}^{@}) = \emptyset$: holes in $\mathtt{e}_2$ (associated to $\mathtt{u}$) are fresh so they cannot match a destination name from $\mathtt{t}$ as they don't exist yet when $\mathtt{t}$ is evaluated.
  $\mathbf{names}(\Gamma_{22}^{@}) \cap \mathbf{names}(\Gamma_{13}^{@}) = \emptyset$: slightly harder. Holes in $\mathtt{e}_1$ (associated to $\mathtt{t}$) are fresh too, so I don't see a way for $\mathtt{u}$ to create a term that could mention them, but sequentially, at least, they exist during $\mathtt{u}$ evaluation. In fact, Γ_{22} might have intersection with Γ_{13} (see TyEff_Union) as long as they share the same modalities (it's even harder to prove I think).
  $\mathbf{names}(\Gamma_{13}^{@}) \cap \mathbf{names}(\Gamma_{23}^{@}) = \emptyset$: freshness of holes in both effects, executed sequentially, should be enough.

  Let say this is solved by Lemma 1, with no holes of $\mathtt{e}_1$ negative context appearing as dests in $\mathtt{e}_2$ positive context.

  By TyEff_Union we get (18) $\Gamma_{12}^{@} \sqcup \Gamma_{22}^{@} \uplus @^{-1}\Gamma_{13}^{@} \sqcup @^{-1}\Gamma_{23}^{@} \Vdash \mathtt{e}_1 \cdot \mathtt{e}_2$.
  Inverting TyCmd_Cmd on (17) we get (19) $m{\cdot}(\Gamma_{111}^{@} \sqcup \Gamma_{131}^{@}) \sqcup \Gamma_{211}^{@} \sqcup \Gamma_{231}^{@} \sqcup \Gamma_{3}^{@} \vdash \mathtt{v}_3 : \mathbf{A}_2$ and (20) $m{\cdot}(\Gamma_{112}^{@} \sqcup \Gamma_{132}^{@}) \sqcup \Gamma_{212}^{@} \sqcup \Gamma_{232}^{@} \uplus @^{-1}\Gamma_{3}^{@} \Vdash \mathtt{e}_3$ where (21) «no parses (char 2): Gi***1@ μ Gi2@ = Gi@ »

  We have
  (18) $\Gamma_{12}^{@} \sqcup \Gamma_{22}^{@} \uplus @^{-1}\Gamma_{13}^{@} \sqcup @^{-1}\Gamma_{23}^{@} \Vdash \mathtt{e}_1 \cdot \mathtt{e}_2$
  (20) $m{\cdot}(\Gamma_{112}^{@} \sqcup \Gamma_{132}^{@}) \sqcup \Gamma_{212}^{@} \sqcup \Gamma_{232}^{@} \uplus @^{-1}\Gamma_{3}^{@} \Vdash \mathtt{e}_3$

  Using (21) on (18) to decompose $@^{-1}\Gamma_{23}^{@}$, we get (22) $\Gamma_{12}^{@} \sqcup \Gamma_{22}^{@} \uplus @^{-1}(\Gamma_{131}^{@} \sqcup \Gamma_{231}^{@}) \sqcup @^{-1}(\Gamma_{132}^{@} \sqcup \Gamma_{232}^{@}) \Vdash \mathtt{e}_1 \cdot \mathtt{e}_2$

  We want $\Gamma_{132}^{@}$ from (22) to cancel $m{\cdot}\Gamma_{132}^{@}$ from (20), but the multiplicity doesn't match apparently.

  $\Gamma_{13}^{@}$ contains dests associated to holes that may have been created when evaluation $\mathtt{t}$ into «no parses (char 4): v1 |*** e1 »
  If $\mathtt{v}_1$ is used with delay (result of multiplying its context by m), then should we also delay the RHS of its associated effect? In other terms, if we have «no parses (char 32): {@h:0 |A|n} |- @h' | h := Inl h'*** », and use $@\mathtt{h}'$ with delay m (e.g stored inside another dest in the body of the function), should we also type the RHS of $\mathtt{h} := \mathsf{Inl}\,\mathtt{h}'$ with delay 1? I think so, if we want to keep the property that age of dests and age of the associated holes are the same. Which means a more refined substitution lemma.

Lemma 1 (Freshness of holes)**.** *Let* $\mathtt{t}$ *be a program with no pre-existing ampar sharing hole names.*
During the reduction of $\mathtt{t}$*, the only other place where the names of the holes on the RHS of an effect can appear is in the accompanying value of the command, as destinations.*

Proof. Names of the holes on the RHS of a new effect:

- either are **fresh** (in all BigStep_Fill⟨*Ctor*⟩ rules), which means the only other place where those names are known and can show up is as destinations on the accompanying value of the command (Γ_{12} in TyCmd_Cmd), but not in positive or negative contexts of the command given by the evaluation of a sibling subterm;

- or are those of pre-existing holes coming from the extended value $\overline{\mathtt{v}_2}$ of an ampar, when BigStep_FillComp is evaluated. Because they come from an ampar, they must be neutralized by this ampar, so the left value $\mathtt{v}_1$ of the ampar is the only place where those names can show up, as destinations, if we disallow pre-existing ampar with shared hole names in the body of the initial program. And $\mathtt{v}_1$ is exactly the accompanying value returned by the evaluation of BigStep_FillComp

TODO: prove that this property is preserved by typing rules □

Theorem 2 (Type safety for complete programs)**.** *If* $\emptyset \vdash \mathtt{t} : \mathtt{A}$ *then* $\mathtt{t} \Downarrow \mathtt{v} \mid \varepsilon$ *and* $\emptyset \vdash \mathtt{v} : \mathtt{A}$.