Interoperation for Lazy and Eager Evaluation

- Matthews & Findler
  - Interoperation
  - Boundaries & natural embedding
  - Type safety and equality
- Kinghorn
- Incompatible evaluation strategies

Lambda calculus
Typing
Interoperation
Model
Laziness
Solution

Set of terms, e

(1) $x \in \mathrm{e}$

(2) $M \in \mathrm{e} \Rightarrow \lambda x \,.\, M \in \mathrm{e}$

(3) $M, N \in \mathrm{e} \Rightarrow M\, N \in \mathrm{e}$

$$\mathrm{e} = x \mid \lambda x \,.\, \mathrm{e} \mid \mathrm{e}\ \mathrm{e}$$

(1) x
(2) $\lambda\ x\ .\ e$
(3) $e\ e$

y
by (1)

$\lambda\ y\ .\ y$
by (1), (2)

$(\lambda\ y\ .\ y)\ (\lambda\ y\ .\ y)$
by (1), (2), (3)

$$\lambda\, x\, .\, x\, x \equiv \lambda\, x\, .\, (x\, x) \neq (\lambda\, x\, .\, x)\, x$$

$$\lambda\, x\, x'\, .\, \mathrm{e} \equiv \lambda\, x\, .\, \lambda\, x'\, .\, \mathrm{e}$$

$$\mathrm{e}\ \mathrm{e}'\ \mathrm{e}'' = (\mathrm{e}\ \mathrm{e}')\ \mathrm{e}''$$

$$\ldots y \ldots \qquad\qquad \lambda\, y\, .\, (\ldots y \ldots)$$

Free variable   *Bound variable*

$$y \qquad\qquad \lambda\, y\, .\, y$$

Open term   *Closed term*

term [*expression argument* / *expression parameter*] = *term′*

$$x\,[\,e \,/\, x\,] \;=\; e$$

$$x\,[\,e \,/\, x'\,] \;=\; x$$

$$(\lambda\, x\,.\,e)\,[\,e' \,/\, x\,] \;=\; \lambda\, x\,.\,e$$

$$(\lambda\, x\,.\,e)\,[\,e' \,/\, x'\,] \;=\; \lambda\, x\,.\,(e\,[\,e' \,/\, x'\,])$$

$$(e\ e')\,[\,e'' \,/\, x\,] \;=\; (e\,[\,e'' \,/\, x\,])\,(e'\,[\,e'' \,/\, x\,])$$

Set of reductions, $\rightarrow$

$$(e, e') \in \rightarrow$$

$$e \rightarrow e'$$

$$e \rightarrow e'$$

$$e' \rightarrow e''$$

$$e \rightarrow e' \rightarrow e''$$

$$(\lambda\, x\, .\, e)\; e' \rightarrow e\, [\, x\, /\, e'\,]$$

$$(\lambda\, x\, .\, e)\; e' \rightarrow e\,[\, x \,/\, e'\,]$$

$$e \rightarrow e' \Rightarrow e\; e'' \rightarrow e'\; e''$$

$$\frac{e \rightarrow e'}{e\; e'' \rightarrow e'\; e''}$$

$$e \rightarrow e' \Rightarrow (\lambda\, x\, .\, e'')\; e \rightarrow (\lambda\, x\, .\, e'')\; e'$$

$$\frac{e \rightarrow e'}{(\lambda\, x\, .\, e'')\; e \rightarrow (\lambda\, x\, .\, e'')\; e'}$$

$$\textit{error condition} \rightarrow \textit{error}$$

$$\frac{e \rightarrow e'}{e\ e'' \rightarrow e'\ e''} \qquad \frac{e \rightarrow \textit{error}}{e\ e' \rightarrow \textit{error}}$$

$$\frac{e \rightarrow e'}{(\lambda\, x\, .\, e'')\ e \rightarrow (\lambda\, x\, .\, e'')\ e'} \qquad \frac{e \rightarrow \textit{error}}{(\lambda\, x\, .\, e')\ e \rightarrow \textit{error}}$$

E = *all evaluation contexts*

$$(\lambda\, x\, .\, \mathrm{e})\ \mathrm{e}' \rightarrow \mathrm{e}\ [x\ /\ \mathrm{e}']$$

$$E\ [\ (\lambda\, x\, .\, \mathrm{e})\ \mathrm{e}'\ ] \rightarrow E\ [\ \mathrm{e}\ [x\ /\ \mathrm{e}']\ ]$$

$$E = [\ ]\ |\ E\ \mathrm{e}\ |\ (\lambda\, x\, .\, \mathrm{e})\ E$$

$$\ldots[\ ]\ldots$$

$$E\ [\ \mathrm{e}\ ] = \ldots\mathrm{e}\ldots$$

$$v \quad = \quad \lambda\, x \,.\, \mathrm{e} \mid \underline{n}$$

$$\mathrm{e} \quad = \quad \begin{array}{l} x \mid v \mid \mathrm{e}\ \mathrm{e} \mid \textbf{+/-}\ \mathrm{e}\ \mathrm{e} \mid \textbf{if0}\ \mathrm{e}\ \mathrm{e}\ \mathrm{e} \\ \textbf{fun?}\ \mathrm{e} \mid \textbf{num?}\ \mathrm{e} \mid \textbf{wrong}\ \textit{string} \end{array}$$

wrong *string*	E [**wrong** *string*] → **Error:** *string*
+ e e	$E\,[\,\mathbf{+}\ \underline{n}\ \underline{n'}\,] \rightarrow E\,[\,\underline{n + n'}\,]$
- e e	$E\,[\,\mathbf{-}\ \underline{n}\ \underline{n'}\,] \rightarrow E\,[\,\underline{\max\,(n - n', 0)}\,]$
if0 e e e	$E\,[\,\mathbf{if0}\ \underline{0}\ \mathrm{e}\ \mathrm{e'}\,] \rightarrow E\,[\,\mathrm{e}\,]$ $E\,[\,\mathbf{if0}\ \underline{n}\ \mathrm{e}\ \mathrm{e'}\,] \rightarrow E\,[\,\mathrm{e'}\,]$
fun? e	$E\,[\,\mathbf{fun?}\ (\lambda\, x\,.\,\mathrm{e})\,] \rightarrow E\,[\,\underline{0}\,]$ $E\,[\,\mathbf{fun?}\ \underline{n}\,] \rightarrow E\,[\,\underline{1}\,]$
num? e	$E\,[\,\mathbf{num?}\ \underline{n}\,] \rightarrow E\,[\,\underline{0}\,]$ $E\,[\,\mathbf{num?}\ (\lambda\, x\,.\,\mathrm{e})\,] \rightarrow E\,[\,\underline{1}\,]$

Lambda calculus

Typing

Interoperation

Model

Laziness

Solution

Set of types, t

$$t = \mathbf{N} \mid t \to t$$

$$\lambda\, x : t \,.\, e$$

$$t \to t \to t \equiv t \to (t \to t)$$

Set of judgments, $\vdash$

$$e : t \equiv (e , t)$$

Γ *is* $x_n : t_n , \ldots , x_1 : t_1$

$$(\Gamma , e : t) \in \vdash$$

$$\Gamma \vdash e : t$$

$$\Gamma \vdash t$$

Number type

$$\vdash \mathbf{N}$$

Function type

$$\frac{\vdash t \quad \text{—} \quad \vdash t'}{\vdash t \rightarrow t'}$$

Number

$$\vdash \underline{n} : \mathbf{N}$$

Variable

$$\Gamma, x : t \vdash x : \mathrm{t}$$

Function

$$\frac{\Gamma, x : t \vdash \mathrm{e} : t'}{\Gamma \vdash \lambda\, x : t \,.\, \mathrm{e} : t \rightarrow t'}$$

Application

$$\frac{\Gamma \vdash \mathrm{e} : t \rightarrow t' \quad \text{—} \quad \Gamma \vdash \mathrm{e}' : t}{\Gamma \vdash \mathrm{e}\ \mathrm{e}' : t'}$$

Arithmetic

$$\frac{\Gamma \vdash \mathrm{e} : \mathbf{N} \quad — \quad \Gamma \vdash \mathrm{e}' : \mathbf{N}}{\Gamma \vdash \mathbf{+/-}\ \mathrm{e}\ \mathrm{e}' : \mathbf{N}}$$

Condition

$$\frac{\Gamma \vdash \mathrm{e} : \mathbf{N} \quad — \quad \Gamma \vdash \mathrm{e}'/\mathrm{e}'' : t}{\Gamma \vdash \mathbf{if0}\ \mathrm{e}\ \mathrm{e}'\ \mathrm{e}'' : t}$$

Error

$$\frac{\Gamma \vdash t}{\Gamma \vdash \mathbf{wrong}\ t\ \textit{string} : t}$$

Type

$$\Gamma \vdash \mathbf{T}$$

Number

$$\Gamma \vdash \underline{n} : \mathbf{T}$$

Variable

$$\Gamma, x : \mathbf{T} \vdash x : \mathbf{T}$$

Application

$$\frac{\Gamma \vdash \mathrm{e} : \mathbf{T} \text{ — } \Gamma \vdash \mathrm{e}' : \mathbf{T}}{\Gamma \vdash \mathrm{e}\ \mathrm{e}' : \mathbf{T}}$$

Function

$$\frac{\Gamma, x : \mathbf{T} \vdash \mathrm{e} : \mathbf{T}}{\Gamma \vdash \lambda\, x \,.\, \mathrm{e} : \mathbf{T}}$$

Arithmetic

$$\frac{\Gamma \vdash \mathrm{e} : \mathbf{T} \text{ — } \Gamma \vdash \mathrm{e}' : \mathbf{T}}{\Gamma \vdash \mathbf{+}/\mathbf{-}\ \mathrm{e}\ \mathrm{e}'}$$

Predicate

$$\frac{\Gamma \vdash \mathrm{e} : \mathbf{T}}{\Gamma \vdash \textbf{fun?/num? } \mathrm{e} : \mathbf{T}}$$

Error

$$\Gamma \vdash \textbf{wrong } \textit{string} : \mathbf{T}$$

$$\lambda x : \mathbf{N} . x$$

$$\lambda x : \mathbf{N} \to \mathbf{N} . x$$

$$\Lambda y . \lambda x : y . x$$

$$(\Lambda y . \lambda x : y . x) \langle \mathbf{N} \rangle \to \lambda x : \mathbf{N} . x$$

$$(\Lambda y . \lambda x : y . x) \langle \mathbf{N} \to \mathbf{N} \rangle \to \lambda x : \mathbf{N} \to \mathbf{N} . x$$

Type variables

y

Type abstraction

$\Lambda\, y\, .\, e$

Type application

$e \langle\, t\, \rangle$

Universally-quantified / for-all types

$\forall\, y\, .\, t$

Free & bound type variables

$\Lambda\, y\, .\, (\ldots\, y\, \ldots)$

$$\Lambda\, y\, y' \,.\, e \equiv \Lambda\, y \,.\, \Lambda\, y' \,.\, e$$

$$e \langle\, t \,\rangle \langle\, t' \,\rangle \equiv (e \langle\, t \,\rangle) \langle\, t' \,\rangle$$

$$\Lambda\, y \,. \ldots y \ldots = \Lambda\, y' \,. \ldots y' \ldots$$

term [type argument / type parameter] = term′

$$x\,[\,t\,/\,y\,] \;=\; x$$

$$(\lambda\, x\,.\,\mathrm{e})\,[\,t\,/\,y\,] \;=\; \lambda\, x\,.\,(\mathrm{e}\,[\,t\,/\,y\,])$$

$$(\mathrm{e}\ \mathrm{e}')\,[\,t\,/\,y\,] \;=\; (\mathrm{e}\,[\,t\,/\,y\,])\ (\mathrm{e}'\,[\,t\,/\,y\,])$$

$$(\mathbf{+/-}\ \mathrm{e}\ \mathrm{e}')\,[\,t\,/\,y\,] \;=\; \mathbf{+/-}\ (\mathrm{e}\,[\,t\,/\,y\,])\ (\mathrm{e}'\,[\,t\,/\,y\,])$$

$$(\mathbf{if0}\ \mathrm{e}\ \mathrm{e}'\ \mathrm{e}'')\,[\,t\,/\,y\,] \;=\; \mathbf{if0}\ (\mathrm{e}\,[\,t\,/\,y\,])\ (\mathrm{e}'\,[\,t\,/\,y\,])\ (\mathrm{e}''\,[\,t\,/\,y\,])$$

$$(\Lambda y\,.\,\mathrm{e})\,[\,t\,/\,y\,] \;=\; \Lambda y\,.\,\mathrm{e}$$

$$(\Lambda y\,.\,\mathrm{e})\,[\,t\,/\,y'\,] \;=\; \Lambda y\,.\,\mathrm{e}\,[\,t\,/\,y'\,]$$

$$(\mathrm{e}\,\langle\, t \,\rangle)\,[\,t'\,/\,y\,] \;=\; (\mathrm{e}\,[\,t'\,/\,y\,])\,\langle\, t \,\rangle$$

$$\textit{type} \; [\; \textit{type argument} \; / \; \textit{type parameter} \;] = \textit{type}'$$

$$\mathbf{N}\,[\,t \,/\, y\,] \;=\; \mathbf{N}$$

$$(t \to t')\,[\,t \,/\, y\,] \;=\; t\,[\,t \,/\, y\,] \to t'\,[\,t \,/\, y\,]$$

$$y\,[\,t \,/\, y\,] \;=\; t$$

$$y\,[\,t \,/\, y'\,] \;=\; y$$

$$(\forall\, y \,.\, t)\,[\,t' \,/\, y\,] \;=\; \forall\, y \,.\, t$$

$$(\forall\, y \,.\, t)\,[\,t' \,/\, y'\,] \;=\; \forall\, y \,.\, t\,[\,t' \,/\, y'\,]$$

$$E[(\Lambda y.e)\langle t\rangle] \to E[e[t/y]]$$

Lambda calculus
Typing
Interoperation
Model
Laziness
Solution

Haskell

$$e_h = \cdots \mid \mathbf{hm}\ t_h\ t_m\ e_m \mid \mathbf{hs}\ t_h\ e_s$$

ML

$$e_m = \cdots \mid \mathbf{mh}\ t_m\ t_h\ e_h \mid \mathbf{ms}\ t_m\ e_s$$

Scheme

$$e_s = \cdots \mid \mathbf{sh}\ t_h\ e_h \mid \mathbf{sm}\ t_m\ e_m$$

Haskell-ML

$$\Gamma \vdash_h \mathbf{hm}\ t_h\ t_m\ e_m : t_h$$

$$\Gamma \vdash_h t_h$$

$$\Gamma \vdash_m t_m$$

$$\Gamma \vdash_m e_m : t_m'$$

$$t_m = t_m'$$

$$t_h = t_m$$

ML-Haskell

$$\Gamma \vdash_m \mathbf{mh}\ t_m\ t_h\ e_h : t_m$$

Haskell-Scheme

$$\Gamma \vdash_h \mathbf{hs}\ t_h\ e_s : t_h$$

$$\Gamma \vdash_h t_h$$

$$\Gamma \vdash_s e_s : \mathbf{T}$$

Scheme-Haskell

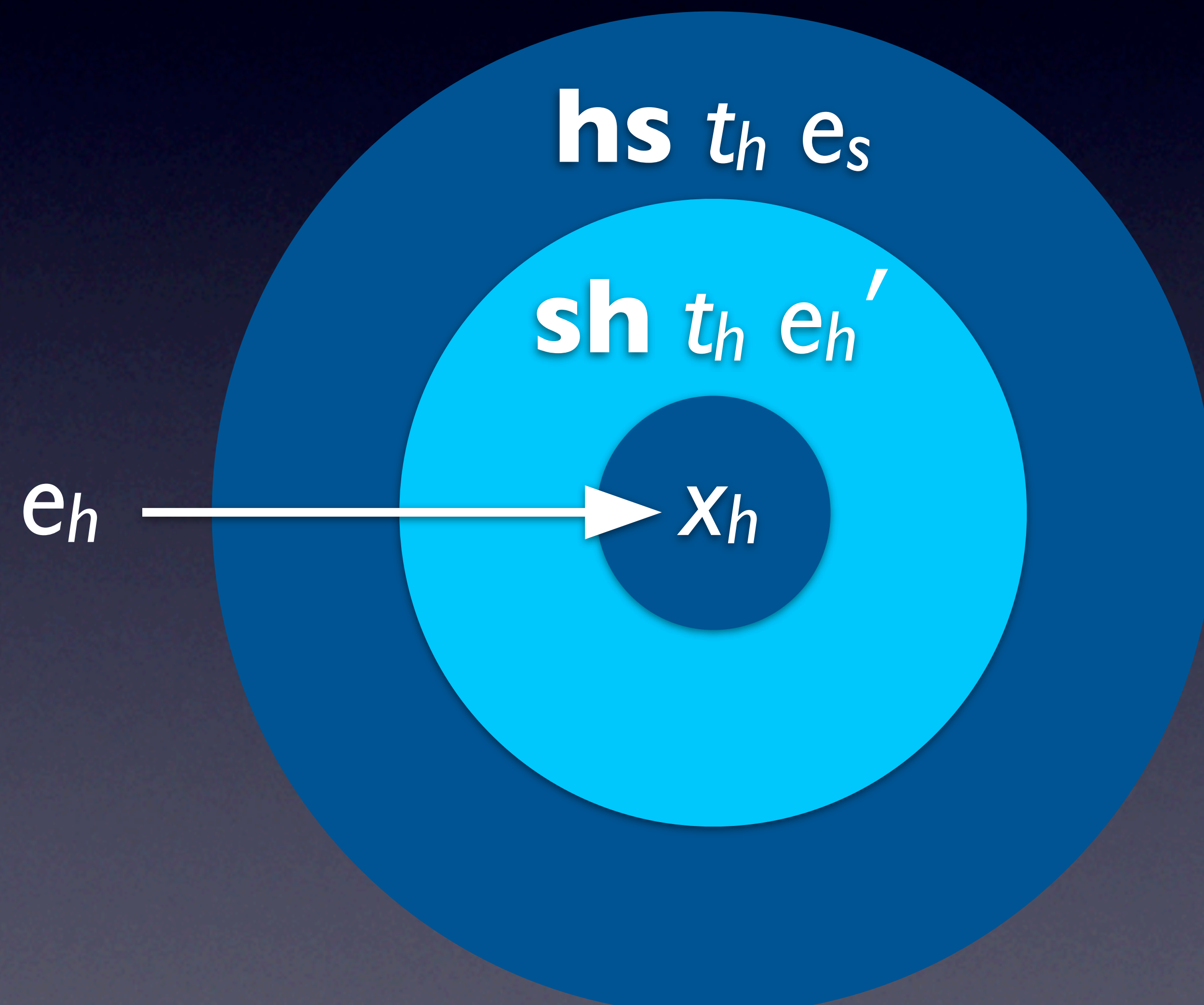
$$\Gamma \vdash_s \mathbf{sh}\ t_h\ e_h : \mathbf{T}$$

$$\Gamma \vdash_h t_h$$

$$\Gamma \vdash_h e_h : t_h'$$

$$t_h = t_h'$$

$$(\mathbf{hs}\ t_h\ (\mathbf{sh}\ t_h\ x_h))\ [e_h\ /\ x_h]$$



Embedding substitution

$$(\mathbf{hm}\ t_h\ t_m\ e_m)\ [e_h\ /\ x_h] = \mathbf{hm}\ t_h\ t_m\ e_m\ [e_h\ /\ x_h]$$

$$(\mathbf{hs}\ t_h\ e_s)\ [e_h\ /\ x_h] = \mathbf{hs}\ t_h\ e_s\ [e_h\ /\ x_h]$$

Foreign substitution

$$\ldots e_m \ldots\ [e_h\ /\ x_h] = \ldots e_m\ [e_h\ /\ x_h] \ldots$$

$$(\lambda\ x_m\ .\ e_m)\ [e_h\ /\ x_h] = \lambda\ x_m\ .\ (e_m\ [e_h\ /\ x_h])$$

$$\mathcal{E}[\ \mathbf{hm\ N\ N}\ \underline{n}\ ]_h \rightarrow \mathcal{E}[\ \underline{n}\ ]$$

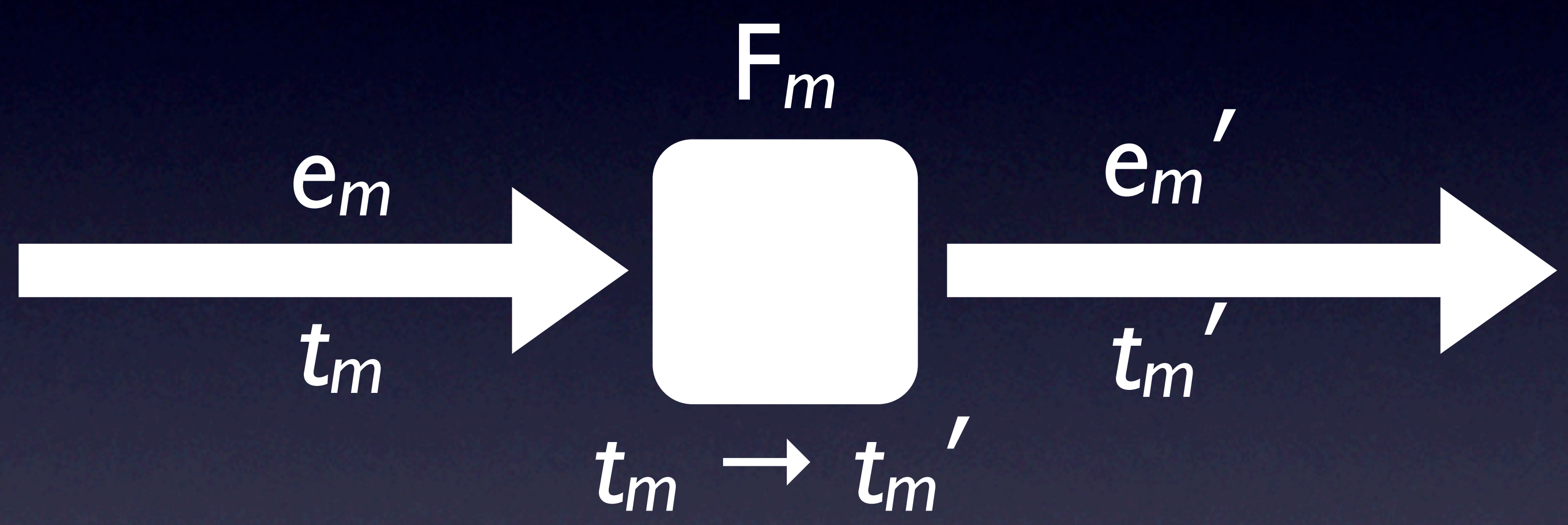
$$\mathcal{E}[\ \mathbf{hs\ N}\ \underline{n}\ ]_h \rightarrow \mathcal{E}[\ \underline{n}\ ]$$

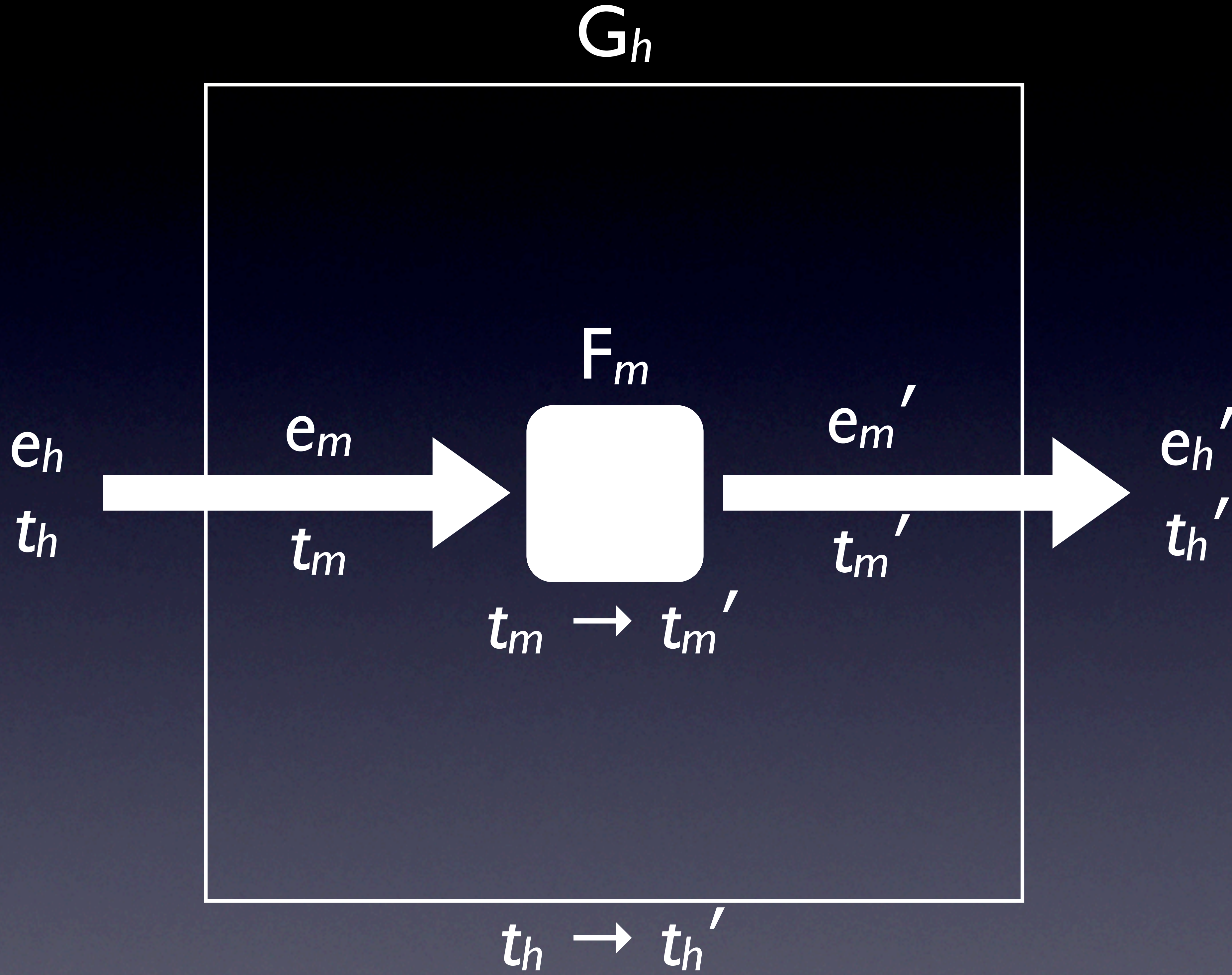
$$\mathcal{E}[\ \mathbf{mh\ N\ N}\ \underline{n}\ ]_m \rightarrow \mathcal{E}[\ \underline{n}\ ]$$

$$\mathcal{E}[\ \mathbf{ms\ N}\ \underline{n}\ ]_m \rightarrow \mathcal{E}[\ \underline{n}\ ]$$

$$\mathcal{E}[\ \mathbf{sh\ N}\ \underline{n}\ ]_s \rightarrow \mathcal{E}[\ \underline{n}\ ]$$

$$\mathcal{E}[\ \mathbf{sm\ N}\ \underline{n}\ ]_s \rightarrow \mathcal{E}[\ \underline{n}\ ]$$


F_m
e_m
t_m
e_m'
t_m'
$t_m \rightarrow t_m'$


G_h
F_m
e_h
t_h
e_m
t_m
e_m'
t_m'
e_h'
t_h'
$t_m \rightarrow t_m'$
$t_h \rightarrow t_h'$

Haskell to ML

$e_m = \mathbf{mh}\ t_m\ t_h\ e_h$

ML application

$e_m' = F_m\ e_m$

ML to Haskell

$e_h' = \mathbf{hm}\ t_h'\ t_m'\ e_m'$

Factor out e_h

$\lambda\, x_h : t_h\ .\ \mathbf{hm}\ t_h'\ t_m'\ (F_m\ (\mathbf{mh}\ t_m\ t_h\ x_h))$

$$\mathcal{E}\,[\ \mathbf{hm}\ (t_h \to t_h{}')\ (t_m \to t_m{}')\ v_m\ ]_h$$

$$\to$$

$$\mathcal{E}\,[\ \lambda\, x_h : t_h\ .\ \mathbf{hm}\ t_h{}'\ t_m{}'\ (v_m\ (\mathbf{mh}\ t_m\ t_h\ x_h))\ ]$$

Brands

Lambda calculus
Typing
Interoperation
Model
Laziness
Solution

ML & Scheme

- Numbers, arithmetic, conditions
- Functions, applications
- Boundaries
- Errors
- Eager evaluation

ML

- Statically typed
- Type abstractions
- Branded types
- Fixed-point operations

Scheme

- Dynamically typed
- Closed term typing
- Type predicates

Eval cxts, nonterms

- Show math defs

Lambda calculus
Typing
Interoperation
Model
Laziness
Solution

- Eager vs. lazy evaluation
- Incompatible evaluation strategies
  - Function behavior
  - Value conversion

$$K_s \equiv \lambda\, x\ \ y\, .\, x$$

$$K_h : \forall\, u_h\ u_h'\, .\, u_h \to u_h' \to u_h$$

$$K_h \equiv \Lambda\, u_h\ u_h'\, .\, \mathbf{hs}\ (u_h \to u_h' \to u_h)\ K_s$$

$$K_{hn} \equiv K_h\ \langle \mathbf{N} \rangle\ \langle \mathbf{N} \rangle = \mathbf{hs}\ (\mathbf{N} \to \mathbf{N} \to \mathbf{N})\ K_s$$

$$K_{hn}\ \Omega\ \underline{0} \not\to$$

$$K_{hn}\ \Omega\ \underline{0} = (\mathbf{hs}\ (\mathbf{N} \to \mathbf{N} \to \mathbf{N})\ K_s)\ \Omega\ \underline{0}$$

$$(\mathbf{hs}\ (\mathbf{N} \to \mathbf{N} \to \mathbf{N})\ K_s)\ \Omega\ \underline{0}$$

$$\to$$

$$(\lambda\ x_h : \mathbf{N}\ .\ \mathbf{hs}\ (\mathbf{N} \to \mathbf{N})\ (K_s\ (\mathbf{sh}\ \mathbf{N}\ x_h)))\ \Omega\ \underline{0}$$

$$\to$$

$$\mathbf{hs}\ (\mathbf{N} \to \mathbf{N})\ (K_s\ (\mathbf{sh}\ \mathbf{N}\ \Omega))\ \underline{0}$$

$$\to$$

$$\mathbf{hs}\ (\mathbf{N} \to \mathbf{N})\ (K_s\ (\mathbf{sh}\ \mathbf{N}\ \Omega))\ \underline{0}$$

$$\vdots$$

$$e_h = \cdots \mid \textbf{nil}\ t_h \mid \textbf{cons}\ e_h\ e_h \mid \textbf{hd}\ e_h \mid \textbf{tl}\ e_h \mid \textbf{null?}\ e_h$$

$$t_h = \cdots \mid \{\ t_h\ \}$$

$$E_h = \cdots \mid \textbf{hd}\ E_h \mid \textbf{tl}\ E_h \mid \textbf{null?}\ E_h$$

$$\frac{\Gamma \vdash_h t_h}{\Gamma \vdash_h \{\ t_h\ \}} \qquad \frac{\Gamma \vdash_h e_h : \{\ t_h\ \}}{\Gamma \vdash_h \mathbf{hd}\ e_h : t_h}$$

$$\frac{\Gamma \vdash_h t_h}{\Gamma \vdash_h \mathbf{nil}\ t_h : \{\ t_h\ \}} \qquad \frac{\Gamma \vdash_h e_h : \{\ t_h\ \}}{\Gamma \vdash_h \mathbf{tl}\ e_h : \{\ t_h\ \}}$$

$$\frac{\Gamma \vdash_h e_h : \{\ t_h\ \}}{\Gamma \vdash_h \mathbf{null?}\ e_h : \mathbf{N}} \qquad \frac{\Gamma \vdash_h e_h : t_h \quad \Gamma \vdash_h e_h' : \{\ t_h\ \}}{\Gamma \vdash_h \mathbf{cons}\ e_h\ e_h' : \{\ t_h\ \}}$$

$\mathscr{E}$ [**hd** (**nil** t_h)]$_h$ → $\mathscr{E}$ [**wrong** t_h "Empty list"]

$\mathscr{E}$ [**hd** (**cons** e_h e_h')]$_h$ → $\mathscr{E}$ [e_h]

$\mathscr{E}$ [**tl** (**nil** t_h)]$_h$ → $\mathscr{E}$ [**wrong** { t_h } "Empty list"]

$\mathscr{E}$ [**tl** (**cons** e_h e_h')]$_h$ → $\mathscr{E}$ [e_h']

$\mathscr{E}$ [**null?** (**nil** t_h)]$_h$ → $\mathscr{E}$ [$\underline{0}$]

$\mathscr{E}$ [**null?** (**cons** e_h e_h')]$_h$ → $\mathscr{E}$ [$\underline{1}$]

$$e_s = \cdots \mid \textbf{cons}\ e_s\ e_s \mid \textbf{hd}\ e_s \mid \textbf{tl}\ e_s$$

$$v_s = \cdots \mid \textbf{nil} \mid \textbf{cons}\ v_s\ v_s$$

$$E_s = \cdots \mid \textbf{cons}\ E_s\ \mathrm{es} \mid \textbf{cons}\ v_s\ E_s \mid \textbf{hd}\ E_s \mid \textbf{tl}\ E_s \mid \textbf{null?}\ E_s$$

$$\mathcal{E}[\,\textbf{hd/tl nil}\,]_s \rightarrow \mathcal{E}[\,\textbf{wrong}\text{ “Empty list”}\,]$$

$$\mathcal{E}[\,\textbf{hd }(\textbf{cons } v_s\ v_s')\,]_s \rightarrow \mathcal{E}[\,v_s\,]$$

$$\mathcal{E}[\,\textbf{tl }(\textbf{cons } v_s\ v_s')\,]_s \rightarrow \mathcal{E}[\,v_s'\,]$$

$$\mathcal{E}[\,\textbf{hd/tl } v_s\,]_s \rightarrow \mathcal{E}[\,\textbf{wrong}\text{ “Not a list”}\,]$$

$$\mathcal{E}[\,\textbf{null? nil}\,]_s \rightarrow \mathcal{E}[\,\underline{0}\,]$$

$$\mathcal{E}[\,\textbf{null? } v_s\,]_s \rightarrow \mathcal{E}[\,\underline{1}\,]$$

$$\text{zeroes}_h \equiv \textbf{fix}\ (\lambda\ x : \{\ \mathbf{N}\ \}\ .\ \textbf{cons}\ \underline{0}\ x)$$

$$\text{zeroes}_h = \textbf{cons}\ \underline{0}\ \text{zeroes}_h$$

$$\text{zeroes}_m \equiv \textbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ \text{zeroes}_h$$

$$\text{zeroes}_m \not\to$$

$\mathrm{zeroes}_m = \mathbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ \mathrm{zeroes}_h \rightarrow$

$\mathbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ (\mathbf{cons}\ \underline{0}\ \mathrm{zeroes}_h) \rightarrow$

$\mathbf{cons}\ (\mathbf{mh}\ \mathbf{N}\ \mathbf{N}\ \underline{0})\ (\mathbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ \mathrm{zeroes}_h) \rightarrow$

$\mathbf{cons}\ \underline{0}\ (\mathbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ \mathrm{zeroes}_h) =$

$\mathbf{cons}\ \underline{0}\ \mathrm{zeroes}_m \not\rightarrow$

Lambda calculus
Typing
Interoperation
Model
Laziness
Solution

Function conversion

$$\vdots$$

$$\textbf{hs}\ (\mathbf{N} \to \mathbf{N})\ (K_s\ (\textbf{sh}\ \mathbf{N}\ \Omega))\ \underline{0} \qquad\qquad v_m\ E_m$$

$$\vdots$$

List construction conversion

$$\vdots$$

$$\textbf{cons}\ \underline{0}\ (\textbf{mh}\ \{\ \mathbf{N}\ \}\ \{\ \mathbf{N}\ \}\ \mathrm{zeroes}_h) \qquad\qquad \textbf{cons}\ v_m\ E_m$$

$$\vdots$$

$E_m =$

$[\,]_m$

$E_m\ e_m$

$v_m\ E_m$

$E_m \langle t_m \rangle$

fix E_m

+/- $E_m\ e_m$

+/- $v_m\ E_m$

if0 $E_m\ e_m\ e_m$

cons $E_m\ e_m$

cons $v_m\ E_m$

hd E_m

tl E_m

null E_m

mh $t_m\ t_h\ E_h$

ms $k_s\ E_s$

$$
E_m = \begin{array}{c}
\vdots \\
v_m\ E_m \\
\textbf{cons}\ E_m\ \mathrm{e}_m \\
\textbf{cons}\ v_m\ E_m \\
\textbf{mh}\ t_m\ t_h\ E_h \\
\vdots \\
E_m\ \mathrm{e}_m \\
v_m\ E_m \\
\vdots
\end{array}
$$

$$
U_m = \begin{array}{c} \vdots \\ v_m\ U_m \\ \textbf{cons}\ U_m\ e_m \\ \textbf{cons}\ u_m\ U_m \\ \\ \vdots \\ F_m\ e_m \\ f_m\ U_m \\ \vdots \end{array}
\qquad
\begin{array}{l} F_m = U_m \mid \textbf{mh}\ t_m\ t_h\ E_h \\ \\ \\ f_m = u_m \mid \textbf{mh}\ t_m\ t_h\ E_h \\ \\ \\ u_m = \lambda\ x_m : t_m\ .\ e_m \mid \cdots \end{array}
$$

$$F_m = U_m \mid \mathbf{mh}\ t_m\ t_h\ E_h$$

$$U_m =$$

$[\ ]_m$	$\mathbf{if0}\ F_m\ e_m\ e_m$
$F_m\ e_m$	$\mathbf{cons}\ U_m\ e_m$
$f_m\ U_m$	$\mathbf{cons}\ u_m\ U_m$
$F_m \langle\ t_m\ \rangle$	$\mathbf{hd}\ F_m$
$\mathbf{fix}\ F_m$	$\mathbf{tl}\ F_m$
$\mathbf{+/-}\ F_m\ e_m$	$\mathbf{null}\ F_m$
$\mathbf{+/-}\ f_m\ F_m$	$\mathbf{ms}\ k_s\ E_s$

$$\mathcal{E}[\,(\lambda\, x_m : t_m\,.\,e_m)\; u_m\,]_m \rightarrow \mathcal{E}[\,e_m\,[u_m / x_m]\,]$$

$$\mathcal{E}[\,\textbf{hd}\;(\textbf{cons}\; u_m\; u_m')\,]_m \rightarrow \mathcal{E}[\,u_m\,]$$

$$\mathcal{E}[\,\textbf{tl}\;(\textbf{cons}\; u_m\; u_m')\,]_m \rightarrow \mathcal{E}[\,u_m'\,]$$

$$\mathcal{E}[\,\textbf{null}\;(\textbf{cons}\; u_m\; u_m')\,]_m \rightarrow \mathcal{E}[\,\bot\,]$$

- Common expressions
- Incompatible strictness points
- Interoperation side effects
- Mirror non-strictness for embeddings

- Matthews & Findler
- Evaluation strategies
- Incompatible strictness points
- Forcing & deferring embedded evaluation