Types
CST Part IB Paper 8 & 9

Victor Zhao
xz398@cam.ac.uk

1 Simply-Typed λ-Calculus

Syntax

Types	T	::=	$1 \mid 0 \mid T_1 \times T_2 \mid T_1 + T_2 \mid T_1 \to T_2$
Terms	e	::=	$x \mid \langle\rangle \mid \langle e_1, e_2\rangle \mid \mathsf{fst}\ e \mid \mathsf{snd}\ e \mid \mathsf{L}\ e \mid \mathsf{R}\ e \mid \mathsf{case}(e, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2)$
		|	$\lambda x : T.\ e \mid e_1\ e_2 \mid \mathsf{abort}$
Values	v	::=	$\langle\rangle \mid \langle v_1, v_2\rangle \mid \lambda x : T.\ e \mid \mathsf{L}\ v \mid \mathsf{R}\ v$
Contexts	Γ	::=	$\cdot \mid \Gamma, x : T$

Typing rules

(I: introduction rule, E: elimination rule, HYP: hypothesis)

$$\frac{}{\Gamma \vdash \langle\rangle : 1}\ 1\mathrm{I} \qquad \frac{\Gamma \vdash e_1 : T_1 \qquad \Gamma \vdash e_2 : T_2}{\Gamma \vdash \langle e_1, e_2\rangle : 1}\ \times\mathrm{I} \qquad \frac{\Gamma \vdash e : T_1 \times T_2}{\Gamma \vdash \mathsf{fst}\ e : T_1}\ \times\mathrm{E}_1 \qquad \frac{\Gamma \vdash e : T_1 \times T_2}{\Gamma \vdash \mathsf{snd}\ e : T_2}\ \times\mathrm{E}_2$$

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \textsc{Hyp} \qquad \frac{\Gamma, x : T \vdash e : T'}{\Gamma \vdash \lambda x : T.\ e : T \to T'}\ \to\mathrm{I} \qquad \frac{\Gamma \vdash e_1 : T \to T' \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1\ e_2 : T'}\ \to\mathrm{E}$$

$$\frac{\Gamma \vdash e : T_1}{\Gamma \vdash \mathsf{L}\ e : T_1 + T_2}\ +\mathrm{I}_1 \qquad \frac{\Gamma \vdash e : T_2}{\Gamma \vdash \mathsf{R}\ e : T_1 + T_2}\ +\mathrm{I}_2$$

$$\frac{\Gamma \vdash e : T_1 + T_2 \qquad \Gamma, x : X \vdash e_1 : T \qquad \Gamma, x : X \vdash e_2 : T}{\Gamma \vdash \mathsf{case}(e, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2) : T}\ +\mathrm{E}$$

(No introduction for 0) $$\frac{\Gamma \vdash e : 0}{\Gamma \vdash \mathsf{abort}\ e : T}\ 0\mathrm{E}$$

Operational semantics

(No rule for unit) $$\frac{e_1 \rightsquigarrow e_1'}{\langle e_1, e_2\rangle \rightsquigarrow \langle e_1', e_2\rangle}\ \textsc{Pair1} \qquad \frac{e_2 \rightsquigarrow e_2'}{\langle v, e_2\rangle \rightsquigarrow \langle v, e_2'\rangle}\ \textsc{Pair2}$$

$$\frac{}{\mathsf{fst}\ \langle v_1, v_2\rangle \rightsquigarrow v_1}\ \textsc{Proj1} \qquad \frac{}{\mathsf{snd}\ \langle v_1, v_2\rangle \rightsquigarrow v_2}\ \textsc{Proj2} \qquad \frac{e \rightsquigarrow e'}{\mathsf{fst}\ e \rightsquigarrow \mathsf{fst}\ e'}\ \textsc{Proj3} \qquad \frac{e \rightsquigarrow e'}{\mathsf{snd}\ e \rightsquigarrow \mathsf{snd}\ e'}\ \textsc{Proj4}$$

$$\frac{e \rightsquigarrow e'}{\mathsf{L}\ e \rightsquigarrow \mathsf{L}\ e'}\ \textsc{Sum1} \qquad \frac{e \rightsquigarrow e'}{\mathsf{R}\ e \rightsquigarrow \mathsf{R}\ e'}\ \textsc{Sum2}$$

$$\frac{e \rightsquigarrow e'}{\mathsf{case}(e, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2) \rightsquigarrow \mathsf{case}(e', \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2)}\ \textsc{Case1}$$

$$\frac{}{\mathsf{case}(\mathsf{L}\ v, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2) \rightsquigarrow [v/x]e_1}\ \textsc{Case2} \qquad \frac{}{\mathsf{case}(\mathsf{R}\ v, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2) \rightsquigarrow [v/y]e_2}\ \textsc{Case3}$$

$$\frac{e_1 \rightsquigarrow e_1'}{e_1\ e_2 \rightsquigarrow e_1'\ e_2}\ \textsc{App1} \qquad \frac{e_2 \rightsquigarrow e_2'}{v\ e_2 \rightsquigarrow v\ e_2'}\ \textsc{App2} \qquad \frac{}{(\lambda x : T.\ e)\ v \rightsquigarrow [v/x]e}\ \textsc{Fn}$$

$$\frac{e \rightsquigarrow e'}{\mathsf{abort}\ e \rightsquigarrow \mathsf{abort}\ e'}\ \textsc{Abort}$$

2 Polymorphic λ-Calculus (System F)

Syntax

$$
\begin{array}{llll}
\text{Types} & T & ::= & \alpha \mid T_1 \to T_2 \mid \forall \alpha.\, T \mid \exists \alpha.\, T \\
\text{Terms} & e & ::= & x \mid \lambda x : T.\, e \mid e_1\ e_2 \mid \Lambda \alpha.\, e \mid e\ T \mid \mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, e_{\mathrm{impl}}) \\
 & & \mid & \mathsf{let\ pack}(\alpha, x) = e_{\mathrm{impl}}\ \mathsf{in}\ e_{\mathrm{use}} \\
\text{Values} & v & ::= & \lambda x : T.\, e \mid \Lambda \alpha.\, e \mid \mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, v_{\mathrm{impl}}) \\
\text{Type Contexts} & \Theta & ::= & \cdot \mid \Theta, \alpha \\
\text{Term Contexts} & \Gamma & ::= & \cdot \mid \Gamma, x : T
\end{array}
$$

Well-formedness of types

$$
\frac{\alpha \in \Theta}{\Theta \vdash \alpha\ \mathsf{type}}
\qquad
\frac{\Theta \vdash T_1\ \mathsf{type} \qquad \Theta \vdash T_2\ \mathsf{type}}{\Theta \vdash T_1 \to T_2\ \mathsf{type}}
\qquad
\frac{\Theta, \alpha \vdash T\ \mathsf{type}}{\Theta \vdash \forall \alpha.\, T\ \mathsf{type}}
$$

Well-formedness of term contexts

$$
\frac{}{\Theta \vdash \cdot\ \mathsf{ctx}}
\qquad
\frac{\Theta \vdash \Gamma\ \mathsf{ctx} \qquad \Theta \vdash T\ \mathsf{type}}{\Theta \vdash \Gamma, x : T\ \mathsf{ctx}}
$$

Typing rules

$$
\frac{x : T \in \Gamma}{\Theta; \Gamma \vdash x : T}\ \textsc{Hyp}
\qquad
\frac{\Theta \vdash T\ \mathsf{type} \qquad \Theta; \Gamma, x : T \vdash e : T'}{\Theta; \Gamma \vdash \lambda x : T.\, e : T \to T'}\ {\to}\mathrm{I}
$$

$$
\frac{\Theta; \Gamma \vdash e_1 : T \to T' \qquad \Theta; \Gamma \vdash e_2 : T}{\Theta; \Gamma \vdash e_1\ e_2 : T'}\ {\to}\mathrm{E}
$$

$$
\frac{\Theta, \alpha; \Gamma \vdash e : T}{\Theta; \Gamma \vdash \Lambda \alpha.\, e : \forall \alpha.\, T}\ \forall\mathrm{I}
\qquad
\frac{\Theta; \Gamma \vdash e : \forall \alpha.\, T' \qquad \Theta \vdash T\ \mathsf{type}}{\Theta; \Gamma \vdash e\ T : [T/\alpha]T'}\ \forall\mathrm{E}
$$

$$
\frac{\Theta, \alpha_{\mathrm{abs}} \vdash T_{\mathrm{sig}}\ \mathsf{type} \qquad \Theta \vdash T_{\mathrm{conc}}\ \mathsf{type} \qquad \Theta; \Gamma \vdash e_{\mathrm{impl}} : [T_{\mathrm{conc}}/\alpha_{\mathrm{abs}}]T_{\mathrm{sig}}}{\Theta; \Gamma \vdash \mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, e_{\mathrm{impl}}) : \exists \alpha_{\mathrm{abs}}.\, T_{\mathrm{sig}}}\ \exists\mathrm{I}
$$

$$
\frac{\Theta; \Gamma \vdash e_{\mathrm{impl}} : \exists \alpha_{\mathrm{abs}}.\, T_{\mathrm{sig}} \qquad \Theta, \alpha; \Gamma, x : [\alpha_{\mathrm{abs}}/\alpha]T_{\mathrm{sig}} \vdash e_{\mathrm{use}} : T_{\mathrm{use}} \qquad \Theta \vdash T_{\mathrm{use}}\ \mathsf{type}}{\Theta; \Gamma \vdash \mathsf{let\ pack}(\alpha, x) = e_{\mathrm{impl}}\ \mathsf{in}\ e_{\mathrm{use}} : T_{\mathrm{use}}}\ \exists\mathrm{E}
$$

Operational semantics

(Cong: congruence rule, Eval: evaluation rule)

$$
\frac{e_1 \rightsquigarrow e_1'}{e_1\ e_2 \rightsquigarrow e_1'\ e_2}\ \textsc{CongFun}
\qquad
\frac{e_2 \rightsquigarrow e_2'}{v\ e_2 \rightsquigarrow v\ e_2'}\ \textsc{CongFunArg}
\qquad
\frac{}{(\lambda x : T.\, e)\ v \rightsquigarrow [v/x]e}\ \textsc{FunEval}
$$

$$
\frac{e \rightsquigarrow e'}{e\ T \rightsquigarrow e'\ T}\ \textsc{CongForall}
\qquad
\frac{}{(\Lambda \alpha.\, e)\ T \rightsquigarrow [T/\alpha]e}\ \textsc{ForallEval}
$$

$$
\frac{e_{\mathrm{impl}} \rightsquigarrow e_{\mathrm{impl}}'}{\mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, e_{\mathrm{impl}}) \rightsquigarrow \mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, e_{\mathrm{impl}}')}\ \textsc{CongExists}
$$

$$
\frac{e_{\mathrm{impl}} \rightsquigarrow e_{\mathrm{impl}}'}{\mathsf{let\ pack}(\alpha, x) = e_{\mathrm{impl}}\ \mathsf{in}\ e_{\mathrm{use}} \rightsquigarrow \mathsf{let\ pack}(\alpha, x) = e_{\mathrm{impl}}'\ \mathsf{in}\ e_{\mathrm{use}}}\ \textsc{CongExistsUnpack}
$$

$$
\frac{}{\mathsf{let\ pack}(\alpha, x) = \mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, v_{\mathrm{impl}})\ \mathsf{in}\ e_{\mathrm{use}} \rightsquigarrow [T_{\mathrm{conc}}/\alpha, v_{\mathrm{impl}}/x]e_{\mathrm{use}}}\ \textsc{ExistsEval}
$$

Church encodings

Pairs

$T_1 \times T_2 \triangleq \forall\alpha.\, (T_1 \to T_2 \to \alpha) \to \alpha$

$\langle e_1, e_2\rangle \triangleq \Lambda\alpha.\, \lambda k : T_1 \to T_2 \to \alpha.\, k\ e\ e'$

$\mathsf{fst}\ e \triangleq e\ T_1\ (\lambda x : T_1.\, \lambda y : T_2.\, x)$

$\mathsf{snd}\ e \triangleq e\ T_2\ (\lambda x : T_1.\, \lambda y : T_2.\, y)$

Sums

$T_1 + T_2 \triangleq \forall\alpha.\, (T_1 \to \alpha) \to (T_2 \to \alpha) \to \alpha$

$\mathsf{L}\ e \triangleq \Lambda\alpha.\, \lambda f : T_1 \to \alpha.\, \lambda g : T_2 \to \alpha.\, f\ e$

$\mathsf{R}\ e \triangleq \Lambda\alpha.\, \lambda f : T_1 \to \alpha.\, \lambda g : T_2 \to \alpha.\, g\ e$

$\mathsf{case}(e, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2) : T \triangleq e\ T\ (\lambda x : T_1 \to T.\, e_1)\ (\lambda y : T_2 \to T.\, e_2)$

Existential types

$\exists\alpha.\, T_{\mathrm{sig}} \triangleq \forall\beta.\, (\forall\alpha.\, T_{\mathrm{sig}} \to \beta) \to \beta$

$\mathsf{pack}_{\alpha_{\mathrm{abs}}.T_{\mathrm{sig}}}(T_{\mathrm{conc}}, e_{\mathrm{impl}}) \triangleq \Lambda\beta.\, \lambda k : \forall\alpha_{\mathrm{abs}}.\, T_{\mathrm{sig}} \to \beta.\, k\ T_{\mathrm{conc}}\ e_{\mathrm{impl}}$

$\mathsf{let}\ \mathsf{pack}(\alpha, x) = e_{\mathrm{impl}}\ \mathsf{in}\ e_{\mathrm{use}} : T_{\mathrm{use}} \triangleq e_{\mathrm{impl}}\ T_{\mathrm{use}}\ (\Lambda\alpha.\, \lambda x : T_{\mathrm{sig}}.\, e_{\mathrm{use}})$

Booleans

$\mathsf{bool} \triangleq \forall\alpha.\, \alpha \to \alpha \to \alpha$

$\mathsf{True} \triangleq \Lambda\alpha.\, \lambda x : \alpha.\, \lambda y : \alpha.\, x$

$\mathsf{False} \triangleq \Lambda\alpha.\, \lambda x : \alpha.\, \lambda y : \alpha.\, y$

$\mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : T \triangleq e\ T\ e_1\ e_2$

Natural numbers

$\mathbb{N} \triangleq \forall\alpha.\, \alpha \to (\alpha \to \alpha) \to \alpha$

$\mathsf{zero} \triangleq \Lambda\alpha.\, \lambda z : \alpha.\, \lambda s : \alpha \to \alpha.\, z$

$\mathsf{succ}(e) \triangleq \Lambda\alpha.\, \lambda z : \alpha.\, \lambda s : \alpha \to \alpha.\, s\ (e\ \alpha\ z\ s)$

$\mathsf{iter}(e, \mathsf{zero} \to e_{\mathsf{z}}, \mathsf{succ}(x) \to e_{\mathsf{s}}) : T \triangleq e\ T\ e_{\mathsf{z}}\ (\lambda x : T.\, e_{\mathsf{s}})$

Lists

$\mathsf{list}\ T \triangleq \forall\alpha.\, \alpha \to (T \to \alpha \to \alpha) \to \alpha$

$[] \triangleq \Lambda\alpha.\, \lambda n : \alpha.\, \lambda c : T \to \alpha \to \alpha.\, n$

$e :: e' \triangleq \Lambda\alpha.\, \lambda n : \alpha.\, \lambda c : T \to \alpha \to \alpha.\, c\ e\ (e'\ \alpha\ n\ c)$

$\mathsf{fold}(e, [] \to e_{\mathsf{n}}, x :: r \to e_{\mathsf{c}}) : T' \triangleq e\ T'\ e_{\mathsf{n}}\ (\lambda x : T.\, \lambda r : T'.\, e_{\mathsf{c}})$

3 Monadic λ-Calculus for State

Syntax

Types	T	$::=$	$1 \mid \mathbb{N} \mid T_1 \to T_2 \mid \mathsf{ref}\ T \mid \mathsf{M}\ T$
Pure Terms	e	$::=$	$\langle\rangle \mid n \mid \lambda x : T.\ e \mid e_1\ e_2 \mid l \mid \{t\}$
Impure Terms	t	$::=$	$\mathsf{new}\ e \mid !e \mid e := e' \mid \mathsf{let}\ x = e; t \mid \mathsf{return}\ e$
Values	v	$::=$	$\langle\rangle \mid n \mid \lambda x : T.\ e \mid l \mid \{t\}$
Stores	σ	$::=$	$\cdot \mid \sigma, l : v$
Contexts	Γ	$::=$	$\cdot \mid \Gamma, x : T$
Store Typings	Σ	$::=$	$\cdot \mid \Sigma, l : T$

Typing rules

Pure terms

$$\frac{x : T \in \Gamma}{\Sigma; \Gamma \vdash x : T}\ \textsc{Hyp} \qquad \frac{}{\Sigma; \Gamma \vdash \langle\rangle : 1}\ 1\mathrm{I} \qquad \frac{}{\Sigma; \Gamma \vdash n : \mathbb{N}}\ \mathbb{N}\mathrm{I}$$

$$\frac{\Sigma; \Gamma, x : T \vdash e : T'}{\Sigma; \Gamma \vdash \lambda x : T.\ e : T \to T'}\ {\to}\mathrm{I} \qquad \frac{\Sigma; \Gamma \vdash e_1 : T \to T' \qquad \Sigma; \Gamma \vdash e_2 : T}{\Sigma; \Gamma \vdash e_1\ e_2 : T'}\ {\to}\mathrm{E}$$

$$\frac{l : T \in \Sigma}{\Sigma; \Gamma \vdash l : \mathsf{ref}\ T}\ \textsc{RefBar} \qquad \frac{\Sigma; \Gamma \vdash t \div T}{\Sigma; \Gamma \vdash \{t\} : \mathsf{M}\ T}\ \mathrm{MI}$$

Impure terms

$$\frac{\Sigma; \Gamma \vdash e : T}{\Sigma; \Gamma \vdash \mathsf{new}\ e \div \mathsf{ref}\ T}\ \textsc{RefI} \qquad \frac{\Sigma; \Gamma \vdash e : \mathsf{ref}\ T}{\Sigma; \Gamma \vdash !e \div T}\ \textsc{RefGet} \qquad \frac{\Sigma; \Gamma \vdash e : \mathsf{ref}\ T \qquad \Sigma; \Gamma \vdash e' : T}{\Sigma; \Gamma \vdash e := e' \div 1}\ \textsc{RefSet}$$

$$\frac{\Sigma; \Gamma \vdash e : T}{\Sigma; \Gamma \vdash \mathsf{return}\ e \div T}\ \textsc{MRet} \qquad \frac{\Sigma; \Gamma \vdash e : \mathsf{M}\ T \qquad \Sigma; \Gamma, x : T \vdash t \div T'}{\Sigma; \Gamma \vdash \mathsf{let}\ x = e; t \div T'}\ \textsc{MLet}$$

Store and configuration

$$\frac{}{\Sigma \vdash \cdot : \cdot}\ \textsc{StoreNil} \qquad \frac{\Sigma \vdash \sigma' : \Sigma' \qquad \Sigma; \cdot \vdash v : T}{\Sigma \vdash (\sigma', l : v) : (\Sigma', l : T)}\ \textsc{StoreCons} \qquad \frac{\Sigma \vdash \sigma : \Sigma \qquad \Sigma; \cdot \vdash t \div T}{\langle \sigma; t \rangle : \langle \Sigma; T \rangle}\ \textsc{ConfigOK}$$

Operational semantics

Pure terms

$$\frac{e_1 \rightsquigarrow e_1'}{e_1\ e_2 \rightsquigarrow e_1'\ e_2} \qquad \frac{e_2 \rightsquigarrow e_2'}{v\ e_2 \rightsquigarrow v\ e_2'} \qquad \frac{}{(\lambda x : T.\ e)\ v \rightsquigarrow [v/x]e}$$

Impure terms

$$\frac{e \rightsquigarrow e'}{\langle \sigma; \mathsf{new}\ e \rangle \rightsquigarrow \langle \sigma; \mathsf{new}\ e' \rangle} \qquad \frac{l \notin \mathrm{dom}(\sigma)}{\langle \sigma; \mathsf{new}\ v \rangle \rightsquigarrow \langle (\sigma, l : v); \mathsf{return}\ l \rangle}$$

$$\frac{e \rightsquigarrow e'}{\langle \sigma; !e \rangle \rightsquigarrow \langle \sigma; !e' \rangle} \qquad \frac{l : v \in \mathrm{dom}(\sigma)}{\langle \sigma; !l \rangle \rightsquigarrow \langle \sigma; \mathsf{return}\ v \rangle}$$

$$\frac{e_1 \rightsquigarrow e_1'}{\langle \sigma; e_1 := e_2 \rangle \rightsquigarrow \langle \sigma; e_1' := e_2 \rangle} \qquad \frac{e_2 \rightsquigarrow e_2'}{\langle \sigma; v := e_2 \rangle \rightsquigarrow \langle \sigma; v := e_2' \rangle}$$

$$\frac{}{\langle(\sigma, l : v, \sigma'); l := v\rangle \rightsquigarrow \langle(\sigma, l : v', \sigma'); \mathsf{return}\ \langle\rangle\rangle}$$

$$\frac{e \rightsquigarrow e'}{\langle\sigma; \mathsf{return}\ e\rangle \rightsquigarrow \langle\sigma; \mathsf{return}\ e'\rangle} \qquad \frac{e \rightsquigarrow e'}{\langle\sigma; \mathsf{let}\ x = e; t\rangle \rightsquigarrow \langle\sigma; \mathsf{let}\ x = e'; t\rangle}$$

$$\frac{}{\langle\sigma; \mathsf{let}\ x = \{\mathsf{return\ v}\}; t\rangle \rightsquigarrow \langle\sigma; [v/x]t\rangle} \qquad \frac{\langle\sigma; t_1\rangle \rightsquigarrow \langle\sigma'; t_1'\rangle}{\langle\sigma; \mathsf{let}\ x = \{t_1\}; t_2\rangle \rightsquigarrow \langle\sigma'; \mathsf{let}\ x = \{t_1'\}; t_2\rangle}$$

4 Monadic λ-Calculus for I/O

Syntax

Types	T	::=	$1 \mid \mathbb{N} \mid T_1 \to T_2 \mid \mathsf{M}_{\mathsf{IO}}\ T$
Pure Terms	e	::=	$\langle\rangle \mid n \mid \lambda x : T.\ e \mid e_1\ e_2 \mid l \mid \{t\}$
Impure Terms	t	::=	$\mathsf{print}\ e \mid \mathsf{let}\ x = e; t \mid \mathsf{return}\ e$
Values	v	::=	$\langle\rangle \mid n \mid \lambda x : T.\ e \mid \{t\}$
Output Tokens	ω	::=	$\cdot \mid n :: \omega$
Contexts	Γ	::=	$\cdot \mid \Gamma, x : T$

Typing rules

Pure terms

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \textsc{Hyp} \qquad \frac{}{\Gamma \vdash \langle\rangle : 1}\ 1\mathrm{I} \qquad \frac{}{\Gamma \vdash n : \mathbb{N}}\ \mathbb{N}\mathrm{I}$$

$$\frac{\Gamma, x : T \vdash e : T'}{\Gamma \vdash \lambda x : T.\ e : T \to T'}\ {\to}\mathrm{I} \qquad \frac{\Gamma \vdash e_1 : T \to T' \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1\ e_2 : T'}\ {\to}\mathrm{E} \qquad \frac{\Gamma \vdash t \div T}{\Gamma \vdash \{t\} : \mathsf{M}_{\mathsf{IO}}\ T}\ \mathrm{MI}$$

Impure terms

$$\frac{\Gamma \vdash e : \mathbb{N}}{\Gamma \vdash \mathsf{print}\ e \div 1}\ \textsc{MPrint} \qquad \frac{\Gamma \vdash e : T}{\Gamma \vdash \mathsf{return}\ e \div T}\ \textsc{MRet} \qquad \frac{\Gamma \vdash e : \mathsf{M}_{\mathsf{IO}}\ T \qquad \Gamma, x : T \vdash t \div T'}{\Gamma \vdash \mathsf{let}\ x = e; t \div T'}\ \textsc{MLet}$$

Operational semantics

Pure terms

$$\frac{e_1 \rightsquigarrow e_1'}{e_1\ e_2 \rightsquigarrow e_1'\ e_2} \qquad \frac{e_2 \rightsquigarrow e_2'}{v\ e_2 \rightsquigarrow v\ e_2'} \qquad \frac{}{(\lambda x : T.\ e)\ v \rightsquigarrow [v/x]e}$$

Impure terms

$$\frac{e \rightsquigarrow e'}{\langle\omega; \mathsf{print}\ e\rangle \rightsquigarrow \langle\omega; \mathsf{print}\ e'\rangle} \qquad \frac{}{\langle\omega; \mathsf{print}\ n\rangle \rightsquigarrow \langle(n :: \omega); \mathsf{return}\ \langle\rangle\rangle}$$

$$\frac{e \rightsquigarrow e'}{\langle\omega; \mathsf{return}\ e\rangle \rightsquigarrow \langle\omega; \mathsf{return}\ e'\rangle} \qquad \frac{e \rightsquigarrow e'}{\langle\omega; \mathsf{let}\ x = e; t\rangle \rightsquigarrow \langle\omega; \mathsf{let}\ x = e'; t\rangle}$$

$$\frac{}{\langle\omega; \mathsf{let}\ x = \{\mathsf{return\ v}\}; t\rangle \rightsquigarrow \langle\omega; [v/x]t\rangle} \qquad \frac{\langle\omega; t_1\rangle \rightsquigarrow \langle\omega'; t_1'\rangle}{\langle\omega; \mathsf{let}\ x = \{t_1\}; t_2\rangle \rightsquigarrow \langle\omega'; \mathsf{let}\ x = \{t_1'\}; t_2\rangle}$$