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	$::=$	$x \mid \langle\rangle \mid \langle e_1, e_2\rangle \mid \mathsf{fst}\ e \mid \mathsf{snd}\ e \mid \mathsf{L}\ e \mid \mathsf{R}\ e \mid \mathsf{case}(e, \mathsf{L}\ x \to e_1, \mathsf{R}\ y \to e_2)$
		$\mid$	$\lambda x : T.\ e \mid e_1\ e_2 \mid \mathsf{abort}$
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

Types	T	$::=$	$\tau \mid T_1 \to T_2 \mid \forall\tau.\ T \mid \exists\tau.\ T$
Terms	e	$::=$	$x \mid \lambda x : T.\ e \mid e_1\ e_2 \mid \Lambda\tau.\ e \mid e\ T \mid \mathsf{pack}_{\tau.T'}(T, e) \mid \mathsf{let\ pack}(\tau, x) = e \mathsf{\ in\ } e'$
Values	v	$::=$	$\lambda x : T.\ e \mid \Lambda\tau.\ e \mid \mathsf{pack}_{\tau.T'}(T, v)$
Type Contexts	Θ	$::=$	$\cdot \mid \Theta, \tau$
Term Contexts	Γ	$::=$	$\cdot \mid \Gamma, x : T$

Well-formedness of types

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

Well-formedness of term contexts

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

Typing rules

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

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

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

$$\frac{\Theta, \tau \vdash T' \text{ type} \qquad \Theta \vdash T \text{ type} \qquad \Theta; \Gamma \vdash e : [T/\tau]T'}{\Theta; \Gamma \vdash \mathsf{pack}_{\tau.T'}(T, e) : \exists\tau.\ T'}\ \exists\mathrm{I}$$

$$\frac{\Theta; \Gamma \vdash e : \exists\tau.\ T \qquad \Theta, \tau; \Gamma, x : T \vdash e' : T' \qquad \Theta \vdash T' \text{ type}}{\Theta; \Gamma \vdash \mathsf{let\ pack}(\tau, x) = e \mathsf{\ in\ } e' : T'}\ \exists\mathrm{E}$$

Operational semantics

(Cong: congruence rule, Eval: evaluation rule)

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

$$\frac{e \leadsto e'}{e\ T \leadsto e'\ T}\ \textsc{CongForall} \qquad \frac{}{(\Lambda\tau.\ e)\ T \leadsto [T/\tau]e}\ \textsc{ForallEval}$$

$$\frac{e \leadsto e'}{\mathsf{pack}_{\tau.T'}(T, e) \leadsto \mathsf{pack}_{\tau.T'}(T, e')}\ \textsc{CongExists}$$

$$\frac{e_1 \leadsto e_1'}{\mathsf{let\ pack}(\tau, x) = e_1 \mathsf{\ in\ } e_2 \leadsto \mathsf{let\ pack}(\tau, x) = e_1' \mathsf{\ in\ } e_2}\ \textsc{CongExistsUnpack}$$

$$\frac{}{\mathsf{let\ pack}(\tau, x) = \mathsf{pack}_{\tau.T'}(T, v) \mathsf{\ in\ } e \leadsto [T/\tau, v/x]e}\ \textsc{ExistsEval}$$

Church encodings

Pairs

$T_1 \times T_2 \triangleq \forall\tau.\ (T_1 \to T_2 \to \tau) \to \tau$

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

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

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

Sums

$T_1 + T_2 \triangleq \forall\tau.\ (T_1 \to \tau) \to (T_2 \to \tau) \to \tau$

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

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

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

Existential types

$\exists\tau.\ T' \triangleq \forall\pi.\ (\forall\tau.\ T' \to \pi) \to \pi$

$\mathsf{pack}_{\tau.T'}(T, e) \triangleq \Lambda\pi.\ \lambda k : \forall\tau.\ T' \to \pi.\ k\ T\ e$

$\mathsf{let\ pack}(\tau, x) = e\ \mathsf{in}\ e' : T' \triangleq e\ T'\ (\Lambda\tau.\ \lambda x : T.\ e')$

Booleans

$\mathsf{bool} \triangleq \forall\tau.\ \tau \to \tau \to \tau$

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

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

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

Natural numbers

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

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

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

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

Lists

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

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

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

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