LAMBDA

MODULE LAMBDA

SYNTAX *Exp* ::= *Id*
| *λId.Exp*
| *Exp Exp* [strict]
| (*Exp*) [bracket]

CONFIGURATION:

$$\left\langle \langle \$PGM{:}Exp \rangle_{\mathsf{k}} \; \langle \bullet_{Map} \rangle_{\mathsf{env}} \; \langle \bullet_{Map} \rangle_{\mathsf{store}} \right\rangle_{\mathsf{T}}$$

SYNTAX *Val* ::= `closure` (*Map*, *Id*, *Exp*)

SYNTAX *Exp* ::= *Val*

SYNTAX *KResult* ::= *Val*

RULE $\left\langle \dfrac{\lambda X{:}Id.E}{\mathtt{closure}\,(\rho, X, E)} \right\rangle_{\mathsf{k}} \; \langle \rho \rangle_{\mathsf{env}}$

RULE $\left\langle \dfrac{\mathtt{closure}\,(\rho, X, E) \;\; V{:}Val}{E \curvearrowright \mathtt{env}\,(\rho')} \right\rangle_{\mathsf{k}} \; \left\langle \dfrac{\rho'}{\rho[N \,/\, X]} \right\rangle_{\mathsf{env}} \; \left\langle \dfrac{\bullet_{Map}}{(N \mapsto V)} \right\rangle_{\mathsf{store}}$ requires `fresh` $(N{:}Nat)$

RULE $\left\langle \dfrac{X}{V} \right\rangle_{\mathsf{k}} \; \langle X \mapsto N \rangle_{\mathsf{env}} \; \langle N \mapsto V \rangle_{\mathsf{store}}$

SYNTAX *K* ::= `env` (*Map*)

RULE $\left\langle —{:}\mathrm{Val} \curvearrowright \dfrac{\mathtt{env}\,(\rho)}{\bullet_{K}} \right\rangle_{\mathsf{k}} \; \left\langle \dfrac{—}{\rho} \right\rangle_{\mathsf{env}}$ [structural]

SYNTAX *Val* ::= *Int*
| *Bool*

SYNTAX *Exp* ::= *Exp* ∗ *Exp* [strict]
| *Exp* / *Exp* [strict]
| *Exp* + *Exp* [strict]
| *Exp* <= *Exp* [strict]

RULE $\dfrac{I1{:}Int * I2{:}Int}{I1 *_{Int} I2}$

RULE $\dfrac{I1{:}Int \,/\, I2{:}Int}{I1 \div_{Int} I2}$

RULE $\dfrac{I1{:}Int + I2{:}Int}{I1 +_{Int} I2}$

RULE $\dfrac{I1{:}Int \;\texttt{<=}\; I2{:}Int}{I1 \leq_{Int} I2}$

SYNTAX *Exp* ::= `if` *Exp* `then` *Exp* `else` *Exp* [strict(1)]

RULE $\dfrac{\mathtt{if\ true\ then}\ E\ \mathtt{else}\ —}{E}$

RULE $\dfrac{\mathtt{if\ false\ then}\ —\ \mathtt{else}\ E}{E}$

SYNTAX *Exp* ::= `let` *Id* = *Exp* `in` *Exp*

RULE $\dfrac{\mathtt{let}\ X = E\ \mathtt{in}\ E'{:}Exp}{(\lambda X.E')\;\; E}$ [macro]

SYNTAX *Exp* ::= `letrec` *Id Id* = *Exp* `in` *Exp*
| *μId.Exp*

RULE $\dfrac{\mathtt{letrec}\ F{:}Id\;\; X = E\ \mathtt{in}\ E'}{\mathtt{let}\ F = \mu F.\lambda X.E\ \mathtt{in}\ E'}$ [macro]

SYNTAX *Exp* ::= `muclosure` (*Map*, *Exp*)

RULE $\left\langle \dfrac{\mu X.E}{\mathtt{muclosure}\,(\rho[N \,/\, X], E)} \right\rangle_{\mathsf{k}} \; \langle \rho \rangle_{\mathsf{env}} \; \left\langle \dfrac{\bullet_{Map}}{(N \mapsto \mathtt{muclosure}\,(\rho[N \,/\, X], E))} \right\rangle_{\mathsf{store}}$ requires `fresh` $(N{:}Nat)$ [structural]

RULE $\left\langle \dfrac{\mathtt{muclosure}\,(\rho, E)}{E \curvearrowright \mathtt{env}\,(\rho')} \right\rangle_{\mathsf{k}} \; \left\langle \dfrac{\rho'}{\rho} \right\rangle_{\mathsf{env}}$

SYNTAX *Exp* ::= `callcc` *Exp* [strict]

SYNTAX *Val* ::= `cc` (*Map*, *K*)

RULE $\left\langle \dfrac{\mathtt{callcc}\ V{:}Val}{V \;\; \mathtt{cc}\,(\rho, K)} \curvearrowright K \right\rangle_{\mathsf{k}} \; \langle \rho \rangle_{\mathsf{env}}$

RULE $\left\langle \dfrac{\mathtt{cc}\,(\rho, K)\;\; V \curvearrowright —}{V \curvearrowright K} \right\rangle_{\mathsf{k}} \; \left\langle \dfrac{—}{\rho} \right\rangle_{\mathsf{env}}$

END MODULE