LAMBDA

MODULE LAMBDA

SYNTAX $Exp ::= Id$
| $\lambda Id.Exp$
| $Exp\ \ Exp$ [strict]
| (Exp) [bracket]

CONFIGURATION:

T: k: $\$PGM{:}Exp$ | env: $\bullet_{Map}$ | store: $\bullet_{Map}$

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

SYNTAX $Exp ::= Val$

SYNTAX $KResult ::= Val$

RULE k: $\dfrac{\lambda X{:}Id.E}{\texttt{closure}\ (\rho, X, E)}$ ... | env: ρ

RULE k: $\dfrac{\texttt{closure}\ (\rho, X, E)\ \ V{:}Val}{E \curvearrowright \texttt{env}\ (\rho')}$ ... | env: $\dfrac{\rho'}{\rho[N \ / \ X]}$ | store: ... $\dfrac{\bullet_{Map}}{(N \mapsto V)}$ ... requires `fresh` $(N{:}Nat)$

RULE k: $\dfrac{X}{V}$ ... | env: ... $X \mapsto N$ ... | store: ... $N \mapsto V$ ...

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

RULE k: $\_{:}Val \curvearrowright \dfrac{\texttt{env}\ (\rho)}{\bullet_K}$ ... | env: $\dfrac{\_}{\rho}$ [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 $\dfrac{\texttt{if true then } E \texttt{ else } \_}{E}$

RULE $\dfrac{\texttt{if false then } \_ \texttt{ else } E}{E}$

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

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

SYNTAX $Exp ::=$ `letrec` $Id\ \ Id$ = Exp `in` Exp

SYNTAX $Id ::=$ `$x`
| `$y`

RULE $\dfrac{\texttt{letrec } F{:}Id\ \ X{:}Id = E \texttt{ in } E'}{\texttt{let } F = (\lambda \$x.((\lambda F.\lambda X.E)\ \ (\lambda \$y.(\$x\ \ \$x\ \ \$y))))\ \ (\lambda \$x.((\lambda F.\lambda X.E)\ \ (\lambda \$y.(\$x\ \ \$x\ \ \$y)))) \texttt{ in } E'}$ [macro]

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

SYNTAX $Val ::=$ `cc` (K)

RULE k: $\dfrac{\texttt{callcc}\ V{:}Val}{V\ \ \texttt{cc}\ (K)} \curvearrowright K$

RULE k: $\dfrac{\texttt{cc}\ (K)\ \ V \curvearrowright \_}{V \curvearrowright K}$

END MODULE