

LAMBDA

MODULE LAMBDA

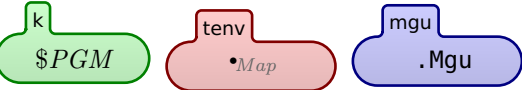
SYNTAX $Exp ::= Int$
| $Bool$
| Id
| (Exp) [bracket]
| $Exp\ Exp$ [strict]
| $Exp * Exp$ [strict]
| Exp / Exp [strict]
| $Exp + Exp$ [strict]
| $Exp <= Exp$ [strict]
| $\text{lambda } Id . Exp$
| $\text{if } Exp \text{ then } Exp \text{ else } Exp$ [strict]
| $\text{let } Id = Exp \text{ in } Exp$ [strict(2)]
| $\text{letrec } Id\ Id = Exp \text{ in } Exp$
| $\text{mu } Id . Exp$

SYNTAX $Type ::= \text{int}$
| bool
| $Type \rightarrow Type$
| $(Type)$ [bracket]

SYNTAX $Exp ::= Type$

SYNTAX $KResult ::= Type$

CONFIGURATION:



RULE $\frac{I}{\text{int}}$

RULE $\frac{B}{\text{bool}}$

RULE

RULE $\frac{T1 * T2}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\frac{T1 / T2}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\frac{T1 + T2}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\frac{T1 <= T2}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{bool}}$

SYNTAX $Exp ::= Exp \rightarrow Exp$ [strict]

RULE

RULE $\frac{T1\ T2}{T1 = (T2 \rightarrow T) \curvearrowright T}$ requires fresh (T)

RULE $\frac{\text{if } T \text{ then } T1 \text{ else } T2}{T = \text{bool} \curvearrowright T1 = T2 \curvearrowright T1}$

SYNTAX $TypeSchema ::= (\text{forall } Set)Type$ [binder]

RULE

RULE

RULE $\frac{\text{letrec } F\ X = E \text{ in } E'}{\text{let } F = \text{mu } F . \text{lambda } X . E \text{ in } E'}$

RULE

SYNTAX $K ::= Type = Type$

RULE

RULE

SYNTAX $K ::= \text{tenv } (Map)$

RULE

END MODULE

[macro]