

# LAMBDA

MODULE LAMBDA

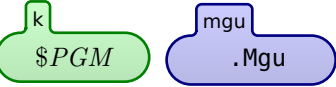
SYNTAX     $Exp ::= Int$   
          |  $Bool$   
          |  $Id$   
          |  $(Exp)$  [bracket]  
          |  $Exp\ Exp$  [strict]  
          |  $Exp * Exp$  [strict]  
          |  $Exp / Exp$  [strict]  
          |  $Exp + Exp$  [strict]  
          |  $Exp <= Exp$  [strict]  
         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

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

SYNTAX     $Exp ::= Type$

SYNTAX     $KResult ::= Type$

CONFIGURATION:



RULE     $\frac{I}{\text{int}}$

RULE     $\frac{B}{\text{bool}}$

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     $\frac{\text{lambda } X . E \quad \text{requires fresh } (T)}{T \rightarrow E[T / X]}$

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

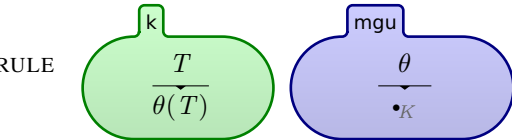
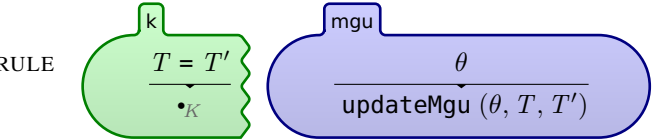
RULE     $\frac{\text{if } T \text{ then } T1 \text{ else } T2}{T = \text{bool} \curvearrowright T1 = T2 \curvearrowright T1}$

RULE     $\frac{\text{let } X = E \text{ in } E'}{E'[E / X]}$  [macro]

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

RULE     $\frac{\text{mu } X . E \quad \text{requires fresh } (T)}{(T \rightarrow T)\ E[T / X]}$

SYNTAX     $K ::= Type = Type$



END MODULE