

# LAMBDA

MODULE LAMBDA

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

SYNTAX     $Exp ::= Id$   
          |  $\text{lambda } Id : Type . Exp$  [binder]  
          |  $Exp \ Exp$  [strict]  
          |  $(Exp)$  [bracket]

SYNTAX     $Exp ::= Type$

SYNTAX     $KResult ::= Type$

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

RULE    
$$\frac{\text{lambda } X : T . E}{T \rightarrow E[T / X]}$$

RULE    
$$\frac{(T1 \rightarrow T2) \ T1}{T2}$$

SYNTAX     $Exp ::= Int$   
          |  $Bool$   
          |  $Exp * Exp$  [strict]  
          |  $Exp / Exp$  [strict]  
          |  $Exp + Exp$  [strict]  
          |  $Exp <= Exp$  [strict]

RULE    
$$\frac{}{\text{int}}$$

RULE    
$$\frac{}{\text{bool}}$$

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

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

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

RULE    
$$\frac{\text{int} <= \text{int}}{\text{bool}}$$

SYNTAX     $Exp ::= \text{if } Exp \text{ then } Exp \text{ else } Exp$  [strict]

RULE    
$$\frac{\text{if } \text{bool} \text{ then } T \text{ else } T}{T}$$

SYNTAX     $Exp ::= \text{let } Id : Type = Exp \text{ in } Exp$

RULE    
$$\frac{\text{let } X : T = E \text{ in } E'}{(\text{lambda } X : T . E') \ E}$$
 [macro]

SYNTAX     $Exp ::= \text{letrec } Id : Type \ Id : Type = Exp \text{ in } Exp$   
          |  $\text{mu } Id : Type . Exp$  [binder]

RULE    
$$\frac{\text{letrec } F : T1 \ X : T2 = E \text{ in } E'}{\text{let } F : T1 = \text{mu } F : T1 . \text{lambda } X : T2 . E \text{ in } E'}$$
 [macro]

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

END MODULE