LAMBDA

MODULE LAMBDA

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

SYNTAX   *Type* ::= int
                | bool
                | *Type* -> *Type*
                | (*Type*) [bracket(bracket())]

SYNTAX   *Exp* ::= *Type*

SYNTAX   *Variable* ::= *Id*

SYNTAX   *KResult* ::= *Type*

CONFIGURATION:

$$\boxed{\text{k}}$$
$$\left(\ \$PGM{:}Exp\ \right)$$

RULE $\dfrac{I{:}Int}{\text{int}}$

RULE $\dfrac{B{:}Bool}{\text{bool}}$

RULE $\dfrac{T1{:}Type \ast T2{:}Type}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\dfrac{T1{:}Type \ / \ T2{:}Type}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\dfrac{T1{:}Type + T2{:}Type}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{int}}$

RULE $\dfrac{T1{:}Type <= T2{:}Type}{T1 = \text{int} \curvearrowright T2 = \text{int} \curvearrowright \text{bool}}$

RULE $\dfrac{\text{lambda } X \ . \ E{:}Exp}{E[T \ / \ X] \curvearrowright T{:}Type \text{ -> } \square}$

RULE $\dfrac{T2{:}Type \curvearrowright T1{:}Type \text{ -> } \square}{T1 \text{ -> } T2}$

RULE $\dfrac{T1{:}Type \ \ T2{:}Type}{T1 = (T2 \text{ -> } T{:}Type) \curvearrowright T}$

RULE $\dfrac{\text{if } T{:}Type \text{ then } T1{:}Type \text{ else } T2{:}Type}{T = \text{bool} \curvearrowright T1 = T2 \curvearrowright T1}$

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

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

RULE $\dfrac{\text{mu } X \ . \ E}{(T{:}Type \text{ -> } T) \ (E[T \ / \ X])}$

SYNTAX   *KItem* ::= *Type* = *Type*

RULE $\dfrac{T = T}{\bullet_K}$

END MODULE