

LAMBDA

MODULE LAMBDA

SYNTAX *Val* ::= *Id*
 | $\lambda Id.Exp$ [binder]

SYNTAX *Exp* ::= *Val*
 | *Exp Exp* [strict]
 | (*Exp*) [bracket]

SYNTAX *KResult* ::= *Val*

RULE
$$\frac{(\lambda X.E) \quad V}{E[V / X]}$$

SYNTAX *Val* ::= *Int*
 | *Bool*

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

RULE
$$\frac{I1 * I2}{I1 *_{Int} I2}$$

RULE
$$\frac{I1 / I2}{I1 \div_{Int} I2}$$

RULE
$$\frac{I1 + I2}{I1 +_{Int} I2}$$

RULE
$$\frac{I1 <= I2}{I1 \leq_{Int} I2}$$

SYNTAX *Exp* ::= if *Exp* then *Exp* else *Exp* [strict(1)]

RULE
$$\frac{\text{if true then } E \text{ else } \text{---}}{E}$$

RULE
$$\frac{\text{if false then } \text{---} \text{ else } E}{E}$$

SYNTAX *Exp* ::= let *Id* = *Exp* in *Exp*

RULE
$$\frac{\text{let } X = E \text{ in } E'}{(\lambda X.E') \quad E}$$
 [macro]

SYNTAX *Exp* ::= letrec *Id Id* = *Exp* in *Exp*
 | $\mu Id.Exp$ [binder]

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

RULE
$$\frac{\mu X.E}{E[(\mu X.E) / X]}$$

END MODULE