

# Semantics for MP2

The following rules describe the semantics for the language you are interpreting.

## Expressions:

For IntExp and BoolExp:

$$\frac{}{(t, env) \Downarrow \text{IntVal } t} \quad t \text{ an integer constant}$$

$$\frac{}{(t, env) \Downarrow \text{BoolVal } t} \quad t \text{ an boolean constant}$$

For IntOpExp:

$$\frac{(e_1, env) \Downarrow v_1 \quad (e_2, env) \Downarrow v_2}{(e_1 \oplus e_2, env) \Downarrow (\text{IntVal } (v_1 \oplus v_2))} \oplus \text{ an integer operation besides division}$$

$$\frac{(e_1, env) \Downarrow v_1 \quad (e_2, env) \Downarrow \text{IntVal } 0}{(e_1 / e_2, env) \Downarrow (\text{ExnVal } \text{"Division by 0"})}$$

$$\frac{(e_1, env) \Downarrow v_1 \quad (e_2, env) \Downarrow v_2}{(e_1 / e_2, env) \Downarrow (\text{IntVal } (v_1 / v_2))} \quad v_2 \neq \text{IntVal } 0$$

$$\frac{}{(e_1 \oplus e_2, env) \Downarrow \text{ExnVal } \text{"No matching operator"}} \oplus \text{ not a valid integer operation}$$

For BoolOpExp:

$$\frac{(e_1, env) \Downarrow v_1 \quad (e_2, env) \Downarrow v_2}{(e_1 \oplus e_2, env) \Downarrow (\text{BoolVal } (v_1 \oplus v_2))} \oplus \text{ a boolean operation}$$

$$\frac{}{(e_1 \oplus e_2, env) \Downarrow \text{ExnVal } \text{"No matching operator"}} \oplus \text{ not a valid boolean operation}$$

For CompOpExp:

$$\frac{(e_1, env) \Downarrow v_1 \quad (e_2, env) \Downarrow v_2}{(e_1 \oplus e_2, env) \Downarrow (\text{BoolVal } (v_1 \oplus v_2))} \oplus \text{ an integer comparison operation}$$

$$\frac{}{(e_1 \oplus e_2, env) \Downarrow \text{ExnVal } \text{"No matching operator"}} \oplus \text{ not a valid integer comparison operation}$$

For IfExp:

$$\frac{(e_1, env) \Downarrow \text{true} \quad (e_2, env) \Downarrow v_2}{(\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \text{ fi}, env) \Downarrow v_2}$$

$$\frac{(e_1, env) \Downarrow \text{false} \quad (e_3, env) \Downarrow v_3}{(\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \text{ fi}, env) \Downarrow v_3}$$

$$\frac{(e_1, env) \Downarrow v}{(\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \text{ fi}, env) \Downarrow \text{ExnVal } \text{"Condition is not a Bool"}} \quad v \text{ is not a BoolVal}$$

For VarExp:

$$\frac{}{(x, env) \Downarrow v} \quad x \in env, v = env(x)$$

$$\frac{}{(x, env) \Downarrow \text{ExnVal } \text{"No match in env"}} \quad x \notin env$$

For FunExp:

$$\frac{}{(fn[x_1, \dots, x_n] \text{ bodyend}, env) \Downarrow \langle [x_1, \dots, x_n], \text{body}, env \rangle}$$

For AppExp:

$$\frac{(f, env) \Downarrow \langle x_1, \dots, x_n, \text{body}, env' \rangle \quad (e_1, env) \Downarrow v_1 \dots (e_n, env) \Downarrow v_n \quad (\text{body}, env' + \{x_1 \mapsto v_1, \dots, x_n \mapsto v_n\}) \Downarrow v_f}{(\text{apply } f(e_1, \dots, e_n), env) \Downarrow v_f} \quad n \geq 0$$

$$\frac{(f, env) \Downarrow v}{(\text{apply } f(e_1, \dots, e_n), env) \Downarrow \text{ExnVal } \text{"Apply to non-closure"}} \quad n \geq 0, v \text{ is not a CloVal}$$

For LetExp:

$$\frac{(e_1, env) \Downarrow v_1 \quad \dots \quad (e_n, env) \Downarrow v_n \quad (e_f, env + \{x_1 \mapsto v_1, \dots, x_n \mapsto v_n\}) \Downarrow v_f}{(\text{let } [x_1 := e_1, \dots, x_n := e_n] e_f \text{end}, env) \Downarrow v_f} \quad n \geq 0$$

**Statements:**

For SeqStmnt:

$$\frac{(S_1, penv, env) \Downarrow (p_1, penv', env') \quad \dots \quad (S_n, penv', env') \Downarrow (p_n, penv'', env'')}{(\text{do } S_1; \dots; S_n \text{od}, penv, env) \Downarrow (p_1 ++ \dots ++ p_n, penv'', env')}$$

For IfStmnt:

$$\frac{(e_1, env) \Downarrow \text{true} \quad (S_2, penv, env) \Downarrow (p_2, penv', env')}{(\text{if } e_1 \text{ then } S_2 \text{ else } S_3 \text{ fi}, penv, env) \Downarrow (p_2, penv', env')}$$

$$\frac{(e_1, env) \Downarrow \text{false} \quad (S_3, penv, env) \Downarrow (p_3, penv', env')}{(\text{if } e_1 \text{ then } S_2 \text{ else } S_3 \text{ fi}, penv, env) \Downarrow (p_3, penv', env')}$$

$$\frac{(e_1, env) \Downarrow v}{(\text{if } e_1 \text{ then } S_2 \text{ else } S_3 \text{ fi}, penv, env) \Downarrow (\text{"exn: Condition is not a Bool"}, penv, env)} \quad v \text{ is not a BoolVal}$$

For SetStmnt:

$$\frac{(e, env) \Downarrow v_1}{(x := e, penv, env) \Downarrow (\text{"", penv, env} + \{x \mapsto v_1\})}$$

For ProcedureStmnt:

$$\frac{}{(\text{procedure } f(ps) \text{ body}; \text{endproc}; penv, env) \Downarrow (\text{"", penv} + \{f \mapsto (\text{ProcedureStmnt } f \text{ ps body})\}, env)}$$

For CallStmnt:

$$\frac{A \quad (\text{body}, penv, env' = env + \{x_1 \mapsto v_1, \dots, x_n \mapsto v_n\}) \Downarrow (p_f, penv', env'')}{(\text{call } f(e_1, \dots, e_n), penv, env) \Downarrow (p_f, penv', env'')} \quad n \geq 0, f \in penv$$

where

$$A = (e_1, env) \Downarrow v_1 \quad \dots \quad (e_n, env) \Downarrow v_n$$

$$penv(f) = (\text{ProcedureStmnt } f \text{ ps body})$$

$$ps = x_1 \dots x_n$$

$$\frac{}{(\text{call } f(e_1, \dots, e_n), penv, env) \Downarrow (\text{"Procedure } f \text{ undefined"}, penv, env)} \quad f \notin penv$$