Semantics for MP2

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

Expressions:

For `IntExp` and `BoolExp`:

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

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

For `IntOpExp`:

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

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

$$\frac{(e_1, env) \Downarrow v_1 \qquad (e_2, env) \Downarrow v_2}{(e_1 \,/\, e_2, env) \Downarrow (\texttt{IntVal}v_1 \,/\, v_2)} \; v_2 \neq \texttt{IntVal 0}$$

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

For `BoolOpExp`:

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

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

For `CompOpExp`:

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

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

For `IfExp`:

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

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

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

For VarExp:

$$\frac{}{(\texttt{x}, \mathit{env}) \Downarrow v}\; x \in \mathit{env}, v = \mathit{env}(x)$$

$$\frac{}{(\texttt{x}, \mathit{env}) \Downarrow \texttt{ExnVal "No match in env"}}\; x \notin \mathit{env}$$

For FunExp:

$$\frac{}{(\texttt{fn}[x_1, ..., x_n]\ \mathit{body}\ \ \texttt{end;}, \mathit{env}) \Downarrow \langle [x_1, ..., x_n], \mathit{body}, \mathit{env} \rangle}$$

For AppExp:

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

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

For LetExp:

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

Statements:

For SeqStmt:

$$\frac{(S_1, \mathit{penv}, \mathit{env}) \Downarrow (p_1, \mathit{penv}', \mathit{env}') \qquad (S_2, \mathit{penv}', \mathit{env}') \Downarrow (p_2, \mathit{penv}'', \mathit{env}'')}{(S_1; S_2, \mathit{penv}, \mathit{env}) \Downarrow (p_1\ \ \texttt{++}\ \ p_2, \mathit{penv}'', \mathit{env}'')}$$

For IfStmt:

$$\frac{(e_1, \mathit{env}) \Downarrow \texttt{true} \qquad (S_2, \mathit{penv}, \mathit{env}) \Downarrow (p_2, \mathit{penv}', \mathit{env}')}{(\texttt{if}\ e_1\ \texttt{then}\ S_2\ \texttt{else}\ S_3, \mathit{penv}, \mathit{env}) \Downarrow (p_2, \mathit{penv}', \mathit{env}')}$$

$$\frac{(e_1, \mathit{env}) \Downarrow \texttt{false} \qquad (S_3, \mathit{penv}, \mathit{env}) \Downarrow (p_3, \mathit{penv}', \mathit{env}')}{(\texttt{if}\ e_1\ \texttt{then}\ S_2\ \texttt{else}\ S_3, \mathit{penv}, \mathit{env}) \Downarrow (p_3, \mathit{penv}', \mathit{env}')}$$

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

For SetStmt:

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

For ProcedureStmt:

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

For CallStmt:

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

where

$$\begin{aligned} A &= \quad (e_1, \mathit{env}) \Downarrow v_1 \quad ... \quad (e_n, \mathit{env}) \Downarrow v_n \\ \mathit{penv}(f) &= \quad (\texttt{ProcedureStmt}\ \ f\ \ ps\ \ \mathit{body}) \end{aligned}$$

$$\frac{}{(\texttt{call}\ \ f\texttt{(}e_1, ..., e_n\texttt{)}, \mathit{penv}, \mathit{env}) \Downarrow \texttt{ExnVal "Procedure}\ \ f\ \ \texttt{undefined"}}\; f \notin \mathit{penv}$$