Not having a good time with the mathpartir package here.

Exercise 4.1

Calls to `g` declare a reference `counter`, set its contents, then return its contents. The information between calls is lost because `counter` is different on each call. That is, the reference data structure represents a different location. The first program was made such that the environment of the closure had access to `counter`, and thus was able to reference that variable.

Exercise 4.2

$$\frac{\texttt{(value-of}\quad exp_1\quad \rho\quad \sigma_0\texttt{)}\ \texttt{=}\ \texttt{(}val_1\texttt{,}\ \sigma_1\texttt{)}}{\begin{array}{c}\texttt{(value-of (zero?-exp}\quad exp_1\texttt{)}\quad \rho\quad \sigma_0\texttt{)}\ \texttt{=} \\ \texttt{((bool-val \#t),}\ \sigma_1\texttt{)}\quad \text{if}\quad \lfloor val_1 \rfloor = 0 \\ \texttt{((bool-val \#f),}\ \sigma_1\texttt{)}\quad \text{if}\quad \lfloor val_1 \rfloor \neq 0\end{array}}$$

Exercise 4.3

$$\frac{\begin{array}{c}\texttt{(value-of}\quad exp_1\quad \rho\quad \sigma_0\texttt{)}\ \texttt{=}\ \texttt{(}val_1\texttt{,}\ \sigma_1\texttt{)} \\ \texttt{(value-of}\quad exp_2\quad \rho\quad \sigma_1\texttt{)}\ \texttt{=}\ \texttt{(}val_2\texttt{,}\ \sigma_2\texttt{)}\end{array}}{\begin{array}{c}\texttt{(value-of (call-exp}\quad exp_1\quad exp_2\texttt{)}\quad \rho\quad \sigma_0\texttt{)} \\ \texttt{= (apply-procedure (expval->proc}\quad val_1\texttt{)}\quad val_2\quad \sigma_2\texttt{)}\end{array}}$$

$$\frac{val_1 = \texttt{(procedure}\quad var\quad body\quad \rho\texttt{)}}{\begin{array}{c}\texttt{(apply-procedure}\quad val_1\quad val_2\quad \rho_0\texttt{)} \\ \texttt{= (value-of}\quad body\quad [var = val_2]\rho\quad \sigma_0\texttt{)}\end{array}}$$

Exercise 4.4

$$\frac{\begin{array}{c}\texttt{(value-of}\ \ exp_1\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}val_1\texttt{,}\ \sigma_1\texttt{)}\\ \vdots \\ \texttt{(value-of}\ \ exp_n\ \ \rho\ \ \sigma_{n-1}\texttt{)} = \texttt{(}val_n\texttt{,}\ \sigma_n\texttt{)}\end{array}}{\texttt{(value-of (begin}\ \ exp_1 \ldots exp_n\texttt{)}\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}val_n\texttt{,}\ \sigma_n\texttt{)}}$$

Exercise 4.5

$$\frac{\begin{array}{c}\texttt{(value-of}\ \ exp_1\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}val_1\texttt{,}\ \sigma_1\texttt{)}\\ \vdots \\ \texttt{(value-of}\ \ exp_n\ \ \rho\ \ \sigma_{n-1}\texttt{)} = \texttt{(}val_n\texttt{,}\ \sigma_n\texttt{)}\end{array}}{\begin{array}{c}\texttt{(value-of (list-exp}\ \ exp_1 \ldots exp_n\texttt{)}\ \ \rho\ \ \sigma_0\texttt{)}\\ = \texttt{((pair-val}\ \ val_1\ \ \texttt{(... (pair-val}\ \ val_n\ \ \texttt{(emptylist-val)) ...)),}\ \sigma_n\texttt{)}\end{array}}$$

Exercise 4.6

$$\frac{\begin{array}{c}\texttt{(value-of}\ \ exp_1\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}l\texttt{,}\ \sigma_1\texttt{)}\\ \texttt{(value-of}\ \ exp_2\ \ \rho\ \ \sigma_1\texttt{)} = \texttt{(}val\texttt{,}\ \sigma_2\texttt{)}\end{array}}{\texttt{(value-of (setref-exp}\ \ exp_1\ \ exp_2\texttt{)}\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}val\texttt{,}\ [l = val]\sigma_2\texttt{)}}$$

Exercise 4.7

$$\frac{\begin{array}{c}\texttt{(value-of}\ \ exp_1\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}l\texttt{,}\ \sigma_1\texttt{)}\\ \texttt{(value-of}\ \ exp_2\ \ \rho\ \ \sigma_1\texttt{)} = \texttt{(}val\texttt{,}\ \sigma_2\texttt{)}\end{array}}{\texttt{(value-of (setref-exp}\ \ exp_1\ \ exp_2\texttt{)}\ \ \rho\ \ \sigma_0\texttt{)} = \texttt{(}\sigma_0(l)\texttt{,}\ [l = val]\sigma_2\texttt{)}}$$

Exercise 4.8

newref, deref, and setref! take linear time.

Exercise 4.9

Exercise 4.10

```
;; the-grammar
(expression
 ("begin" expression (arbno ";" expression) "end")
 begin-exp)

;; value-of
(begin-exp (exp1 rest-exps)
           (letrec ((begin-inner
                      (lambda (exps final-val)
                        (if (null? exps)
                            final-val
                            (begin-inner (cdr exps) (value-of (car
             (begin-inner rest-exps (value-of exp1 env))))
```

Exercise 4.11

```
;; the-grammar
(expression
 ("list" "(" (separated-list expression ",") ")")
 list-exp)

;; value-of
```