Specifying the Behavior of Expressions

Exercise 3.1

$\lfloor\texttt{(value-of <<x>>}\ \rho\texttt{)}\rfloor = \texttt{10}$

$\lfloor\texttt{(value-of <<3>>}\ \rho\texttt{)}\rfloor = \texttt{3}$

$\lfloor\texttt{(value-of <<v>>}\ \rho\texttt{)}\rfloor = \texttt{5}$

$\lfloor\texttt{(value-of <<i>>}\ \rho\texttt{)}\rfloor = \texttt{1}$

Exercise 3.2

A $val \in ExpVal$ must be that which is in $Int{+}Bool$. Then a $val \in ExpVal$ for which $\lceil\lfloor val\rfloor\rceil \neq val$ is where $val \in Bool$, such as $val = true$.

Exercise 3.3

We are able to describe the arithmetic operations in terms of subtraction. We cannot do so if we chose addition.

Exercise 3.4

Let $\rho = \texttt{[x=}\lceil 33\rceil\texttt{, y=}\lceil 22\rceil\texttt{]}$.

$$\cfrac{\texttt{(value-of-program <<if zero? (-(x,11)) then -(y,2) else -(y,4)>>)}}{\cfrac{\texttt{(value-of <<if zero? (-(x,11)) then -(y,2) else -(y,4)>>}\ \rho\texttt{)}}{\cfrac{\texttt{(value-of <<zero? (-(x, 11))>>}\ \rho\texttt{) = (bool-val \#f)}}{\cfrac{\texttt{(value-of <<-(y,4)>>}\ \rho\texttt{)}}{\lceil 18\rceil}}}}$$

Exercise 3.5

$$\frac{\begin{array}{c}\texttt{(value-of <<let x = 7}\\ \texttt{in let y = 2}\\ \texttt{in let y = let x = -(x,1) in -(x,y)}\\ \texttt{in -(-(x,8),y)>> } \rho_0\texttt{)}\end{array}}{skib}$$

Exercise 3.6

```
(minus-exp (exp1 expression?))

(minus-exp (exp1)
           (value-of (diff-exp (const-exp 0)
                               exp1)
                     env))
```

Exercise 3.7

```
(add-exp
 (exp1 expression?)
 (exp2 expression?))
(mul-exp
 (exp1 expression?)
 (exp2 expression?))
(div-exp
 (exp1 expression?)
 (exp2 expression?))

(add-exp (exp1 exp2)
         (num-val (+ (expval->num (value-of exp1 env))
                     (expval->num (value-of exp2 env)))))
(mul-exp (exp1 exp2)
         (num-val (* (expval->num (value-of exp1 env))
                     (expval->num (value-of exp2 env)))))
(div-exp (exp1 exp2)
         (let ((val2 (expval->num (value-of exp2 env))))
           (if (= 0 val2)
               (report-division-by-zero)
               (num-val (/ (expval->num (value-of exp1 env))
                           val2)))))
```