**Specifying the Behavior of Expressions**

Exercise 3.1

$\lfloor$`(value-of <<x>> ρ)`$\rfloor$ = `10`

$\lfloor$`(value-of <<3>> ρ)`$\rfloor$ = `3`

$\lfloor$`(value-of <<v>> ρ)`$\rfloor$ = `5`

$\lfloor$`(value-of <<i>> ρ)`$\rfloor$ = `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 $\rho$ = `[x=`$\lceil$`33`$\rceil$`, y=`$\lceil$`22`$\rceil$`]`.

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

Exercise 3.5

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

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)))))
```

Exercise 3.8

```
(equal?-exp
 (exp1 expression?)
 (exp2 expression?))
(greater?-exp
 (exp1 expression?)
 (exp2 expression?))
(less?-exp
 (exp1 expression?)
 (exp2 expression?))

(equal?-exp (exp1 exp2)
            (bool-val (= (expval->num (value-of exp1 env))
                         (expval->num (value-of exp2 env)))))
(greater?-exp (exp1 exp2)
              (bool-val (> (expval->num (value-of exp1 env))
                           (expval->num (value-of exp2 env)))))
(less?-exp (exp1 exp2)
           (bool-val (< (expval->num (value-of exp1 env))
                        (expval->num (value-of exp2 env)))))
```

Exercise 3.9

```
(list-val
 (first expval?)
 (next expval?))
```