Programming Languages
CS320 Spring Semester, 2019
= Homework No. 3 =

Due: April 19, 2019 (Friday) 11:59PM (by TA clock)
Submit (1 rkt file) To: 张志忠(Zhizhong Zhang) by email mrz15120@163.com

Implement MUWAE which deals with multiple values. Begin with the "WAE-implementation.rkt" to serve as the basis of your work. As a first step, replace all of the "WAE"s with "MUWAE".

- **Multiple Values**
  Extend the language so that instead of only plain numeric values we actually deal with multiple values. Thus, a single number and a list of numbers are both valid WAE expressions. As a representation for this, we replace the number in the output of eval (and run) with (listof number).

- **Fixing the Arithmetic Operators**
  Next, we need to fix the arithmetic operators. This is a bit tricky, since each of them receives two inputs that are both lists, and they should apply the operator on each pair from these two inputs, and collect a list of all of the results. So to make it easy, here is a skeleton of a utility function that will do this work. It is near-complete, and you have a tiny hole to fill:

```
;bin-op : (number number -> number) (listof number) (listof number) -> (listof number)
;; applies a binary numeric function on all combinations of numbers from
;; the two input lists, and return the list of all of the results
(define (bin-op op ls rs)
  (define (helper l rs)
    ;; f : number -> number
    ...
    (map f rs))
  (if (null? ls)
    null
    (append (helper (first ls) rs) (bin-op op (rest ls) rs))))
```

- **Fixing the with**
  The next that requires fixing is the one used in the evaluation of with. The problem there is that we're still wrapping the numeric result of eval in a num so we can use it with subst, and num expects a single number.
  One way to resolve this would be to add a new variant called nums to our AST definition. But this would require reworking new cases for it in a few places. So instead, we will choose an easier solution: just change the existing num so instead of holding a single number it will hold a (listof number). Once you do that, you will have three easy fixes to do. First, the code that parses numbers should put a list with the number in the num. Next, there are two small fixes left in the eval function, and everything will work fine with it.
  Don't forget to add tests that demonstrate that this works: that using with to bind a name to a multi-valued expression works as expected. (You need to do this test even though you should already have

complete coverage at this point.) Here are some tests that should work once you’re done with this part:

```
(test (run "{+ 3 7}") '(10))
(test (run "{- 10 {3 5}}") '(7 5))
(test (run "{with {x {+ 5 5}} {+ x x}}") '(20))
```

- **Adding More Arithmetic Operators**
  Finally, add two arithmetic operators, muwae-min and muwae-max that take three expressions evaluating to integers and return the minimum and the maximum number, respectively, to MUWAE:

```
(test (run "{muwae-min 3 4 5}") '(3))
(test (run "{muwae-max {+ 1 2} 4 5}") '(5))
(test (run "{+ {muwae-min 9 3 7} {muwae-max 6 2 20}}") '(23))
```

Here are some tests:

```
(test (run "{+ {1 2} {3 4}}") '(4 5 5 6))
(test (run "{- {+ {1 2} {3 4}} {1 2}}") '(3 2 4 3 4 3 5 4))
(test (run "{- {10 2 1} {3 2}}") '(7 8 -1 0 -2 -1))
(test (run "{with {x {1 2}} {+ x {4 3}}}") '(5 4 6 5))
(test (run "{with {x 9} {+ x {with {x 3} x}}}") '(12))
(test (run "{with {x 100} {+ x {with {y 3} x}}}") '(200))
(test (run "{with {x 5} {+ x {with {x 3} 10}}}") '(15))
(test (run "{with {x {7 5}} {+ x x}}") '(14 12 12 10))
(test (run "{with {x {1 2}} {+ x {4 3}}}") '(5 4 6 5))
(test (run "{with {x 2} {- {+ x x} x}}") '(2))
(test (run "{+ {muwae-min 3 5 7} {muwae-min 10 100 1000}}") '(13))
(test (run "{+ {muwae-min 9 3 7} {muwae-max 6 2 20}}") '(23))
(test (run "{with {x 10} {muwae-max x 2 3}}") '(10))
(test (run "{with {x 20}
                  {with {y 5}
                        {with {z {10 20}} {+ z {muwae-max {+ x y} 0 12}}}}}")
           '(35 45))

(test (run "{with {x 20}
                  {with {y 5}
                        {with {z {10 20}} {+ z {muwae-min {+ x y} 0 12}}}}}")
       '(10 20))

(test (run "{with {x {muwae-min 3 9 5}} {with {y {- x 3}} y}}") '(0))
(test (run "{with {x {muwae-max 2 3 5}} {muwae-min x 7 6}}") '(5))
(test (run "{with {x {muwae-max 9 7 10}} {muwae-max 8 x {+ 1 x}}}") '(11))
(test (run "{- {muwae-min 6 4 5} {muwae-max 2 3 4}}") '(0))
(test (run "{with {x {+ 7 2}} {muwae-min x 7 0}}") '(0))
(test (run "{+ {muwae-min 9 3 7} {muwae-max 6 2 20}}") '(23))
(test (run "{with {x {13}} {muwae-min x 1 12}}") '(1))
(test (run "{with {x {muwae-min 2 1 3}} {+ x x}}") '(2))
(test (run "{with {a 10} {with {b 19} {with {c 2} {muwae-min a b c}}}}")
       '(2))


(test (run "{with {x 3} {muwae-max 3 4 {+ x x}}}") '(6))
```

```
(test (run "{with {a 10} {with {b 19} {with {c 2} {muwae-max a b c}}}}")
       '(19))
(test (run "{with {x {muwae-min 2 5 4}} {+ x x}}") '(4))
(test (run "{with {x {muwae-max 2 5 4}} {+ x x}}") '(10))
(test (run "{with {x {- 11 3}} {muwae-max x {+ x x} {- x x}}}") '(16))
(test (run "{with {x {- 11 3}} {muwae-min x {+ x x} {- x x}}}") '(0))
(test (run "{muwae-min {+ 4 4} {with {x 5} {+ x {with {x 3} 10}}} 3}") '(3))
(test (run "{muwae-max {+ 4 4} {with {x 5} {+ x {with {x 3} 10}}} 3}") '(15))
(test (run "{with {x {13}} {muwae-min x 1 12}}") '(1))
(test (run "{with {x {10} } {muwae-max x 2 3}}") '(10))
(test (run "{with {x {muwae-min 2 1 3}} {+ x x}}") '(2))
(test (run "{with {x {muwae-max 2 1 3}} {+ x x}}") '(6))
(test (run "{with {x 2} {muwae-min x 3 10}}") '(2))
(test (run "{with {x 2} {muwae-max x 3 10}}") '(10))
(test (run "{muwae-min {+ 4 4} 2 3} ") '(2))
(test (run "{muwae-max {+ 4 4} 2 3} ") '(8))
(test (run "{with {x 10} {muwae-min x 2 3}}") '(2))
(test (run "{with {x 10} {muwae-max x 2 3}}") '(10))
(test (run "{with {x {10}} {muwae-max x 2 3}}") '(10))
(test (run "{muwae-min {+ 3 4} 5 6}") '(5))
(test (run "{muwae-max {+ 3 4} 5 6}") '(7))
(test (run "{with {x {10}} {muwae-min x {3} {5}}}") '(3))
(test (run "{with {x {10}} {muwae-max x {3} {5}}}") '(10))
(test (run "{muwae-min {3} 4 5}") '(3))
(test (run "{muwae-max {3} 4 {5}}") '(5))
(test (run "{+ {10 100 1000 10000} {muwae-min {- 3 4} 5 6}}")
       '(9 99 999 9999))
```