Homework: FuncLang (Part II)

Instructions:

- Early deadline: Oct 4 (Wed) 2017 at 11:00 PM; Regular deadline: Oct 6 (Fri) 2017 at 11:00 PM
- Download and install a *fresh* copy of the Funclang interpreter. Use this fresh copy of Funclang interpreter for all questions in HW5.
- For Question 1, write your answer in HW5.pdf and save it at pldis-code/src/funclang/examples/HW5.pdf
- Extend the Funclang interpreter for Question 2 to Question 4.
- Export your homework solution as archive file (File->Export->General->archive). Attach the archive file. Please provide the complete project as submission not just the src directory.

Questions:

1. (5 points) Write a curried form of a function using lambda expression that takes length, width, and height of a cuboid, and computes the volume of cuboid by multiplying length, width and height. Use that function, to compute volume of a cuboid with length equal to 3, width equal to 4 and height equal to 2.

2. (20 points) Current implementation of Funclang allows equality check only for the numeric values. Comparing two string values, Boolean values or list values results in DynamicError. For example, following transcript shows the result of comparing two Strings

   $ (= "name" "name")
   funclang.Value$StringVal cannot be cast to funclang.Value$NumVal

   Extend the Funclang language and its implementation such that the equality operator works for each kind of value supported by the language. For example,

   $ (= "abc" "abc")
   #t
   $ (= "abc" "abcdef")
   #f

   $ (= #t #t)
   #t
   $ (= #t #f)
   #f

   Two list values are considered equal if they have the same size and each element of the list is equal to corresponding element in the other list.

   $ (= (list) (list))
   #t

```
$ (= (list 1 2 3 4) (list 1 2 3 4))
#t
$ (= (list 1 2 3 4) (list 1 2 3 4 5))
#f
$ (= (list 1 2 3 4 5) (list 1 2 3 4))
#f
$ (= (list 1 2 3 4 (list)) (list 1 2 3 4 (list)))
#t
$ (= (car (list 1 2 3)) 1)
#t
$ (= (car (list 1 2 3)) 2)
#f
$ (= (cdr (list 1 2 3)) 2)
#f
$ (= (cdr (list 1 2 3)) (list 2 3))
#t
$ (= (cdr (list 1 2 3)) (cdr (list 4 2 3)))
#t
$ (= (cons 0 (list 1 2)) (list 0 (list 1 2)))
#f
$ (= (cons 0 (list 1 2)) (list 0 1 2))
#t
```

Two function values are only considered equal, if they point to the same FunVal object.

```
$ (define test (lambda (x) x))
$ (define test1 (lambda (y) y))
$ (= test test1)
#f
$ (= test test)
#t
```

3. (25 points) This question's goal is to familiarize you with inbuilt function on different data structures. Extend the functionality of the Funclang interpreter to add support for following eight predicates: number?, boolean?, string?, procedure?, pair?, list?, null?, unit? A predicate is a function from Funclang value to boolean, i.e. #t or #f.

   The predicate number? evaluates to #t if its input is a number. Similarly, boolean?, string? procedure? pair?, list?, null?, and unit? evaluate to #t if their inputs are boolean, string, procedure, pair, list, null, and unit respectively. Following transcript illustrates some of these predicates.

```
> (number? 342)
#t
> (boolean? (> 300 42))
#t
> (string? "342")
#t
```

```
> (procedure? (lambda (x) x))
#t
> (list? (list 3 4 2))
#t
> (list? (cons 1 2))
#f
> (pair? (cons 1 2))
#t
> (list? (cons 1 (list)))
#t
> (list? (list ))
#t
> (null? (list ))
#t
```

Funclang doesn't currently support expressions that produce unit values, so you wouldn't be able to test unit?, but you should implement it and we will verify your code directly to see if it behaves as intended.

4. (20 points) Rajan's book Exercise 5.9.5 in Ch 5