Computer Science & Engineering Department
I. I. T. Kharagpur

Principles of Programming Languages: CS40032

Elective

Assignment – 1: λ-Calculus *Marks: 25*

Assign Date: *19th February, 2022* Submit Date: *23:55, 28th February, 2022*

Instructions: Please solve the questions using pen and paper and scan the images. Every image should contain your roll number and name.

1. Mark the free variables in the following lambda expressions **[1+2+2 = 5]**
   (a) $\lambda x.\ x\ z\ \lambda y.\ x\ y$
   (b) $(\lambda x.\ x\ z)\ \lambda y.\ w\ \lambda w\ w\ y\ z\ x$
   (c) $\lambda x.\ x\ y\ \lambda x.\ y\ x$

2. Proof the following using encoding in λ-calculus. **[2 * 5 = 10]**
   (a) $NOT(NOT\ FALSE)\ =\ FALSE$
   Given:
   $$NOT\ =\ \lambda x.\ ((x\ TRUE)\ FALSE)$$
   $$TRUE\ =\ \lambda x.\ \lambda y.\ x$$
   $$FALSE\ =\ \lambda x.\ \lambda y.\ y$$
   (b) $OR\ FALSE\ TRUE\ =\ TRUE$
   Given:
   $$OR\ =\ \lambda x.\ \lambda y.\ ((x\ TRUE)\ y)$$
   $$TRUE\ =\ \lambda x.\ \lambda y.\ x$$
   $$FALSE\ =\ \lambda x.\ \lambda y.\ y$$
   (c) Solve: $add\ \overline{5}\ \overline{1}$
   Given: $add\ =\ \lambda n.\lambda m.\lambda f.\lambda x.\ n\ f\ (m\ f\ x)$
   (d) $IF\ TRUE\ THEN\ x\ ELSE\ y\ =\ y$
   Given:
   $$IF\ a\ THEN\ b\ ELSE\ c\ =\ a\ b\ c$$
   $$TRUE\ =\ \lambda x.\ \lambda y.\ x$$
   $$FALSE\ =\ \lambda x.\ \lambda y.\ y$$
   (e) *add* and *mul* are commutative.
   Given:
   $$mul\ =\ \lambda n.\lambda m.\lambda x.\ (n\ (m\ x))$$
   $$add\ =\ \lambda n.\lambda m.\lambda f.\lambda x.\ n\ f\ (m\ f\ x)$$

3. Answer the following questions of Functional Programming language. **[2 * 5 = 10]**
   (a) Write the corresponding eager evaluation version of the $$ operator of Haskell. The definition of the $$ operator is given below.

```
$$:: Bool -> Bool -> Bool
True  && x = x
False && _ = False
```

   Name the eager evaluation version as $$!
   (b) Write a function to rotate a list N places to left in Lisp.
   (c) Write a function in Scheme to define Ackermann function.
   (d) Write a function in Haskell to find the sum of all the odd value squares smaller than the value 10,000.
   (e) Define a function in Haskell to return the smallest divisor of a number using the 'otherwise' guard in haskell.
   (Otherwise is a predicate in Haskell)