Homework: Lambda Calculus

Learning Objectives:

1. Understand lambda calculus, theory of functional programming
2. Understand β-reduction, church encoding

Instructions:

- Total points: 53 pt
- Early deadline: Oct 24 (Wed) 2018 at 6:00 PM; Regular deadline: Oct 26 (Fri) 2018 at 6:00 PM (or till TAs start grading the homework)
- Submit one pdf file to Canvas under Assignments, Homework 6. You are encouraged to use latex. But we will accept a scanned copy as well.

Questions:

1. (9 pt) Perform β-reduction for the following λ expressions.

   (a) (3 pt) $((\lambda(x)\, x)((\lambda(y)\, y)(((\lambda(v)(\lambda(w)\, w))\, a)\, b)))$

   (b) (3 pt) $(((\lambda(x)(\lambda(y)(x\, y)))((\lambda(w)\, w)\, a))\, b)$

   (c) (3 pt) $(((\lambda(x)(\lambda(y)(y\, y)))(\lambda(a)\, a))\, b)$

2. (6 pt) The goal of this problem is to help you understand the evaluation order of lambda calculus. In the following, show the steps of β-reduction for the lambda expression using two types of evaluation orders

   $$((\lambda(x)\, p)((\lambda(y)(y\, y))(\lambda(z)(z\, z))))$$

3. (3 pt) Define the logic Boolean operations of *or a b* using *true*, *false* and *ite* given in the lecture.

4. (20 pt) Using the Church numeral encoding and also *succ*, *true*, *false* provided in the lecture, answer the following two questions:

   (a) (5 pt) What is the result of $((\lambda(z)((n\, f)\, z))\, three)$?

   (b) Suppose we define *third*: $(\lambda(x)(\lambda(y)(\lambda(z)\, z)))$ and *g*: $(\lambda(n)((n\, third)\, true))$, what is the result of:

      i. (4 pt) $(g\, zero)$
      ii. (3 pt) $(g\, one)$
      iii. (3 pt) $(g\, two)$
      iv. (5 pt) What mathematical/logical operation is computed by g?

5. (15 pt) Given:
   true: $(\lambda(x)(\lambda(y)\, x))$
   false: $(\lambda(x)(\lambda(y)\, y))$
   g: $(\lambda(n)((n(\lambda(x)\, false))\, true))$
   zero: $(\lambda(f)(\lambda(x)\, x))$
   one: $(\lambda(f)(\lambda(x)(f\, x)))$.

   (a) (3 pt) What is the result of $(g\, zero)$?

   (b) (3 pt) What is the result of $(g\, one)$?

   (c) (3 pt) What computation does g performs?

   (d) (6 pt) Suppose we define *ite*: $(\lambda(c)(\lambda(t)(\lambda(e)((c\, t)\, e))))$ to represent if then else ((if c t) e). Write a lambda calculus expression that uses g and *ite* to define *IsEqual* that tests if two numbers m and n have the equal values. (Hint: You can directly use the sub(subtraction) function created in question 3, and (sub m n) equals zero if m<n)