G6021: Comparative Programming

Exam Practice

1. Consider the λ-term $t = (\lambda x.xx)(II)$, where $I = \lambda x.x$.

   (a) Give the definition of a β-reduction graph.

   (b) Draw the β-reduction graph of the term t.

   (c) Which reduction strategy gives the shortest reduction sequence for reducing t to normal form?

2. Give the types of the following:

   (a) `f(x) = if x>0 then True else False`

   (b) `f(g,x) = g(g x)`

   (c) `inl = \f->f+1`

   (d) `\(x,y,z) -> x (y z)`

3. Using list comprehension, write a Haskell expression that will generate Pythagorean triples (triple of numbers that satisfy Pythagoras' theorem: $a^2 + b^2 = c^2$) for numbers up to 100.

4. Using `map` and a one-off function (written using Haskell's lambda notation), write a function that will swap all pairs of a list of pairs of numbers. I.e. write a function `f` such that `f [(1,2),(3,4)] = [(2,1),(4,3)]`.

5. Convert the following function to accumulating parameter style. Include in your answer the type of the converted function.

   ```
   fact (n) = if n<1 then 1 else n*(fact (n-1))
   ```

6. Convert the following function to accumulating parameter style. Include in your answer the type of the converted function.

   ```
   power(x,y) = if y==0 then 1 else x*power(x,y-1)
   ```

7. * Define `add` in PCF (a function that takes two arguments, and computes the sum of the arguments).