CPSC 417: midterm exam

Robin Cockett

November 5, 2007

This exam is worth 20% of the course each question is worth 4 points:

1. (Describe the translation from list comprehension syntax to core Haskell code. Translate the following function:

   ```
   pairs:: [a] -> [b] -> [(a,b)]
   pairs xs ys = [(x,y)| x <- xs, y <- ys, not x==y]
   ```

   Rewrite this function using the "do" syntax.

2. Given the following representation of λ-terms:

   ```
   data  LTree = Var Int
               | Abs Int LTree
               | App LTree LTree
   ```

   (a) Write the Haskell code to determine the list of free variables of a λ-term so represented.

   ```
   freeVars:: LTree -> [Int]
   ```

   (b) Write the Haskell code to perform a substitution of a (free) variable in a λ-term.

   ```
   subst:: LTree -> (Int,LTree) -> LTree
   ```

3. (a) Explain how conditional statements

$$\mathsf{if}\ e\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$$

are programmed in the λ-calculus.

(b) Explain what a fixed point combinator is. Prove that

$$\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

is a fixed point combinator.

(c) Explain how to program the `factorial` function using fixed points (you may assume that you have basic arithmetic functions defined).

4. In the λ-calculus how do you represent the following binary trees:

```
data Tree a b = Leaf a | Node (Tree a b) b (Tree a b)
```

What are the λ-terms for the constructors and the associated map and fold function for this datatype.

(Harder) how do you write the case function for the above datatype?

5. (a) Explain the by-value reduction strategy: what are its shortcomings?

(b) What is the difference between leftmost outermost (normal order reduction) and outermost reduction (by-name).

(c) What is head reduction?

(d) Demonstrate the leftmost outermost and by-value reduction strategies on the following λ-terms:

i. $(\lambda x.x(\lambda nsz.s(nsz))x)(\lambda sz.s(sz))$

ii. $(\lambda xy.x)(\lambda z.z)((\lambda x.xx)(\lambda x.xx))$

iii. $(\lambda xy.yx)(\lambda xy.y)(\lambda x.xx)(\lambda x.xx)$