Lund University
Department of Computer Science

EDAN40: Functional Programming
19th December 2011, 8.00–13.00

Exam

1. What is the type of the expression `map (const (++))`?
   What is the type of the expression `const (map (++))`?

2. Explain the terms *currying* and *uncurrying*. Why would you do either of them?

3. What is the type and value of the expression:

   ```
   do [1, 2, 3, 4]; "curry"
   ```

   What would be the answer in case of

   ```
   do [1, 2, 3, 4]; return "uncurry"
   ```

   Please explain both answers.

4. Explain what the following function does:

   ```
   c a = (a \\) . (a \\)
   ```

   where

   ```
   (\\) = foldl (flip delete)
   ```

   and

   ```
   delete x [] = []
   delete x (y:ys)
     | x == y    = ys
     | otherwise = y:(delete x ys)
   ```

5. Rewrite the following two definitions eliminating argument symbols from the left-hand side (in so called point-free form):

   ```
   f x y = (5 + x) / y
   g x y = x y
   ```

6. Define a type `Tree` where all nodes of a tree, including its leaves, can keep an Integer value; then define a predicate

   ```
   subTree t1 t2
   ```

   returning `True` when tree `t1` is a subtree of tree `t2`.