

## Class Test 1

All the questions in this test deal with the following mystery function *times* written in applied lambda calculus:

$$\text{times} = (\lambda n. \lambda f. \text{if } (n = 0) (\lambda x. x) \\ (\lambda x. \text{times } (n - 1) f (f x)))$$

### 1. Types

Calculate the principal type (most general type) of *times*. [20%]  
(At a minimum, you should guess the correct type of *times*.)

### 2. Lambda calculus comprehension

Write a one-sentence description of what the *times* function is, i.e., what does it return when applied to arguments *n* and *f* in general?

You might want to analyse the function carefully and work through some examples of its use in order to figure this out. (It is not expected to be obvious.) [20%]

### 3. Reduction semantics

Reduce the following term to normal form:

$$\mathbf{let} \text{ inc} = \lambda z. z + 1 \\ \mathbf{in} \text{ times } 3 \text{ inc } x$$

Assume that the **Y** combinator can be reduced using the rule  $\mathbf{Y} M \longrightarrow M (\mathbf{Y} M)$ .

You can use any reduction strategy. You need not show all the steps of the reduction in full glory, but include enough detail to make sure that your reductions are correct. Abbreviate terms as necessary to keep your work compact and manageable. [20%]

### 4. Lambda calculus comprehension

Given below is a function *m* defined using *times*:

$$m = \lambda x. \lambda y. \mathbf{let} \text{ a} = \lambda z. z + y \\ \mathbf{in} \text{ times } x \text{ a } 0$$

What is the value of *m* 3 4? [10%]

Give a one-sentence description of the function. *m* [10%]

### 5. Lambda calculus

Define a function *drop*, which takes as arguments an integer *n* and a list *l*, and returns the list obtained by dropping the first *n* elements of *l*. For example *drop* 2 (**list** 12345) should have the value (**list** 345).

You must define *drop* using the function *times*, without using recursion directly. [20%]