The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2014-2015

G54FOP FOUNDATIONS OF PROGRAMMING

Time allowed 2 hours

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer all 4 questions
Marks available for sections of questions are shown in square brackets.

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn your examination paper over until instructed to do so

Question 1: [25 pts]

UNTYPED λ-CALCULUS

(a) [8 pts]

Give a definition of normal form. Give a short statement of the *confluence* property of λ-terms and explain its consequence on the uniqueness of normal forms.

(b) [8 pts]

For each of the following λ-terms, state if it is normalizable and, in case it is, write its normal form.

1. $(\lambda x.\lambda y.x\ y)\ (\lambda z.z\ z)\ (\lambda u.u\ u)$
2. $(\lambda x.\lambda y.y)\ (\lambda z.z\ z)\ (\lambda u.u\ u)$
3. $(\lambda x.\lambda y.(x\ y)\ x)\ (\lambda u.\lambda v.u)\ (\lambda z.z)$
4. $(\lambda x.\lambda y.(x\ x)\ y)(\lambda x.\lambda y.(x\ x)\ y)(\lambda z.z)$

(c) [9 pts]

One of the principal ideas of the λ-calculus is that *programs and data structures are the same.* Explain what it means and illustrate it by defining the Church numerals and implementing the exponential operation.

Continued on next page

Question 2: [25 pts]

Simply typed λ-calculus

(a) [8 pts]

Give a clear and short explanation of the properties of Progress and Subject Reduction (also called Preservation) of a type system.

(b) [8 pts]

The following are incomplete terms in the simply typed λ-calculus with a base type o. Rewrite the terms, replacing each question mark with a type; also state what type each resulting term has:

1. $\lambda x :?.\ \lambda y :?.\ \lambda z :?.\ z\ (x\ y)\ y$
2. $\lambda x :?.\ \lambda y :?.\ x\ (y\ x)$
3. $\lambda x :?.\ \lambda y :?.\ \lambda z :?.\ x\ y\ z$

(c) [9 pts]

What type do the Church numerals have in the simply typed λ calculus?

Explain why, for certain operations, we need to have types of natural numbers *at different levels*.

Illustrate it by showing how the exponential operation needs to take arguments of different types.

Question 3: [25 pts]

System T and Inductive Types

(a) [8 pts]

Write the rules for the type Nat of natural numbers in system T:

- Introduction rules (constructors);
- Elimination rule (recursor);
- Reduction rules (computation).

(b) [8 pts]

Implement, using the recursor of system T, the function that sums the squares of the natural numbers up to a certain point:

$$\mathsf{sumsq}\ n = 0^2 + 1^2 + 2^2 + 3^2 + \cdots + n^2$$

(You can assume that you already have a term plus for addition and a term square for squaring.)

(c) [9 pts]

Write the rules of introduction, elimination and reduction for the type Tree_A of binary trees with elements of a type A in the leaves. A tree is either a leaf containing an member of A or a node with two children, which are in turn trees.

Continued on next page

Question 4: [25 pts]

System F and Coinductive Types

(a) [8 pts]

Explain the notion of polymorphism and give the rules of introduction, elimination and reduction of the second-order universal quantifier.

(b) [8 pts]

What is the type of natural numbers in system F? Show how the exponential operation can be defined, with both arguments in the same type.

(c) [9 pts]

Give a definition of pointed *stream coalgebra* on a type A and explain how it computes an infinite sequence of elements of A.

Assume that you have two stream coalgebras x and y, computing streams $x_1, x_2, x_3 \ldots$ and $y_1, y_2, y_3 \ldots$. Construct another coalgebra that computes the interleaving of the two streams: $x_1, y_1, x_2, y_2, x_3, y_3 \ldots$.