SPECIMEN EXAM ONLY – NOT QUALITY CONTROLLED

UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INTRODUCTION TO THEORETICAL COMPUTER SCIENCE

Today

Two Hours

INSTRUCTIONS TO CANDIDATES

Answer ALL questions from Section A,
and ONE question from Section B.
Section A carries 30 marks, and Section B carries 20 marks.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: ITO-Will-Determine
External Examiners: ITO-Will-Determine

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

NOTE: this specimen exam has not gone through the usual quality control procedures for exams. It is intended to give a rough guide to the format and style of questions you should expect. Should you find any errors or unclear points in the questions, please contact the course lecturer.

SECTION A

Answer ALL questions in this section.

In the real exam, section A will be of this format: three shorter questions, one on each of the three parts computability, complexity, λ-calculus.

1. (a) Let $Q \subseteq \mathbb{N}$. State (precisely) what it means to say that Q is *decidable*. *[3 marks]*

   (b) Show that if Q_1 and Q_2 are both decidable, then so is $Q_1 \setminus Q_2$. *[3 marks]*

   (c) Show (e.g. by giving a counter-example) that if $Q \subseteq \mathbb{N} \times \mathbb{N}$ is decidable, then the predicate $Q' \subseteq \mathbb{N}$ given by $Q'(x) \Leftrightarrow \exists y.Q(x, y)$ need not be decidable. (You may assume the (un)decidability of any query shown in the course to be (un)decidable.) *[4 marks]*

2. (a) Define the decision problem CLIQUE. *[3 marks]*

   (b) Show that CLIQUE is in the class NP. *[3 marks]*

   (c) Give an outline of the reduction from 3SAT to CLIQUE. *[4 marks]*

3. (a) Using call-by-name, evaluate each of the following untyped λ-terms as far as possible (intermediate steps need not be written down):

   i. $(\lambda x.x)(\lambda y.yy)$

   ii. $(\lambda y.yyy)(\lambda z.z)$

   iii. $(\lambda x.yx)((\lambda w.ww)(\lambda z.z)v)$

   [4 marks]

   (b) Draw a type derivation tree to show the type of the simply typed λ-term

   $$\lambda f{:}\mathsf{nat} \to \mathsf{nat}.\lambda x{:}\mathsf{nat}.fx$$

   [4 marks]

   (c) What is the type of the polymorphic λ-term $\lambda x.\lambda y.xy$? *[2 marks]*

SECTION B

Answer ONE question in this section.

In the real exam, section B will contain only TWO questions, covering an arbitrary two of the three parts of the course. This specimen contains an example question for all three parts.

1. (a) What does it mean for a set $X \subseteq \mathbb{N}$ to be *semi-decidable*? [*3 marks*]

   (b) Let EH be the set of codes of register machines that halt on *some* input. Prove that EH is semi-decidable. If you need to construct a register machine, you may use pseudo-code; a detailed formal encoding is not necessary. [*7 marks*]

   (c) Let H_0 be the set of codes of register machines that halt on input zero. Give a many–one reduction from H to H_0. You may refer to notations used in your answer to the previous part. [*4 marks*]

   (d) Let $BH(n)$ be the set of codes for machines that halt on all inputs less than n. For a given n, is the set $BH(n)$, undecidable, semi-decidable, or co-semi-decidable? Justify your answer. [*6 marks*]

2. (a) Explain what it means for a problem to be NP-complete, including definitions of 'NP' and 'reduction'. [*6 marks*]

   (b) Suppose Q is NP-complete, and there is a polynomial reduction from Q to a problem Q'. Prove that Q' is NP-hard. [*4 marks*]

   (c) An instance of VERTEX-COVER is a graph G and an integer k, and the question is whether there is a set C of vertices of size k such that every edge has at least one end in C.

   By reduction from CLIQUE, or otherwise, show that VERTEX-COVER is NP-hard. [*6 marks*]

   (d) To show that VERTEX-COVER is NP-complete, it remains to show that it is in NP. Outline *two* techniques for showing that a problem is in NP. [*4 marks*]

3. (a) Give the typing rules for the simply typed λ-calculus. *[6 marks]*

   (b) Suppose that we add to the language a type constructor $\mathbf{Tree}(\tau)$, which is intended to be the type of binary trees with nodes of type τ. We will need associated term constructors:

   - leaf_τ to build a leaf node containing just a value of type τ
   - tree_τ to build a tree with a value of type τ at the root, and left and right subtrees.

   What are appropriate types for these constructors? *[6 marks]*

   (c) We can also represent binary trees directly in the pure calculus, similarly to the Church encodings of numbers. Here is a possible representation, omitting types, of the $\mathsf{tree}_{\mathsf{nat}}$ constructor:

$$\mathsf{tree}_{\mathsf{nat}} \stackrel{\mathsf{def}}{=} \lambda x.\lambda f.\lambda g.\lambda t.\lambda l.tx(ftl)(gtl)$$

   i. Give a similar representation for the $\mathsf{leaf}_{\mathsf{nat}}$ constructor *[3 marks]*

   ii. Write down the encoding of the tree

$$\begin{tikzcd} & 3 \arrow[dl] \arrow[dr] & \\ 2 & & 1 \end{tikzcd}$$

   [3 marks]

   iii. What are the types of the variables in your $\mathsf{leaf}_{\mathsf{nat}}$ constructor? *[2 marks]*