Candidate Number

G5029

THE UNIVERSITY OF SUSSEX

BSc FINAL YEAR EXAMINATION
May/June 2019 (A2)

Limits of Computation

Assessment Period: May/June 2019 (A2)

DO NOT TURN OVER UNTIL INSTRUCTED TO BY THE LEAD INVIGILATOR

Candidates should answer TWO questions out of THREE. If all three questions are attempted only the first two answers will be marked.

The time allowed is TWO hours.

Each question is worth 50 marks.

At the end of the examination the question paper and any answer books/answer sheets, used or unused, will be collected from you before you leave the examination room.

1. This question is about `WHILE`, its semantics, and other notions of effective computability.

   (a) What does the judgment $\mathtt{S} \vdash \sigma_1 \to \sigma_2$ mean for a list of `WHILE`-statements `S` and stores σ_1 and σ_2? [4 marks]

   (b) Give the rule that defines the operational semantics of `WHILE`-assignments, i.e. for the judgement $\mathtt{X} := \mathtt{E} \vdash \sigma_1 \to \sigma_2$ define precisely what σ_2 must be in terms of σ_1. [4 marks]

   (c) Regarding the semantics of `WHILE`, what does $\mathcal{E}\,[\![\mathtt{E}]\!]\,\sigma$ denote exactly? [3 marks]

   (d) Give the semantic equation for $\mathcal{E}\,[\![\mathtt{X}]\!]\,\sigma$ where `X` is a variable. [3 marks]

   (e) What is $\mathcal{E}\,[\![\mathtt{cons\ X\ nil}]\!]\,\{\mathtt{X} : \ulcorner 3 \urcorner\}$? As usual, $\ulcorner n \urcorner \in \mathbb{D}$ denotes the encoding of natural number n. You do *not* have to show intermediate results. [2 marks]

   (f) Give an example of a *single* `WHILE`-command C such that

   $$C \vdash \{\mathtt{X} : \ulcorner n \urcorner, \mathtt{Y} : \ulcorner 0 \urcorner\} \to \{\mathtt{X} : \ulcorner 0 \urcorner, \mathtt{Y} : \ulcorner n\ \mathtt{div}\ 2 \urcorner\}$$

   where $\ulcorner n \urcorner \in \mathbb{D}$ denotes the encoding of natural number n and `div` denotes integer division without remainder. [7 marks]

   (g) Assuming that we start counting variables from 0, give the program-as-data representation of the following `WHILE`-program:


```
prog read X {
           Z:= nil;
           X:= cons Z X;
           if X { Y:= cons Y X }
           }
write Y
```

   [11 marks]

   (h) Why is it important to have a computation model that supports programs-as-data? [6 marks]

   (i) List <u>FOUR</u> (different) notions of computation other than `WHILE` with which one can compute the same functions on natural numbers as one can with `WHILE`. [4 marks]

   (j) Assume `L`-data = `S`-data and an `L`-program `p` which has the following property:

   $$[\![\mathtt{p}]\!]^{\mathtt{L}}\,(\mathtt{d}_1, \mathtt{d}_2) = [\![\mathtt{d}_1]\!]^{\mathtt{S}}\,(\mathtt{d}_2)$$

   where $(\mathtt{d}_1, \mathtt{d}_2)$ denotes pairing in `L`.

   i. Describe informally what program `p` does (without restating the equation above in words). [4 marks]

   ii. Besides `L`-data = `S`-data, what other implicit assumption(s) are necessary for `p` to be well defined by the above? [2 marks]

2. This question is about semi-decidability, decidability, the recursion theorem, and reduction.

   (a) Let $A \subseteq \mathbb{D}$. Assume there exists a WHILE-program p such that the following holds:

   $$[\![\mathtt{p}]\!]^{\mathtt{WHILE}}(\mathtt{d}) = \mathtt{true} \quad \text{if, and only if,} \quad \mathtt{d} \in A.$$

   What property does A have according to the above? [3 marks].

   (b) Give an example of a set A that has the property of Question 2(a). [2 marks]

   (c) Answer each question below with '*YES*' or '*NO*'. If your answer is '*YES*' provide ONE example of a problem (i.e. set) that has the given property combination.

      i. Is there a set that is finite and decidable? [2 marks]

      ii. Is there a set that is finite and undecidable? [2 marks]

      iii. Is there a set that is infinite and decidable? [2 marks]

      iv. Is there a set that is infinite and undecidable? [2 marks]

   (d) For the following sets $A \subseteq$ WHILE-data state whether they are WHILE-decidable and explain your answer. In cases where A is decidable this explanation should consist of a description of the decision procedure. Recall that $\ulcorner p \urcorner$ denotes the encoding of WHILE-program p in $\mathbb{D}$.

      i. $A = \{ \ulcorner p \urcorner \mid p$ returns nil if its input encodes a natural number $\}$ [5 marks]

      ii. $A = \{ \ulcorner p \urcorner \mid \ulcorner p \urcorner = c \}$ where $c \in \mathbb{D}$ is fixed. [5 marks]

   (e) This question addresses the importance of *effective problem reduction* in computability theory.

      i. What is the definition of an *effective problem reduction* $A \leq_{\text{rec}} B$ between two problems $A \subseteq X$ and $B \subseteq Y$? [6 marks]

      ii. Consider the following problems: HALT (as given in lectures), HALT2 = $\{ \ulcorner p \urcorner \mid [\![p]\!]^{\mathtt{WHILE}}(\ulcorner p \urcorner) \downarrow \}$, ODD = $\{ \ulcorner n \urcorner \mid n \in \mathbb{N}$ is odd $\}$, EVEN = $\{ \ulcorner n \urcorner \mid n \in \mathbb{N}$ is even $\}$.
      Which of the following statements are correct? No explanation is required.

         A. ODD $\leq_{\text{rec}}$ EVEN

         B. EVEN $\leq_{\text{rec}}$ ODD

         C. EVEN $\leq_{\text{rec}}$ HALT

         D. HALT $\leq_{\text{rec}}$ EVEN

         E. HALT $\leq_{\text{rec}}$ HALT2

         F. HALT2 $\leq_{\text{rec}}$ HALT [6 marks]

      iii. Explain how the concept of reducibility is employed in computability *and* complexity theory. *Hint: what kind of results can one prove with the help of (various forms of) reduction?* [7 marks]

(f) Explain what the Kleene's Recursion Theorem tells us about the semantics of programming languages. [8 marks]

3. This question is about complexity.

   (a) Consider the statement:

   "***P** is the class of feasible problems.*"

   where the term "feasible problems" informally refers to problems that can be decided in an acceptable amount of time.
   Answer the following questions in relation to this statement:

   i. Define **P**. [3 marks]
   ii. Explain why the statement does not (need to) refer to a particular class $\mathbf{P}^{\mathrm{L}}$. [6 marks]
   iii. Give an example of a problem in **P**. [2 marks]
   iv. By what name is this statement known? [2 marks]
   v. Give <u>ONE</u> argument in favour of this statement. [3 marks]
   vi. Give <u>ONE</u> argument against this statement. [3 marks]

   (b) For each of the following statements, state whether they are (known to be) true, (known to be) false, or whether the answer is currently still unknown. Accordingly, write as answer either *true*, *false*, or *unknown*, respectively.

   For the *true* statements, and only for those, explain in one sentence why they are true (sometimes it may be best to cite a theorem or a result shown in the lectures).

   i. There is a b such for all $a \geq 1$ it holds that: $\mathbf{TIME}^{\mathtt{WHILE}}(a \times n) \subsetneq \mathbf{TIME}^{\mathtt{WHILE}}(a \times b \times n)$
   ii. **NP** is closed under complement
   iii. The *Satisfiability Problem* (SAT) is in **P**.
   iv. The *Satisfiability Problem* (SAT) is **NP**-complete.
   v. The Halting Problem is in **NP**.
   vi. The complexity class **P** is robust regarding changes of the underlying computational model.
   vii. The *Prime Factorisation Problem* is in **P**.
   viii. Quantum computers can solve **NP**-complete problems in polynomial time. [16 marks]

   (c) Discuss the approximability (in polynomial time) of the *Travelling Salesman Problem* (TSP). Here TSP is considered in the optimisation problem version rather than as decision problem. Address the general issue of approximability, and how TSP is special in that respect. Consider the situation for specific versions of TSP as well. Define any complexity classes you introduce in your answer. *Marks will be assigned based on the correctness of your answer and the coverage of the various aspects.* [15 marks]