Computation Theory

Exercise Sheet 1

Alistair O'Brien

February 2, 2024

Before Attempting the Problems

The primary goal of this course is to introduce to you *formal notion of computability* and various computational models, such as *register machines*, *Turing machines*, and the *lambda calculus*. It is vitial that you master many of these topics for various other Tripos courses[1].

Please complete these exercises and submit your solutions **96 hours** before our scheduled supervisions to `ajo41@cam.ac.uk` as a **PDF** attachment. Feel free to choose the format of your answers (handwritten, typed, LaTeX'd) – just ensure that they're (mostly) legible[2].

Attempt to complete at least 75% of the set exercises – if you get stuck simply make a note of it in your answer and move onto the next question. We will be able to discuss all the solutions during the supervisions.

These supervisions will focus on both examinable and non-examinable material, the latter being for pedigogical reasons. Each week I will provide various *practical programming exercises* related to the course content. You are welcome to skip these questions, they are simply there for your benefit.

Some of the exercises are taken from Prof. Dawar's exercise sheets which you can find here: https://www.cl.cam.ac.uk/teaching/2324/CompTheory/exercise-sheet.pdf.

[1] In particular, Complexity Theory (IB), Semantics (IB), Denotational Semantics (II), Types (II), Hoare Logic and Model Checking (II), and Category Theory (II)

[2] Don't worry too much about this though, I can't handwrite anything legible either

Exercises

1. Show that the following arithmetic functions are all register machine computable.
   (a) First projection function $p : \mathbb{N}^2 \to \mathbb{N}$, where $p(x, y) \triangleq x$.
   (b) Constant function with value $n \in \mathbb{N}$, $c : \mathbb{N} \to \mathbb{N}$, where $c(x) \triangleq n$.
   (c) Truncated subtraction function, $\dot{-} : \mathbb{N}^2 \to \mathbb{N}$, defined by $x \dot{-} y \triangleq \begin{cases} x - y & \text{if } y \leq x \\ 0 & \text{otherwise} \end{cases}$
   (d) Integer division function $div : \mathbb{N}^2 \to \mathbb{N}$, defined as
   $$x \; div \; y \triangleq \begin{cases} q, & \text{if } y > 0 \\ 0, & \text{if } y = 0 \end{cases}$$
   where $q, r \in \mathbb{N}$ such that $r < y$ and $x = q \times y + r$.
   (e) Integer remainder function $mod : \mathbb{N}^2 \to \mathbb{N}$, where $x \; mod \; y \triangleq x \dot{-} y \times (x \; div \; y)$.
   (f) Exponential base 2, $e : \mathbb{N} \to \mathbb{N}$, where $e(x) \triangleq 2^x$.
   (g) Logarithm base 2, $\log_2 : \mathbb{N} \to \mathbb{N}$, defined as
   $$\log_2(x) \triangleq \begin{cases} y, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \end{cases}$$
   where $y \in \mathbb{N}$ is the greatest y such that $2^y \leq x$.

2. (*optional*) Prove or disprove the following statements:
   (a) The set of natural numbers is countable.
   (b) The set of integers is countable.
   (c) The Cartesian product of two countable sets is countable.
   (d) The set of rational numbers is countable.
   (e) The finite n-ary product of countable sets is countable.
   (f) The set of polynomials with coefficients from a countable set is countable.
   (g) The powerset of a countable set is countable.
   (h) The set of real numbers is countable.

3. Rephrase the proof of the undecidability of the Halting Problem (with an abstract definition of an algorithm) as a *diagonal argument*.

4. Let $\phi_e : \mathbb{N} \rightharpoonup \mathbb{N}$ denote the unary partial function computed by the register machine with code e. Show that for any given register machine computable unary partial function $f : \mathbb{N} \rightharpoonup \mathbb{N}$, there are infinitely many numbers e such that $\phi_e = f$.

5. Gödel numbering is a general technique for assigning a natural number to some mathematical object (such as a well-formed sentance w in some formal language $L \subseteq \Sigma^*$). The numbering is often computed by translating every symbol $a \in \Sigma$ of the sentance w to a natural number $|a|$, then combining the codes to create a unique Gödel number $G(w) \triangleq 2^{|a_0|} \cdot 3^{|a_1|} \cdot \ldots \cdot p^{|a_k|}$.

   For example, in the formal language of *first order logic* with the mapping:

$$\begin{aligned} \forall &\mapsto 1 \\ . &\mapsto 2 \\ x &\mapsto 3 \\ = &\mapsto 4 \end{aligned}$$

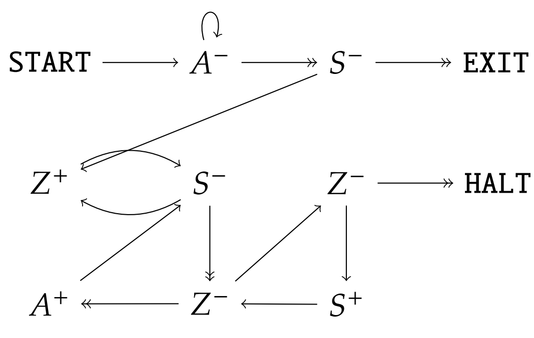
   The formula $\psi \triangleq \forall x.x = x$ has the Gödel number $G(\psi) = 14894594564850$.

   (a) Is the Gödel numbering of register machines described in the notes a bijection, an injection, a surjection, a total function, a partial function, or a relation? Justify your answer.

   (b) In the example of first-order logic above, is a particular Gödel numbering a bijection, an injection, a surjection, a total function, a partial function, or a relation? Justify your answer.

   (c) Describe another encoding to produce a unique number for a sequence of codes $c_0, c_1, \ldots, c_k$.

6. Consider the list of register machine instructions whose graphical representation is shown below. Assuming that register Z holds 0 initially, describe what happens when the code is executed (both in terms of the effect on registers A and S and whether the code halts by jumping to the label `EXIT` or `HALT`).



7. Show that there is a register machine computable partial function $f : \mathbb{N} \rightharpoonup \mathbb{N}$ such that both $\{x \in \mathbb{N} : f(x) \downarrow\}$ and $\{y \in \mathbb{N} : \exists x \in \mathbb{N}. f(x) = y\}$ are register machine undecidable.

8. (*optional*) Write a register machine interpreter in OCaml.

   (a) Define a set of combinators for constructing register machines (e.g. assignment, loops, if statements, etc).

   (b) Implement the universal register machine U.