PART II AUTOMATA AND FORMAL LANGUAGES
MICHAELMAS 2020-21
EXAMPLE SHEET 1

$*$ denotes a harder problem. By convention, we take $\mathbb{N} := \{0, 1, 2, \ldots\}$.

(1) Give an example of a register machine, either via a program diagram or a sequence of instructions, for computing each of the following functions.

(a) $f(m,n) = \begin{cases} 1 \text{ if } m = n \\ 0 \text{ if } m \neq n \end{cases}$

(b) $f(n) = 3n$

(c) $f(m,n) = mn$

(d) $f(m,n) = m \mod (n+1)$

(2) Draw a program diagram for each of the following sequences of instructions, and identify the upper register index of each program. Also, for the specified n, write down the function on n variables that the program computes.

(a) $(1,+,2),(1,+,0)$. $n = 1$.

(b) $(1,-,2,5),(2,+,3),(3,+,4),(4,+,1),(3,-,6,0),(2,-,7,8),(1,+,6),(4,-,9,11),$ $(5,+,10),(2,+,8),(5,-,12,5),(4,+,11)$. $n = 1$.

(c) $(2,+,2),(4,+,3),(3,-,5,7),(1,-,8,6),(5,+,6),(8,+,3),(3,+,0),$ $(1,+,8)$. $n = 4$.

(3) Build up each of the following total recursive functions from the basic functions via composition, recursion and minimisation.

(a) The "predecessor function" $p(n) = n - 1$ and $p(0) = 0$.

(b) $f(a,b,c,x) = ax^2 + bx + c$

(c) $f(m,n) = m^n$

(d*) $f(m,n) = m \mod (n+1)$

(4) Show that, for each $k > 1$, each of the following functions is primitive recursive

(a) $\text{floor}_k(n) = \lfloor \frac{n}{k} \rfloor$

(b) $\text{divide}_k(n) = \begin{cases} \frac{n}{k} & \text{if } n \equiv 0 \mod k \\ 0 & \text{otherwise} \end{cases}$

(c) $\text{power}_k(n,m) = \begin{cases} \frac{n}{k^m} & \text{if } n \equiv 0 \mod k^m \\ 0 & \text{otherwise} \end{cases}$

Date: October 13, 2020.

For the remainder of this sheet, in order to prove that there is an algorithm or that a function is partial computable, you do not need to define a register machine or build up the function from basic functions. It is sufficient to have a mathematical argument in the usual style of a mathematical proof.

(5) Show there is an algorithm that, on input of a code n for a register machine program P_n, halts iff P_n halts on *some* input in *some* number of variables.

(6) Let E be an infinite subset of $\mathbb{N}$. Show that E is recursive iff there is a strictly increasing total recursive function $f : \mathbb{N} \to \mathbb{N}$ whose image is precisely E.

(7) (a) Show that the set of prime numbers is recursive.
(b) Show there is an algorithm that, on input of an integer $n > 1$, outputs the largest prime p which divides n.
(c) How do you thus respond to a critic who says "given an integer n which is the product of two primes, first it is claimed that the difficulty of finding those primes can be used as a basis for secure encryption, but now we're saying that a straightforward register machine can find these primes".

(8) Let $f : \mathbb{N} \to \mathbb{N}$ be a total bijective function. Show that f is total recursive iff f^{-1} is total recursive.

(9) By using (s-m-n) or otherwise, show there is a total computable function $h : \mathbb{N} \to \mathbb{N}$ such that, for each m which defines a register machine code, the partial computable function $f_{h(m),1}$ satisfies:

$$f_{h(m),1}(x) = f_{m,1}(x) + 1 \quad \forall x \in \mathbb{N}$$

(10*) Show there is an r.e. set E such that for every $n \in E$, $f_{n,1}$ is *primitive* recursive, and moreover every primitive recursive function on 1 variable occurs as $f_{n,1}$ for *some* $n \in E$.

(11) (*Not so hard but rather long*)
We define *Cantor's pairing function* $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ by

$$\langle x, y \rangle := \frac{1}{2}(x + y)(x + y + 1) + y$$

and its natural extension, $\langle \cdot, \dots, \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$, inductively via

$$\langle x_1, \dots, x_k \rangle_k := \big\langle \langle x_1, \dots, x_{k-1} \rangle_{k-1}, x_k \big\rangle$$

(a) Show that $\langle \cdot, \cdot \rangle$ is a total bijection from $\mathbb{N}^2 \to \mathbb{N}$.

(b) Show that $\langle \cdot, \cdot \rangle$ is a total computable function.

(c) Show that $\langle \cdot, \dots, \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$ is a total computable bijection.

(d) Show that there is a total computable function $h : \mathbb{N}^2 \to \mathbb{N}$ such that, for each m, k, the partial computable function $f_{m,k}$ satisfies:

$$f_{m,k}(x_1, \dots, x_k) = f_{h(m,k),1}(\langle x_1, \dots, x_k \rangle_k) \quad \forall (x_1, \dots, x_k) \in \mathbb{N}^k$$

(e) Show that with $\langle \cdot, \dots, \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$ we can produce all multi-variable partial computable functions from just the one-variable ones.