

**PART II AUTOMATA AND FORMAL LANGUAGES**  
**MICHAELMAS 2017-18**  
**EXAMPLE SHEET 1**

\* denotes a harder problem.

- (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)  $f(n) = n + 3$

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

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

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

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

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

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

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

(b)  $f(n) = n!$

(c\*)  $f(m, n) = m \bmod (n + 1)$

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

(a)  $\text{rem}_k(n) = n \bmod k$

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

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

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

For the remainder of this sheet, the phrase “*use Church’s thesis*” should be taken to mean “*an informal proof using the conventions of Church’s thesis is sufficient, provided it is clearly stated and justified*”.

- (5) Use Church’s thesis to 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  $\mathbb{N}$ . Use Church’s thesis to show that  $E$  is recursive iff there is a strictly increasing total recursive function  $f : \mathbb{N} \rightarrow \mathbb{N}$  whose image is precisely  $E$ .
- (7) (a) Use Church’s thesis to show that the set of prime numbers is recursive.  
 (b\*) Construct an explicit register machine which computes the characteristic function of the set of prime numbers.
- (8) Use Church’s thesis to show there is an algorithm that, on input of an integer  $n > 1$ , outputs the largest prime  $p$  which divides  $n$ .
- (9) Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  be a total bijective function. Use Church’s thesis to show that  $f$  is total recursive iff  $f^{-1}$  is total recursive.
- (10) Use Church’s thesis to show there is a total computable function  $h : \mathbb{N} \rightarrow \mathbb{N}$  such that, for each  $m$  which defines a register machine code, the partial computable function  $f_{h(m),1}$  satisfies:

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

- (11\*) Use Church’s thesis to 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*  $n \in E$ .
- (12\*) We define *Cantor’s pairing function*  $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \rightarrow \mathbb{N}$  by

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

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

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

- (a) Show that  $\langle \cdot, \cdot \rangle$  is a total bijection from  $\mathbb{N}^2 \rightarrow \mathbb{N}$ .
- (b) Show that  $\langle \cdot, \cdot \rangle$  is a total computable function.
- (c) Show that  $\langle \cdot, \dots, \cdot \rangle_k : \mathbb{N}^k \rightarrow \mathbb{N}$  is a total computable bijection.
- (d) Use Church’s thesis to show that there is a total computable function  $h : \mathbb{N}^2 \rightarrow \mathbb{N}$  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  $\langle \cdot, \dots, \cdot \rangle_k : \mathbb{N}^k \rightarrow \mathbb{N}$  we can produce all multi-variable partial computable functions from just the one-variable ones.