PART II AUTOMATA AND FORMAL LANGUAGES
MICHAELMAS 2025-26
EXAMPLE SHEET 3

(1) Let G be the CFG given by

$$S \to ABS \mid AB, \quad A \to aA \mid a, \quad B \to bA$$

For each of the words $aabaab, aaaaba, aabbaa, abaaba$, determine whether or not they lie in $\mathcal{L}(G)$. If so, give a derivation and a parse tree; if not, explain why not.

(2) (a) Show that the following two languages are both CFLs: $L_1 := \{a^n b^n c^i \mid n, i \geq 1\}$, and $L_2 := \{a^i b^n c^n \mid n, i \geq 1\}$.

(b) Show that the language $L := \{a^n b^n c^n \mid n \geq 1\}$ is not a CFL.

(c) Show that $L_1 \cap L_2 = L$, and hence that the intersection of two CFLs is not necessarily a CFL.

(d) Conclude that the complement of a CFL need not be a CFL.

(3) For each of the following languages, either show that it is a CFL by constructing a CFG for it, or use the pumping lemma to show that it is not a CFL:

(a) $\{a^n b^m \mid n \neq m\}$
(b) $\{a^m b^n c^m d^n \mid m, n \geq 1\}$
(c) $\{a^n b^m c^k d^l \mid 2n = 3m \text{ and } 5k = 7l\}$
(d) $\{a^n b^m c^k d^l \mid 2n = 3k \text{ and } 5m = 7l\}$
(e) $\{a^n b^m c^k d^l \mid 2n = 3k \text{ or } 5m = 7l\}$
(f) $\{ww \mid w \in \{a, b\}^*\}$
(g*) $\{a, b\}^* \setminus \{ww \mid w \in \{a, b\}^*\}$

(4) Give an example of a register machine, via a program diagram and 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) $f(m, n) = m \mod (n + 1)$
(e) $f(m, n) = 1$ if $m = n$ and 0 if $m \neq n$.

(5) 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) $(1, +, 2), (1, +, 0)$. $n = 1$.

Jack Button, jb128@cam.ac.uk, November 10, 2025.

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

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

(a) The "predecessor function" $\pi(n) = n - 1$ for $n \geq 1$ and $\pi(0) = 0$.
(b) $f(a, b, c, x) = ax^2 + bx + c$
(c) $f(n) = m^n$

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

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

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

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

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

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.

(8) 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.

(9) 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.

(10) (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, mathematicians claim that the difficulty of finding those primes can be used as a basis for secure encryption, but a straightforward register machine can find these primes"?

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

(12) Show that 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* $n \in E$.