Part II

Automata and Formal Languages

Year

Paper 1, Section I

4I Automata & Formal Languages

(a) Define what it means for a grammar to be in *Chomsky normal form*.

(b) Suppose G is a grammar in Chomsky normal form. If $w \in \mathcal{L}(G)$ has $|w| = n$, what is the length of a G-derivation of w? [No justification is required.]

(c) Let $\Sigma = \{a, b\}$, $V = \{S, A, B, C\}$. Consider the grammar $G = (\Sigma, V, P, S)$ in Chomsky normal form given by $P = \{S \to AC, C \to BA, A \to AB, B \to BA, A \to a, B \to b\}$. Show that the word *abbabba* is in $\mathcal{L}(G)$ by providing a G-parse tree for it.

A grammar G is said to be in *weak Chomsky normal form* if all production rules are either of the form $A \to a$, $A \to BC$, or $A \to BCD$, for variables A, B, C, D and letters a.

(d) Give a grammar G' in weak Chomsky normal form that is

(i) equivalent to the grammar G from part (c) and

(ii) there is a G'-derivation for the word *abbabba* of length strictly shorter than the number given in part (b).

Justify your answer.

Paper 2, Section I

4I Automata & Formal Languages

Let Σ be an alphabet and $\mathbb{W} := \Sigma^*$ be the set of words over Σ.

(a) Define what it means for $A \subseteq \mathbb{W}$ to be *computably enumerable*.

[You do not need to define what it means for a partial function to be computable.]

(b) Prove that for $\varnothing \neq A \subseteq \mathbb{W}$ the following statements are equivalent:

(i) the set A is computably enumerable;

(ii) the set A is the domain of a partial computable function;

(iii) the set A is the range of a partial computable function;

(iv) the set A is the range of a total computable function.

[*You may assume that the truncated computation function is computable, and that the map $w \mapsto ((w)_0, (w)_1)$ is a bijection from $\mathbb{W}$ to $\mathbb{W}^2$ that can be performed by a register machine.*]

Paper 3, Section I
4I Automata & Formal Languages

(a) Let $G = (\Sigma, V, P, S)$ be a formal grammar and let $\Omega = \Sigma \cup V$. Define $\mathcal{L}(G)$.

[You do not need to define the binary relation $\xrightarrow{G}$ on Ω^*.]

(b) Define what it means for two grammars to be *equivalent.*

(c) Define what it means for two grammars to be *isomorphic.*

(d) Fix $\Sigma = \{a, b, c\}$ and consider the following pairs of grammars with start symbol S and given by their respective sets of productions P_0 and P_1; for each pair, determine whether they are equivalent or non-equivalent. Justify your answers.

(i) $P_0 = \{S \to Aa,\ S \to Sb,\ A \to Ab,\ A \to a,\ B \to Aa,\ B \to b\}$,
$P_1 = \{S \to Sb,\ C \to Da,\ C \to b,\ D \to Db,\ D \to a,\ S \to Da\}$.

(ii) $P_0 = \{S \to AB,\ A \to Aa,\ A \to a,\ B \to Bb,\ B \to b,\ AB \to c\}$,
$P_1 = \{S \to XabY,\ X \to Xa,\ X \to a,\ Y \to Yb,\ Y \to b,\ XY \to c\}$.

(iii) $P_0 = \{S \to aAa,\ A \to bAb,\ A \to b\}$,
$P_1 = \{S \to aYa, Y \to ZZ, Z \to aZa, Z \to bZY, Z \to YZ, Y \to bYb, Y \to b\}$.

[You may assume that isomorphic grammars are equivalent.]

Paper 4, Section I
4I Automata & Formal Languages

(i) Define what it means for a grammar to be *regular.*

(ii) Let $G = (\Sigma, V, P, S)$ be a regular grammar and $\Omega = \Sigma \cup V$. Prove that if $\alpha \in \Omega^*$ and $S \xrightarrow{G} \alpha$, then there are $w \in \mathbb{W}$ and $A \in V$ such that $\alpha = wA$ or $\alpha = w$.

(iii) Let $G = (\Sigma, V, P, S)$ be a regular grammar, $A, B \in V$, and $w, v \in \mathbb{W}$. Prove that if $wA \xrightarrow{G} vB$, then there is some word $u \in \mathbb{W}$ such that $A \xrightarrow{G} uB$.

If $G = (\Sigma, V, P, S)$ is a regular grammar and A is a variable, we call A *accessible in* G if there is a word $w_1 \in \Sigma^*$ such that $S \xrightarrow{G} w_1A$; we call A *looping in* G if there is a word $w_2 \in \Sigma^*$ such that $A \xrightarrow{G} w_2A$; we call A *terminable in* G if there is a word $w_3 \in \Sigma^*$ such that $A \xrightarrow{G} w_3$.

(iv) Let G be a regular grammar. Prove that if $\mathcal{L}(G)$ is infinite then there is a variable that is accessible, looping, and terminable in G.

Paper 1, Section II

12I Automata & Formal Languages

Let Σ be an alphabet, $\mathbb{W}$ the set of words over Σ, $A, B \subseteq \mathbb{W}$, and $\mathcal{C}$ any set of subsets of $\mathbb{W}$.

(i) Define what $A \leqslant_{\mathrm{m}} B$ means.

(ii) Define what it means for a set A to be $\mathcal{C}$-*complete.*

(iii) Define what it means for A to be in Σ_1.

(iv) Define the *halting problem* $\mathbf{K}$.

(v) Prove that the halting problem $\mathbf{K}$ is Σ_1-complete.

A set $P \subseteq \mathbb{W}$ is in Π_2 if and only if there is a computable partial function $f : \mathbb{W} \times \mathbb{W} \dashrightarrow \mathbb{W}$ such that for all $w \in \mathbb{W}$, we have that $w \in P$ if and only if for all $v \in \mathbb{W}$, $f(w, v)\downarrow$.

(vi) We define $\mathbf{Tot} \subset \mathbb{W}$ to be the set $\{v\,;\, \mathrm{W}_v = \mathbb{W}\}$. Show that $\mathbf{Tot}$ is Π_2-complete.

[*In this entire question, you are allowed to use the fact that truncated computation functions are computable, provided that you give a precise and correct definition of the function used. You may use the partial function $f_{w,1}$ without providing a definition.*]

Paper 3, Section II

12I Automata & Formal Languages

Let Σ be an alphabet and $\mathbb{W}$ the set of words over Σ. Let $D = (\Sigma, Q, \delta, q_0, F)$ be a deterministic automaton.

(i) Define $\mathcal{L}(D)$, the set of words accepted by the automaton D, precisely defining all auxiliary functions needed for your definition.

(ii) State the *pumping lemma* for the language $\mathcal{L}(D)$. Specify the pumping number precisely in terms of D.

[No proof is required.]

(iii) Let $\Sigma = \{a, b\}$. Consider the regular language

$$L := \{wa^k\,;\, w \in \Sigma^* \text{ with } |w| \leqslant 10 \text{ and } k > 0\}.$$

Show that the minimal deterministic automaton for L has at least ten states.

Let $A \subseteq \mathbb{W}$. Define an equivalence relation on $\mathbb{W}$ by

$$v \sim_A w :\iff \text{ for all } u, \text{ we have } vu \in A \text{ if and only if } wu \in A.$$

(iv) Let $A \subseteq \mathbb{W}\backslash\{\varepsilon\}$. Show that A is a regular language if and only if the relation $\sim_A$ has finitely many equivalence classes.

Paper 1, Section I

4I Automata and Formal Languages

What are the nth *register machine* P_n and the nth *recursively enumerable set* W_n?

Given subsets $A, B \subseteq \mathbb{N}$, define a *many–one reduction* $A \leqslant_m B$ of A to B.

State *Rice's theorem.*

Is there a total algorithm that, on input n in register 1 and m in register 2, terminates with 0 if $W_m = W_n$ and 1 if $W_m \neq W_n$? Is there a partial algorithm that, with the same inputs as above, terminates with 0 if $W_m = W_n$ and never halts if $W_m \neq W_n$? Justify your answers.

[*You may assume without proof that the halting set* $\mathbb{K}$ *is not recursive.*]

Paper 2, Section I

4I Automata and Formal Languages

State and prove the *pumping lemma for regular languages.*

Are the following languages over the alphabet $\Sigma = \{0, 1\}$ regular? Justify your answers.

(i) $\{0^n 1 \mid n \geqslant 0\}$.

(ii) $\{0^n 1^{n^2} \mid n \geqslant 0\}$.

(iii) The set of all words in Σ^* containing the same number of 0s and 1s.

Paper 3, Section I

4I Automata and Formal Languages

Define a *context-free grammar* (CFG) and a *context-free language* (CFL).

State the *pumping lemma for CFLs.*

Which of the following languages over the alphabet $\{a, b, c\}$ are CFLs? Justify your answers.

(i) $\{a^n b^{2n} c^n \mid n \geqslant 0\}$.

(ii) $\{a^n b^{2i} c^n \mid n, i \geqslant 0\}$.

Paper 4, Section I

4I Automata and Formal Languages

Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form.*

What are an ϵ*–production* and a *unit production*?

Let G_1 be the CFG

$$\begin{aligned} S &\rightarrow \epsilon \mid aTa \mid bTa \\ T &\rightarrow Ta \mid Tb \mid c \end{aligned}$$

and let G_2 be the CFG

$$\begin{aligned} S &\rightarrow XZ \mid YZ \\ T &\rightarrow TX \mid TY \mid c \\ X &\rightarrow a,\, Y \rightarrow b,\, Z \rightarrow TX. \end{aligned}$$

What is the relationship between the language of G_1 and the language of G_2? Justify your answer carefully.

Paper 1, Section II

12I Automata and Formal Languages

Give the definition of a *primitive recursive function* $f : \mathbb{N}^k \rightarrow \mathbb{N}$.

Show directly from the definition that, when $k = 2$, the functions

$$P(m,n) = m + n \text{ and } T(m,n) = mn$$

are both primitive recursive.

Show further that for $k \geqslant 2$ the function

$$T_k(n_1, \ldots, n_k) = n_1 \cdots n_k$$

is primitive recursive, as is $E_a : \mathbb{N} \rightarrow \mathbb{N}$ given by $E_a(n) = a^n$, where $a \geqslant 1$ is a fixed integer.

Suppose $F : \mathbb{N}^k \rightarrow \mathbb{N}^k$, where $F = (f_0, \ldots, f_{k-1})$ with each coordinate function f_i primitive recursive. Describe how F can be encoded as a primitive recursive function $\overline{F} : \mathbb{N} \rightarrow \mathbb{N}$.

Let the Fibonacci function $B : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $B(0) = 0, B(1) = 1$ and $B(n+2) = B(n+1) + B(n)$ for $n \geqslant 0$. Is B primitive recursive? Justify your answer.

If $f : \mathbb{N} \rightarrow \mathbb{N}$ is a primitive recursive function, must there exist some $R > 0$ such that $f(n) \leqslant R^n$ for all $n \geqslant 1$? Justify your answer.

[*You may use without proof that for fixed $j \geqslant 2$ the maxpower function M_j is primitive recursive, where $M_j(n)$ is the exponent of the highest power of j that divides n. If you use any other results from the course, you should prove them.*]

Paper 3, Section II
12I Automata and Formal Languages

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA).

What does it mean to say that $q \in Q$ is an *accessible* state? What does it mean to say that $p, q \in Q$ are *equivalent* states?

Explain the construction of the *minimal DFA* $D/{\sim}$ and show that the languages of D and of $D/{\sim}$ are the same. Show also that no two distinct states of $D/{\sim}$ are equivalent.

Now let Σ be the single-letter alphabet $\{1\}$. Suppose that D is a DFA with no inaccessible states and exactly one accept state. Justifying your answer, describe the corresponding minimal DFA $D/{\sim}$ in the form of a transition diagram or otherwise. [Remember that you need only consider accessible states.]

Paper 1, Section I

4F Automata and Formal Languages

Let $f_{n,k}$ be the partial function on k variables that is computed by the nth machine (or the empty function if n does not encode a machine).

Define the *halting set* $\mathbb{K}$.

Given $A, B \subseteq \mathbb{N}$, what is a *many–one reduction* $A \leqslant_m B$ of A to B?

State the $s - m - n$ theorem and use it to show that a subset X of $\mathbb{N}$ is recursively enumerable if and only if $X \leqslant_m \mathbb{K}$.

Give an example of a set $S \subseteq \mathbb{N}$ with $\mathbb{K} \leqslant_m S$ but $\mathbb{K} \neq S$.

[*You may assume that* $\mathbb{K}$ *is recursively enumerable and that* $0 \notin \mathbb{K}$.]

Paper 2, Section I

4F Automata and Formal Languages

Assuming the definition of a deterministic finite-state automaton (DFA) $D = (Q, \Sigma, \delta, q_0, F)$, what is the *extended transition function* $\hat{\delta}$ for D? Also assuming the definition of a nondeterministic finite-state automaton (NFA) N, what is $\hat{\delta}$ in this case?

Define the *languages* accepted by D and N, respectively, in terms of $\hat{\delta}$.

Given an NFA N as above, describe the *subset construction* and show that the resulting DFA $\overline{N}$ accepts the same language as N. If N has one accept state then how many does $\overline{N}$ have?

Paper 3, Section I

4F Automata and Formal Languages

Define a *regular expression* R and explain how this gives rise to a language $\mathcal{L}(R)$.

Define a *deterministic finite-state automaton* D and the language $\mathcal{L}(D)$ that it accepts.

State the relationship between languages obtained from regular expressions and languages accepted by deterministic finite-state automata.

Let L and M be regular languages. Is $L \cup M$ always regular? What about $L \cap M$?

Now suppose that $L_1, L_2, \ldots$ are regular languages. Is the countable union $\bigcup L_i$ always regular? What about the countable intersection $\bigcap L_i$?

Paper 4, Section I

4F Automata and Formal Languages

State the pumping lemma for regular languages.

Which of the following languages over the alphabet $\{0,1\}$ are regular?

(i) $\{0^i 1^i 01 \mid i \geqslant 0\}$.

(ii) $\{w\overline{w} \mid w \in \{0,1\}^*\}$ where $\overline{w}$ is the reverse of the word w.

(iii) $\{w \in \{0,1\}^* \mid w$ does not contain the subwords 01 or 10$\}$.

Paper 1, Section II

12F Automata and Formal Languages

For $k \geqslant 1$ give the definition of a *partial recursive* function $f : \mathbb{N}^k \to \mathbb{N}$ in terms of basic functions, composition, recursion and minimisation.

Show that the following partial functions from $\mathbb{N}$ to $\mathbb{N}$ are partial recursive:

(i) $s(n) = \begin{cases} 1 & n = 0 \\ 0 & n \geqslant 1\,, \end{cases}$

(ii) $r(n) = \begin{cases} 1 & n \text{ odd} \\ 0 & n \text{ even}\,, \end{cases}$

(iii) $p(n) = \begin{cases} \text{undefined if } n \text{ is odd} \\ 0 \text{ if } n \text{ is even}\,. \end{cases}$

Which of these can be defined without using minimisation?

What is the class of functions $f : \mathbb{N}^k \to \mathbb{N}$ that can be defined using only basic functions and composition? [*Hint: See which functions you can obtain and then show that these form a class that is closed with respect to the above.*]

Show directly that every function in this class is computable.

Paper 3, Section II

12F Automata and Formal Languages

Suppose that G is a context-free grammar without ϵ-productions. Given a derivation of some word w in the language L of G, describe a *parse tree* for this derivation.

State and prove the pumping lemma for L. How would your proof differ if you did not assume that G was in Chomsky normal form, but merely that G has no ϵ- or unit productions?

For the alphabet $\Sigma = \{a, b\}$ of terminal symbols, state whether the following languages over Σ are context free, giving reasons for your answer.

(i) $\{a^i b^i a^i \mid i \geqslant 0\}$,

(ii) $\{a^i b^j \mid i \geqslant j \geqslant 0\}$,

(iii) $\{wabw \mid w \in \{a, b\}^*\}$.

Paper 1, Section I

4F Automata and Formal Languages

Define an *alphabet* Σ, a *word* over Σ and a *language* over Σ.

What is a *regular expression* R and how does this give rise to a language $\mathcal{L}(R)$?

Given any alphabet Σ, show that there exist languages L over Σ which are not equal to $\mathcal{L}(R)$ for any regular expression R. [You are not required to exhibit a specific L.]

Paper 2, Section I

4F Automata and Formal Languages

Assuming the definition of a partial recursive function from $\mathbb{N}$ to $\mathbb{N}$, what is a *recursive subset* of $\mathbb{N}$? What is a *recursively enumerable* subset of $\mathbb{N}$?

Show that a subset $E \subseteq \mathbb{N}$ is recursive if and only if E and $\mathbb{N} \setminus E$ are recursively enumerable.

Are the following subsets of $\mathbb{N}$ recursive?

(i) $\mathbb{K} := \{n \mid n$ codes a program and $f_{n,1}(n)$ halts at some stage$\}$.

(ii) $\mathbb{K}_{100} := \{n \mid n$ codes a program and $f_{n,1}(n)$ halts within 100 steps$\}$.

Paper 3, Section I

4F Automata and Formal Languages

Define a *context-free grammar* G, a *sentence* of G and the *language* $\mathcal{L}(G)$ generated by G.

For the alphabet $\Sigma = \{a, b\}$, which of the following languages over Σ are context-free?

(i) $\{a^{2m}b^{2m} \mid m \geqslant 0\}$,

(ii) $\{a^{m^2}b^{m^2} \mid m \geqslant 0\}$.

[*You may assume standard results without proof if clearly stated.*]

Paper 4, Section I

4F Automata and Formal Languages

Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form* (CNF).

Describe without proof each stage in the process of converting a CFG $G = (N, \Sigma, P, S)$ into an equivalent CFG $\overline{G}$ which is in CNF. For each of these stages, when are the nonterminals N left unchanged? What about the terminals Σ and the generated language $\mathcal{L}(G)$?

Give an example of a CFG G whose generated language $\mathcal{L}(G)$ is infinite and equal to $\mathcal{L}(\overline{G})$.

Paper 1, Section II

12F Automata and Formal Languages

(a) Define a *register machine*, a *sequence of instructions* for a register machine and a *partial computable* function. How do we encode a register machine?

(b) What is a *partial recursive* function? Show that a partial computable function is partial recursive. [You may assume that for a given machine with a given number of inputs, the function outputting its state in terms of the inputs and the time t is recursive.]

(c) (i) Let $g : \mathbb{N} \to \mathbb{N}$ be the partial function defined as follows: if n codes a register machine and the ensuing partial function $f_{n,1}$ is defined at n, set $g(n) = f_{n,1}(n) + 1$. Otherwise set $g(n) = 0$. Is g a partial computable function?

(ii) Let $h : \mathbb{N} \to \mathbb{N}$ be the partial function defined as follows: if n codes a register machine and the ensuing partial function $f_{n,1}$ is defined at n, set $h(n) = f_{n,1}(n) + 1$. Otherwise, set $h(n) = 0$ if n is odd and let $h(n)$ be undefined if n is even. Is h a partial computable function?

Paper 3, Section II

12F Automata and Formal Languages

Give the definition of a *deterministic finite state automaton* and of a *regular language*.

State and prove the pumping lemma for regular languages.

Let $S = \{2^n \mid n = 0, 1, 2, \ldots\}$ be the subset of $\mathbb{N}$ consisting of the powers of 2. If we write the elements of S in base 2 (with no preceding zeros), is S a regular language over $\{0, 1\}$?

Now suppose we write the elements of S in base 10 (again with no preceding zeros). Show that S is not a regular language over $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. [*Hint: Give a proof by contradiction; use the above lemma to obtain a sequence $a_1, a_2, \ldots$ of powers of 2, then consider $a_{i+1} - 10^d a_i$ for $i = 1, 2, 3, \ldots$ and a suitable fixed d.*]

Paper 1, Section I

4H Automata and Formal Languages

(a) State the *pumping lemma* for context-free languages (CFLs).

(b) Which of the following are CFLs? Justify your answers.

(i) $\{ww^R \mid w \in \{a,b\}^*\}$, where w^R is the reverse of the word w.

(ii) $\{0^p 1^p \mid p \text{ is a prime}\}$.

(iii) $\{a^m b^n c^k d^l \mid 3m = 4l \text{ and } 2n = 5k\}$.

(c) Let L and M be CFLs. Show that the concatenation LM is also a CFL.

Paper 4, Section I

4H Automata and Formal Languages

(a) Which of the following are regular languages? Justify your answers.

(i) $\{w^n \mid w \in \{a,b\}^*,\ n \geqslant 2\}$.

(ii) $\{w \in \{a,b,c\}^* \mid w \text{ contains an odd number of } b\text{'s and an even number of } c\text{'s}\}$.

(iii) $\{w \in \{0,1\}^* \mid w \text{ contains no more than 7 consecutive 0's}\}$.

(b) Consider the language L over alphabet $\{a,b\}$ defined via

$$L := \{wab^n \mid w \in \{a,b\}^*,\ n \in \mathbb{K}\} \cup \{b\}^*.$$

Show that L satisfies the pumping lemma for regular languages but is not a regular language itself.

Paper 3, Section I

4H Automata and Formal Languages

(a) Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form* (CNF). Can a CFG in CNF ever define a language containing ϵ? If G_{Chom} denotes the result of converting an arbitrary CFG G into one in CNF, state the relationship between $\mathcal{L}(G)$ and $\mathcal{L}(G_{\text{Chom}})$.

(b) Let G be a CFG in CNF. Give an algorithm that, on input of any word v on the terminals of G, decides if $v \in \mathcal{L}(G)$ or not. Explain why your algorithm works.

(c) Convert the following CFG G into a grammar in CNF:

$$S \to \; Sbb \mid aS \mid T$$

$$T \to cc$$

Does $\mathcal{L}(G) = \mathcal{L}(G_{\text{Chom}})$ in this case? Justify your answer.

Paper 2, Section I

4H Automata and Formal Languages

(a) Define a *recursive set* and a *recursively enumerable (r.e.) set.* Prove that $E \subseteq \mathbb{N}_0$ is recursive if and only if both E and $\mathbb{N}_0 \setminus E$ are r.e. sets.

(b) Let $E = \{f_{n,k}(m_1, \ldots, m_k) \mid (m_1, \ldots, m_k) \in \mathbb{N}_0^k\}$ for some fixed $k \geqslant 1$ and some fixed register machine code n. Show that $E = \{m \in \mathbb{N}_0 \mid f_{j,1}(m) \downarrow\}$ for some fixed register machine code j. Hence show that E is an r.e. set.

(c) Show that the function $f : \mathbb{N}_0 \to \mathbb{N}_0$ defined below is primitive recursive.

$$f(n) = \begin{cases} n-1 & \text{if } n > 0 \\ 0 & \text{if } n = 0. \end{cases}$$

[*Any use of Church's thesis in your answers should be explicitly stated. In this question* $\mathbb{N}_0$ *denotes the set of non-negative integers.*]

Paper 1, Section II

12H Automata and Formal Languages

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA). Define what it means for two states of D to be *equivalent*. Define the *minimal* DFA $D/\sim$ for D.

Let D be a DFA with no inaccessible states, and suppose that A is another DFA on the same alphabet as D and for which $\mathcal{L}(D) = \mathcal{L}(A)$. Show that A has at least as many states as $D/\sim$. [You may use results from the course as long as you state them clearly.]

Construct a minimal DFA (that is, one with the smallest possible number of states) over the alphabet $\{0, 1\}$ which accepts precisely the set of binary numbers which are multiples of 7. You may have leading zeros in your inputs (e.g.: 00101). Prove that your DFA is minimal by finding a distinguishing word for each pair of states.

Paper 3, Section II

12H Automata and Formal Languages

(a) State the *s-m-n theorem* and the *recursion theorem*.

(b) State and prove *Rice's theorem*.

(c) Show that if $g : \mathbb{N}_0^2 \to \mathbb{N}_0$ is partial recursive, then there is some $e \in \mathbb{N}_0$ such that

$$f_{e,1}(y) = g(e, y) \quad \forall y \in \mathbb{N}_0.$$

(d) Show there exists some $m \in \mathbb{N}_0$ such that W_m has exactly m^2 elements.

(e) Given $n \in \mathbb{N}_0$, is it possible to compute whether or not the number of elements of W_n is a (finite) perfect square? Justify your answer.

[*In this question* $\mathbb{N}_0$ *denotes the set of non-negative integers. Any use of Church's thesis in your answers should be explicitly stated.*]

Paper 4, Section I

4G Automata and Formal Languages

(a) State the s-m-n theorem, the recursion theorem, and Rice's theorem.

(b) Show that if $g : \mathbb{N}^2 \to \mathbb{N}$ is partial recursive, then there is some $e \in \mathbb{N}$ such that

$$f_{e,1}(y) = g(e, y) \quad \forall y \in \mathbb{N}.$$

(c) By considering the partial function $g : \mathbb{N}^2 \to \mathbb{N}$ given by

$$g(x, y) = \begin{cases} 0 & \text{if } y < x \\ \uparrow & \text{otherwise,} \end{cases}$$

show there exists some $m \in \mathbb{N}$ such that W_m has exactly m elements.

(d) Given $n \in \mathbb{N}$, is it possible to compute whether or not W_n has exactly 9 elements? Justify your answer.

[*Note that we define* $\mathbb{N} = \{0, 1, \ldots\}$. *Any use of Church's thesis in your answers should be explicitly stated.*]

Paper 3, Section I

4G Automata and Formal Languages

(a) Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form* (CNF).

(b) Give an algorithm for converting a CFG G into a corresponding CFG G_{Chom} in CNF satisfying $\mathcal{L}(G_{\text{Chom}}) = \mathcal{L}(G) - \{\epsilon\}$. [You need only outline the steps, without proof.]

(c) Convert the following CFG G:

$$S \to ASc \mid B \ , \quad A \to a \ , \quad B \to b \ \ ,$$

into a grammar in CNF.

Paper 2, Section I

4G Automata and Formal Languages

(a) Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ be a nondeterministic finite-state automaton with ϵ-transitions (ϵ-NFA). Define the deterministic finite-state automaton (DFA) $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ obtained from E via the subset construction with ϵ-transitions.

(b) Let E and D be as above. By inducting on lengths of words, prove that

$$\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w) \text{ for all } w \in \Sigma^*.$$

(c) Deduce that $\mathcal{L}(D) = \mathcal{L}(E)$.

Paper 1, Section I

4G Automata and Formal Languages

(a) State the pumping lemma for context-free languages (CFLs).

(b) Which of the following are CFLs? Justify your answers.

(i) $\{ww \mid w \in \{a, b, c\}^*\}$

(ii) $\{a^m b^n c^k d^l \mid 3m = 4n \text{ and } 2k = 5l\}$

(iii) $\{a^{3^n} \mid n \geqslant 0\}$

(c) Let L be a CFL. Show that L^* is also a CFL.

Paper 3, Section II

12G Automata and Formal Languages

(a) State and prove the pumping lemma for regular languages.

(b) Let D be a minimal deterministic finite-state automaton whose language $\mathcal{L}(D)$ is finite. Let Γ_D be the transition diagram of D, and suppose there exists a non-empty closed path γ in Γ_D starting and ending at state p.

(i) Show that there is no path in Γ_D from p to any accept state of D.

(ii) Show that there is no path in Γ_D from p to any other state of D.

Paper 1, Section II

12G Automata and Formal Languages

(a) Define the *halting set* $\mathbb{K}$. Prove that $\mathbb{K}$ is recursively enumerable, but not recursive.

(b) Given $A, B \subseteq \mathbb{N}$, define a *many-one reduction* of A to B. Show that if B is recursively enumerable and $A \leqslant_m B$, then A is also recursively enumerable.

(c) Show that each of the functions $f(n) = 2n$ and $g(n) = 2n + 1$ are both *partial recursive* and *total*, by building them up as partial recursive functions.

(d) Let $X, Y \subseteq \mathbb{N}$. We define the set $X \oplus Y$ via

$$X \oplus Y := \{2x \mid x \in X\} \cup \{2y + 1 \mid y \in Y\}.$$

(i) Show that both $X \leqslant_m X \oplus Y$ and $Y \leqslant_m X \oplus Y$.

(ii) Using the above, or otherwise, give an explicit example of a subset C of $\mathbb{N}$ for which neither C nor $\mathbb{N} \setminus C$ are recursively enumerable.

(iii) For every $Z \subseteq \mathbb{N}$, show that if $X \leqslant_m Z$ and $Y \leqslant_m Z$ then $X \oplus Y \leqslant_m Z$.

[*Note that we define* $\mathbb{N} = \{0, 1, \ldots\}$. *Any use of Church's thesis in your answers should be explicitly stated.*]

Paper 1, Section I

4H Automata and Formal Languages

(a) Prove that every regular language is also a context-free language (CFL).

(b) Show that, for any fixed $n > 0$, the set of regular expressions over the alphabet $\{a_1, \ldots, a_n\}$ is a CFL, but not a regular language.

Paper 2, Section I

4H Automata and Formal Languages

(a) Give explicit examples, with justification, of a language over some finite alphabet Σ which is:

  (i) context-free, but not regular;

  (ii) recursive, but not context-free.

(b) Give explicit examples, with justification, of a subset of $\mathbb{N}$ which is:

  (i) recursively enumerable, but not recursive;

  (ii) neither recursively enumerable, nor having recursively enumerable complement in $\mathbb{N}$.

Paper 3, Section I

4H Automata and Formal Languages

(a) Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form* (CNF). Give an example, with justification, of a context-free language (CFL) which is not defined by any CFG in CNF.

(b) Show that the intersection of two CFLs need not be a CFL.

(c) Let L be a CFL over an alphabet Σ. Show that $\Sigma^* \setminus L$ need not be a CFL.

Paper 4, Section I

4H Automata and Formal Languages

(a) Describe the process for converting a deterministic finite-state automaton D into a regular expression R defining the same language, $\mathcal{L}(D) = \mathcal{L}(R)$. [You need only outline the steps, without proof, but you should clearly define all terminology you introduce.]

(b) Consider the language L over the alphabet $\{0,1\}$ defined via

$$L := \{w01^n \mid w \in \{0,1\}^*, n \in \mathbb{K}\} \cup \{1\}^*.$$

Show that L satisfies the pumping lemma for regular languages but is not a regular language itself.

Paper 1, Section II

11H Automata and Formal Languages

(a) Give an *encoding* to integers of all deterministic finite-state automata (DFAs). [Here the alphabet of each such DFA is always taken from the set $\{0,1,\ldots\}$, and the states for each such DFA are always taken from the set $\{q_0, q_1, \ldots\}$.]

(b) Show that the set of codes for which the corresponding DFA D_n accepts a *finite* language is recursive. Moreover, if the language $\mathcal{L}(D_n)$ is finite, show that we can compute its size $|\mathcal{L}(D_n)|$ from n.

Paper 3, Section II

11H Automata and formal languages

(a) Given $A, B \subseteq \mathbb{N}$, define a *many-one reduction* of A to B. Show that if B is recursively enumerable (r.e.) and $A \leqslant_m B$ then A is also recursively enumerable.

(b) State the *s-m-n* theorem, and use it to prove that a set $X \subseteq \mathbb{N}$ is r.e. if and only if $X \leqslant_m \mathbb{K}$.

(c) Consider the sets of integers $P, Q \subseteq \mathbb{N}$ defined via

$$P := \{n \in \mathbb{N} \mid n \text{ codes a program and } W_n \text{ is finite}\}$$
$$Q := \{n \in \mathbb{N} \mid n \text{ codes a program and } W_n \text{ is recursive}\}.$$

Show that $P \leqslant_m Q$.

Paper 4, Section I

4F Automata and Formal Languages

(a) Construct a register machine to compute the function $f(m,n) := m+n$. State the relationship between partial recursive functions and partial computable functions. Show that the function $g(m,n) := mn$ is partial recursive.

(b) State Rice's theorem. Show that the set $A := \{n \in \mathbb{N} \mid |W_n| > 7\}$ is recursively enumerable but not recursive.

Paper 3, Section I

4F Automata and Formal Languages

(a) Define what it means for a context-free grammar (CFG) to be in *Chomsky normal form* (CNF). Can a CFG in CNF ever define a language containing ϵ? If G_{Chom} denotes the result of converting an arbitrary CFG G into one in CNF, state the relationship between $\mathcal{L}(G)$ and $\mathcal{L}(G_{\text{Chom}})$.

(b) Let G be a CFG in CNF, and let $w \in \mathcal{L}(G)$ be a word of length $|w| = n > 0$. Show that every derivation of w in G requires precisely $2n-1$ steps. Using this, give an algorithm that, on input of any word v on the terminals of G, decides if $v \in \mathcal{L}(G)$ or not.

(c) Convert the following CFG G into a grammar in CNF:

$$S \to \ aSb \mid SS \mid \epsilon\,.$$

Does $\mathcal{L}(G) = \mathcal{L}(G_{\text{Chom}})$ in this case? Justify your answer.

Paper 2, Section I

4F Automata and Formal Languages

(a) Which of the following are regular languages? Justify your answers.

(i) $\{w \in \{a,b\}^* \mid w$ is a nonempty string of alternating a's and b's$\}$.

(ii) $\{wabw \mid w \in \{a,b\}^*\}$.

(b) Write down a nondeterministic finite-state automaton with ϵ-transitions which accepts the language given by the regular expression $(\mathbf{a}+\mathbf{b})^*(\mathbf{bb}+\mathbf{a})\mathbf{b}$. Describe in words what this language is.

(c) Is the following language regular? Justify your answer.

$$\{w \in \{a,b\}^* \mid w \text{ does not end in } ab \text{ or } bbb\}.$$

Paper 1, Section I

4F Automata and Formal Languages

State the *pumping lemma* for context-free languages (CFLs). Which of the following are CFLs? Justify your answers.

(i) $\{a^{2n}b^{3n} \mid n \geqslant 3\}$.

(ii) $\{a^{2n}b^{3n}c^{5n} \mid n \geqslant 0\}$.

(iii) $\{a^p \mid p \text{ is a prime}\}$.

Let L, M be CFLs. Show that $L \cup M$ is also a CFL.

Paper 3, Section II

11F Automata and Formal Languages

(a) Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite-state automaton. Define the *extended transition function* $\hat{\delta} : Q \times \Sigma^* \to Q$, and the *language accepted by* D, denoted $\mathcal{L}(D)$. Let $u, v \in \Sigma^*$, and $p \in Q$. Prove that $\hat{\delta}(p, uv) = \hat{\delta}(\hat{\delta}(p, u), v)$.

(b) Let $\sigma_1, \sigma_2, \ldots, \sigma_k \in \Sigma$ where $k \geqslant |Q|$, and let $p \in Q$.

(i) Show that there exist $0 \leqslant i < j \leqslant k$ such that $\hat{\delta}(p, \sigma_1 \cdots \sigma_i) = \hat{\delta}(p, \sigma_1 \cdots \sigma_j)$, where we interpret $\sigma_1 \cdots \sigma_i$ as ϵ if $i = 0$.

(ii) Show that $\hat{\delta}(p, \sigma_1 \cdots \sigma_i \sigma_{j+1} \cdots \sigma_k) = \hat{\delta}(p, \sigma_1 \cdots \sigma_k)$.

(iii) Show that $\hat{\delta}(p, \sigma_1 \cdots \sigma_i (\sigma_{i+1} \cdots \sigma_j)^t \sigma_{j+1} \cdots \sigma_k) = \hat{\delta}(p, \sigma_1 \cdots \sigma_k)$ for all $t \geqslant 1$.

(c) Prove the following version of the pumping lemma. Suppose $w \in \mathcal{L}(D)$, with $|w| \geqslant |Q|$. Then w can be broken up into three words $w = xyz$ such that $y \neq \epsilon$, $|xy| \leqslant |Q|$, and for all $t \geqslant 0$, the word xy^tz is also in $\mathcal{L}(D)$.

(d) Hence show that

(i) if $\mathcal{L}(D)$ contains a word of length at least $|Q|$, then it contains infinitely many words;

(ii) if $\mathcal{L}(D) \neq \emptyset$, then it contains a word of length less than $|Q|$;

(iii) if $\mathcal{L}(D)$ contains all words in Σ^* of length less than $|Q|$, then $\mathcal{L}(D) = \Sigma^*$.

Paper 1, Section II

11F Automata and Formal Languages

(a) Define a *recursive set* and a *recursively enumerable (r.e.) set*. Prove that $E \subseteq \mathbb{N}$ is recursive if and only if both E and $\mathbb{N} \setminus E$ are r.e.

(b) Define *the halting set* $\mathbb{K}$. Prove that $\mathbb{K}$ is r.e. but not recursive.

(c) Let $E_1, E_2, \ldots, E_n$ be r.e. sets. Prove that $\bigcup_{i=1}^{n} E_i$ and $\bigcap_{i=1}^{n} E_i$ are r.e. Show by an example that the union of infinitely many r.e. sets need not be r.e.

(d) Let E be a recursive set and $f : \mathbb{N} \to \mathbb{N}$ a (total) recursive function. Prove that the set $\{f(n) \mid n \in E\}$ is r.e. Is it necessarily recursive? Justify your answer.

[*Any use of Church's thesis in your answer should be explicitly stated.*]