SEMESTER 1 EXAMINATION 2022/2023

THEORY OF COMPUTING

DURATION 120 MINS (2 Hours)

This paper is divided into Section A and Section B.

Section A contains FOUR questions. You must answer all of them. Each question from this section is worth 10 marks.

Section B contains FOUR questions. You must answer only THREE of them. Each question from this section is worth 20 marks.

Together, Section A and Section B add up to 100 marks, representing 100% of the module.

An outline marking scheme is shown in brackets to the right of each question.

Only university–approved calculators may be used.

6 page examination paper.

SECTION A

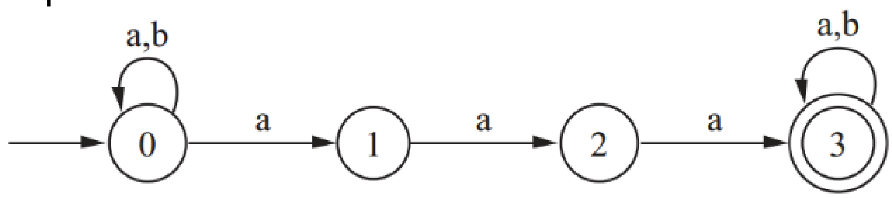
QUESTION 1

A Finite State Automaton (FSA) is an abstract machine that can be in exactly one of a finite number of states at any given time.

(a) Construct a Deterministic Finite Automaton (DFA) over the alphabet $\{a, b, c\}$ that does **not** recognize the string $aabb$.

(5 marks)

(b) Convert the Nondeterministic Finite Automata (NFA) below into its equivalent DFA.



(5 marks)

QUESTION 2

In the theory of computation, a pushdown automaton (PDA) is a type of automaton that employs a stack, while a context–free grammar (CFG) is a formal grammar which is used to generate all possible strings in a given formal language.

(a) Construct a PDA diagram for the language

$$L = \{a^n b^m c^m \mid n, m \geq 1\}.$$

(5 marks)

(b) Give context-free grammars that generate the language

$$L = \{w \in \{0,1\}^* \mid |w| \text{ odd , middle symbol is } 0 \}$$

(5 marks)

QUESTION 3

(a) Define *recursively enumerable* sets and *recursive sets*.

(4 marks)

(b) What is the relationship between recursively enumerable sets and recursive sets?

(2 marks)

(c) Give one example for each of the two types of sets in part (b).

(2 marks)

(d) State whether the following statement is true / false:
A language recognisable by a PDA is then also a recursively enumerable set.

(1 mark)

(e) State whether the following statement is true / false:
A set of strings that is reducible to a halting problem can then also be accepted by a Turing Machine.

(1 mark)

QUESTION 4

(a) Define what it means for a function $f(n)$ to be $O(g(n))$.

(2 marks)

(b) Give proof of small–o and big–O relationships between n^3 and $\log(n!)$.

(4 marks)

(c) Explain what is meant by saying that a decision problem is in

(i) P
(ii) NP

(4 marks)

SECTION B

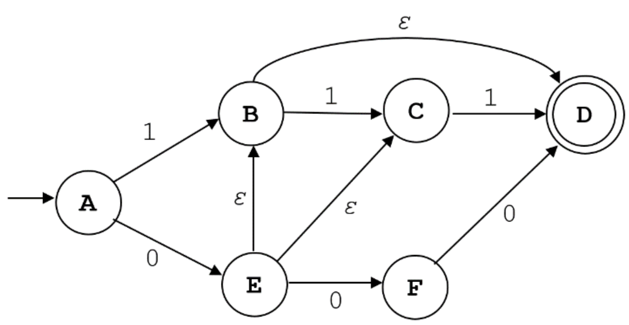
QUESTION 1

A language L that cannot be defined by a regular expression is a non-regular language or an irregular language.

(a) Prove that $L = \{0^n 1^m \mid n < 3m\}$ is not regular.

(10 marks)

(b) Convert the ε–NFA below to its equivalent NFA.
Show all the steps, transition tables, and an NFA diagram.



(10 marks)

QUESTION 2

(a) Let G be the grammar

$$(\{S, A\}, \{0,1\}, \{S \to 0SA \mid 0A \mid SA\,, A \to 1\}, S).$$

Is G a context–free grammar? Explain the reason for your statement. *(2 marks)*

(b) Is G in Greibach Normal Form? Explain the reason for your statement. *(2 marks)*

(c) Convert the grammar G into Chomsky Normal Form.
You only need to write all the production rules.

(4 marks)

(d) Show the leftmost derivation for 00111 from grammar in part (c). *(4 marks)*

(e) Convert the grammar G into an equivalent PDA. You only need to write all the transitions in the transition relations. *(4 marks)*

(f) Simulate the PDA on accepting string 00111 .

(4 marks)

QUESTION 3

(a) Define a Turing machine (TM) that accepts

$$L = \left\{ 0^{(2^n)} \,\middle|\, n \geq 0 \,, n \in \mathbb{Z} \right\}.$$

You are allowed to describe the idea of how the transition function works by using English words, but explicitly define the states and the symbols used in the tape.

(8 marks)

(b) Show how your TM accepts 00 .

(6 marks)

(c) Is the TM that you constructed a total TM? Briefly explain your answer. *(3 marks)*

(d) Is the problem of checking whether an integer x is a power of 2 a **decidable**, **semi–decidable** or **undecidable** problem? Briefly explain your answer.

(3 marks)

QUESTION 4

(a) Explain what is meant by saying that a decision problem is NP-complete.

(3 marks)

(b) Consider the following pair of decision problems:

(i) SAT: given a Boolean formula is there an assignment of values to the variables such that the formula is satisfied?

(ii) 3COL: given a graph G is it possible to colour the vertices with 3 colours such that no two adjacent vertices have the same colour?

Using the fact that 3COL is NP–complete, show that SAT is NP-complete.

Hint: for each vertex you may want to introduce variables for each of the 3 colours.

(13 marks)

(c) Show that any PSPACE–hard problem is also NP–hard.

(4 marks)

END OF PAPER