ST. PAUL'S UNIVERSITY

Private Bag - 00217 Limuru, Kenya
Tel. Office: 020-2020505/10; Mobile: 0728-669000
Website: www.spu.ac.ke

FACULTY OF BUSINESS, COMPUTER SCIENCE AND COMMUNICATION

BACHELOR OF COMPUTER SCIENCE

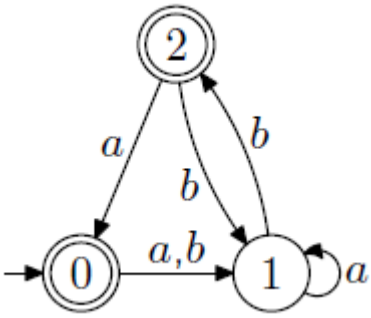
SEPTEMBER-DECEMBER 2019/2020 SEMESTER

CSC 2210 AUTOMATA THEORY

CAT

ANSWER ALL QUESTIONS

a) Convert the following NFA to a regular CFG: **(4 marks)**



Ans

$$S \to aA \mid bA \mid \varepsilon$$
$$A \to aA \mid bB$$
$$B \to aS \mid bA \mid \varepsilon$$

b) Mathematically define a DFA that accepts the language of all strings over alphabet $\{a, b\}$ in which each b is separated from the next by at least 100 a's. **(6 marks)**

Ans

Define $A = (Q, \{a, b\}, \delta, q_0, F)$ with $Q = [0, 100] \cap \mathbb{Z}$, $q_0 = 100$, $F = Q$, and

$$\delta = \{((100, a), 100),\ ((100, b), 0)\} \cup \{((q, a), q + 1) \mid q \in Q - \{100\}\}$$

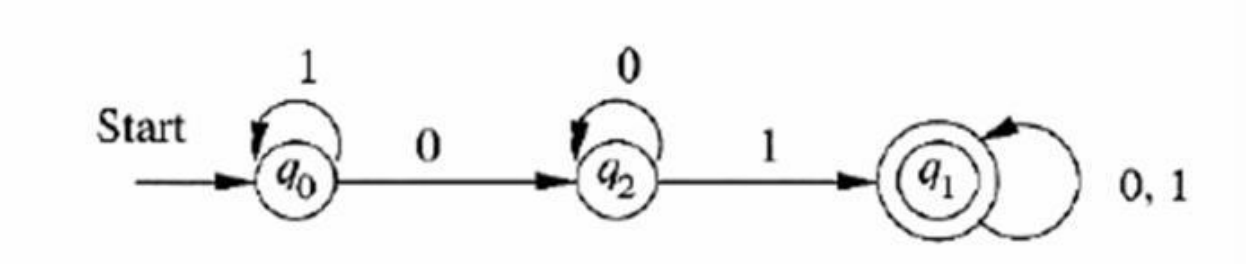
c) Write a CFG that generates the language of ALL palindromes over alphabet {*a, b*} that do not contain the substring *aa*. **(6 marks)**

Ans

$$S \rightarrow abSba \mid bSb \mid aba \mid a \mid b \mid \varepsilon$$

d) Obtain DFAs to accept strings of a's and b's having exactly one *a*. **(4 marks)**

Ans



e) Discuss FOUR applications of Finite Automata. **(4 marks)**

Ans

String Processing
Consider finding all occurrences of a short string (pattern string) within a long string (text string).
This can be done by processing the text through a DFA: the DFA for all strings that end with the pattern
string. Each time the accept state is reached, the current position in the text is output.
Finite-State Machines
A finite-state machine is an FA together with actions on the arcs.
Statecharts
Statecharts model tasks as a set of states and actions. They extend FA diagrams.
Lexical Analysis

In compiling a program, the first step is lexical analysis. This isolates keywords, identifiers etc., while
eliminating irrelevant symbols. A token is a category, for example "identifier", "relation operator" or specific keyword.

f) Explain decision properties of regular language. **(6 marks)**

Ans

To locate the regular languages in the Chomsky hierarchy, one notices that every regular language is context-free. The converse is not true: for example the language consisting of all strings having the same number of *a*'s as *b*'s is context-free but not regular. To prove that a language such as this is not regular, one often uses the Myhill–Nerode theorem or the pumping lemma among other methods.[5]

There are two purely algebraic approaches to define regular languages. If:

- Σ is a finite alphabet,
- Σ* denotes the free monoid over Σ consisting of all strings over Σ,
- $f : \Sigma^* \to M$ is a monoid homomorphism where M is a *finite* monoid,
- S is a subset of M

then the set $\{w \in \Sigma^* \mid f(w) \in S\}$ is regular. Every regular language arises in this fashion.

If L is any subset of Σ*, one defines an equivalence relation ~ (called the syntactic relation) on Σ* as follows: $u \sim v$ is defined to mean $uw \in L$ if and only if $vw \in L$ for all $w \in \Sigma^*$

The language L is regular if and only if the number of equivalence classes of ~ is finite (A proof of this is provided in the article on the syntactic monoid). When a language is regular, then the number of equivalence classes is equal to the number of states of the minimal deterministic finite automaton accepting L.

A similar set of statements can be formulated for a monoid $M \subset \Sigma^*$. In this case, equivalence over M leads to the concept of a recognizable language.