TOC TT#02 Assignment

Submitted From:

Emran Ahmed

Reg: 2020831040

Session: 2020-21

Submitted To:

Mohammed Raihan Ullah

Lecturer, IICT,

SUST.

Date of Submission: 21/11/2023

TT#02 Course: Theory of Computation (SWE 227)

Marks: 20 Time: **35 mins**

1. Why does the Finite Automata can't solve the counting problem but the PDA can? **- 04**
2. Give the formal definition of PDA. **- 02**
3. Write the Regular expression that matches the following types of patterns :
   "pencil#2","mambo#5","grade#8" **- 03**
4. Remove unit production from the following grammar. **- 05**
   S→XY, X→a, Y→Z|b, Z→M, M→N, N→a
5. Draw the PushDown Automata for the language **- 06**
   D = { $a^i b^j c^k$ | i, j, k ≥ 0, and i = j or j = k }

TOC TT#02: Assignment

Que: 1. Why does the Finite Automata can't solve the counting problem but the PDA can?

Solution: Finite Automata

(i) Finite Automata are limited in terms of memory. They hava a finite number of states and no additional memory storage like a stack.

(ii) Since finite automata lack the ability to store and retrieve an unbounded amount of information, they are unable to solve problems that require counting beyond a fixed, predeterministic threshold.

(iii) Finite Automata are particularly well-suited for recognizing regular languages which are languages that can be described by regular expression. Regular language typically involves pattern that do not require counting.

Pushdown Automata (PDA)

(i) PDA have access to a stack, which provides them with additional memory. The stack allows PDAs to keep track of information about the input string, making them more powerful than finite automata.

(ii) The ability to use a stack for storage enables PDAs to recognize context free language, which include languages with nested

structure and require a form of counting.

Ques-2: Give the formal definition of PDA.

Soln: A Pushdown Automaton (PDA) is formally defined as 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where:

$Q \rightarrow$ set of finite states

$\Sigma \rightarrow$ input alphabet

$\Gamma \rightarrow$ stack alphabet

$\delta \rightarrow$ transition function

$q_0 \rightarrow$ initial state

$F \rightarrow$ set of final states

A PDA process an input string, using its stack for additional memory, and accepts the input if it reaches an accepting state.

Ques-3: Write the regular expression that matches the followings types of pattern:

"pencil#2", "mambo#5", "grade#8".

Soln: The regular expression is -

/[a-z]+#[0-9]/

Que-4: Remove unit production from the following grammar.

$S \to XY$, $X \to a$, $Y \to Z \mid b$, $Z \to M$, $M \to N$, $N \to a$

Soln:

$S \to XY$
$X \to a$
$Y \to a \mid b$
$Z \to a$
$M \to a$
$N \to a$

Remove Unreachable Variables

$S \to XY$
$X \to a$
$Y \to a \mid b$

Que-5: Draw the PushDown Automata for the language:

$D = \{a^i b^j c^k \mid i, j, k \geq 0, \text{ and } i=j \text{ or } j=k\}$

Soln: