Birla Institute of Technology and Science, Pilani

Department of Computer Science and Information Systems

Comprehensive Exam, First Semester 2020-21

CS F351, Theory of Computation, <u>Open Book</u>

Date: 14-12-2020 **10 to 12 PM** **MM: 33**

Note: a) Answer the questions on A4 size sheet of papers (or equivalent), scan them, and upload.
b) Answer the questions sequentially, legitimately, and to the point without giving any unnecessary details.
c) #a(x) should be read as number of a's in string x.
d) Division of marks among parts of each question will depend upon the complexity.

Q1. Answer the following two parts together:

a) Convert the following context free grammar G into an equivalent CFG in Chomsky normal form. Illustrate all the steps in detail.

G = ({S, A, a}, {a}, {S→ASA | A | ε, A→aa | ε}, S)

b) Construct a pushdown automaton for the language L = {w | w ϵ {0, 1}* and w has the same number of 01's and 10's}. Give a formal 6-tuple specification of the PDA.
[Note: string 101 has one occurrence of 01 and one occurrence of 10.]

Q1 [5 Marks]. (a).

1. Introduce the new start variable "So" and new rule So → S, which yields

   So → S
   S → ASA | A | ε
   A → aa | ε

   (1 mark)
2. Then remove ε-productions/rules.
   a. Removing A → ε yields

      So → S
      S → ASA | AS | SA | S | A | ε
      A → aa

      (We can remove S → S as S obviously produces itself.)
   b. Removing S → ε yields

      So → S | ε
      S → ASA | AS | SA | A | AA
      A → aa

   (1 mark)
3. Then remove unit rules:
   a. Removing S → A yields:

      So → S | ε
      S → ASA | AS | SA | aa | AA
      A → aa
   b. Removing So → S yields:

      So → ASA | AS | SA | aa | AA | ε
      S → ASA | AS | SA | aa | AA
      A → aa

   (1 mark)
4. Shortening the right hand side of productions yields

   So → AA_1 | AS | SA | aa | AA | ε
   S → AA_1| AS | SA | aa | AA
   A → aa

$A_1 \rightarrow SA$

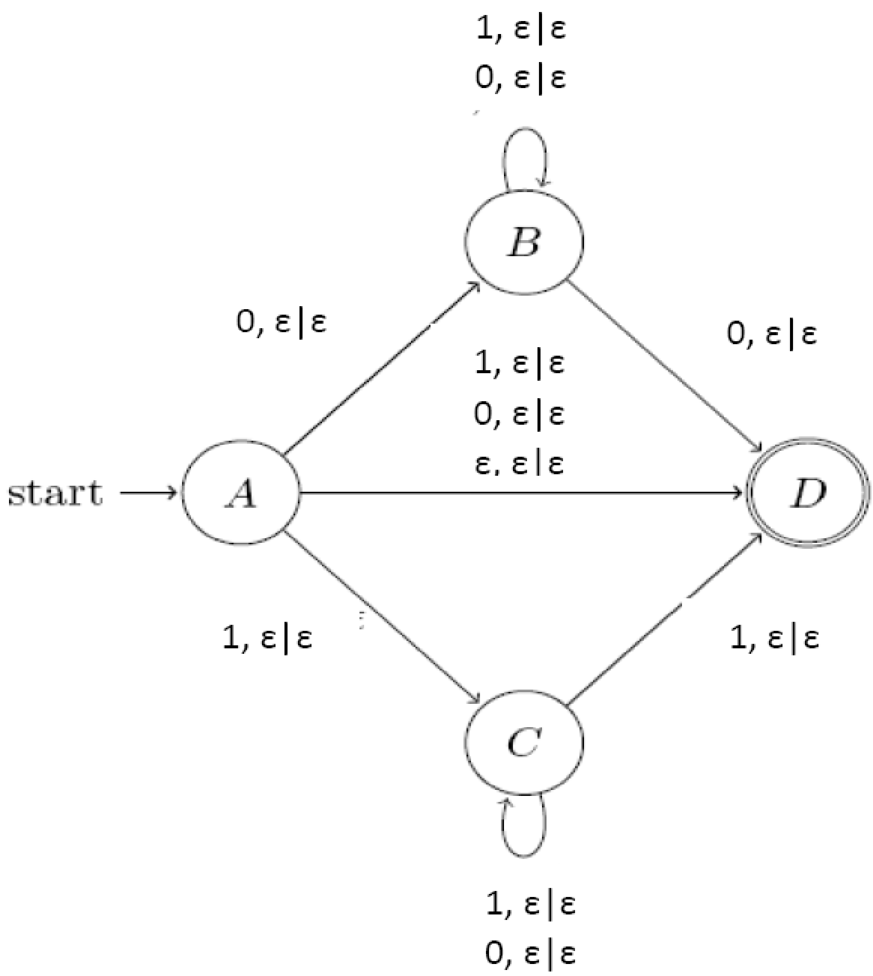
(1 mark)

5. Replacing few terminals on the right hand side yields

$S_0 \rightarrow AA_1 \mid AS \mid SA \mid BB \mid AA \mid \varepsilon$
$S \rightarrow AA_1 \mid AS \mid SA \mid BB \mid AA$
$A \rightarrow BB$
$A_1 \rightarrow SA$
$B \rightarrow a$

(1 mark)

Q1. (b) [3 Marks].



The PDA should accept empty string. (0.5 marks)

The PDA should accept just 0 and 1. (0.5 marks)

The PDA should accept strings containing all 0's and all 1's. (0.5 marks)

The PDA should accept the remaining strings in the given language. (1.5 marks)

[8M]

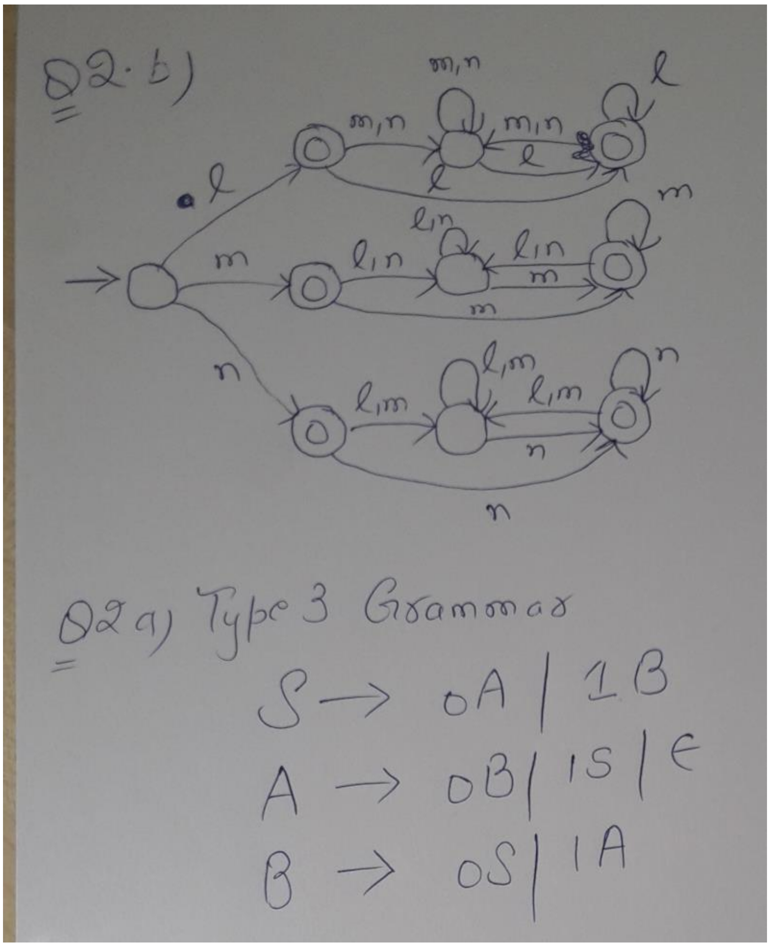
Q2. Answer the following two parts together:

a) Design a Type-3 grammar (with rules as per Chomsky hierarchy of Type-3) generating the following language P over $\Sigma = \{0,1\}$

$$P = \{ x \in \Sigma^* \mid (\#0(x) - \#1(x)) \bmod 3 = 1 \}$$

b) Construct a minimal DFA which recognizes the following language P:

P = { x ∈ {l, m, n}* | x starts and ends with the same symbol.}



Marking Scheme:

Q2a (5 Marks) Zero marks is given if someone has written the grammar in Type 2 form. Else, full 5 marks .

Q2b (0/5 Marks).

[10M]

Q3. Answer the following three parts together:

a) **[3M]** Consider P and Q are two languages over the similar ∑ such that P and P.Q are type-3 languages. Show or contradict the following: Q is also Type-3 language.
False. It is not necessary. For example, If P = a* (or phi), Q is a^n b^n, then P.Q is a*b* (or phi).
Marking: 1M for writing FALSE. 2M for justification.

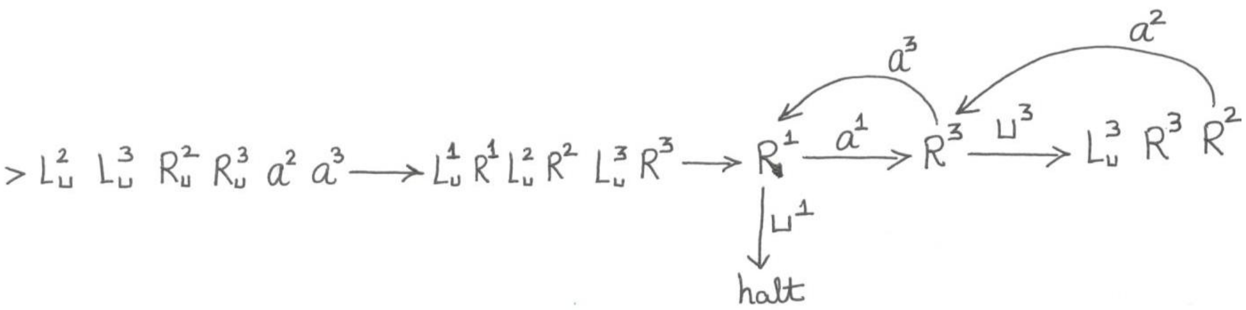
b) **[6M]** Prove that the following language L is decidable (by giving a description of TM which decides it):
L = {“M” | M is a Deterministic Finite Automaton and there exists a string w which M accepts such that #a(w) > #b(w).}
[Hint: Think about theorems proved in class.]
Sol: Let L1 = {x | x has more a’s than b’s}. We know that L1 is a CFL and we can construct a PDA for L1. Turing Machine M1 to decide L is as follows:
M1 on input “M”, where “M” is a DFA:

a. Construct L2 = L1 ∩ L(M). We know that language L2 is a CFL. We also know that testing whether the language of a CFL is ϕ or not is decidable.

b. If language of L2 is ϕ, then M1 halts in reject state (i.e. we can conclude that the DFA M does not accept any string with number of a's greater that number of b's). Rather, if language of L2 ≠ ϕ, then M1 halts in accept state.

Marking Scheme: 0/6.

c) **[6M]** Over ∑ = {a}, consider the task of designing a 3-Tape Turing Machine M which does the following computation. The initial configuration of Tape-1 is ⊳ <u>⊔</u> **x**, where x ∈ ∑*, |x| = k, and k > 0. The initial configuration of Tape-2 and Tape-3 is ⊳ <u>⊔</u>, and ⊳ <u>⊔</u> respectively. Final configuration of Tape-2 is ⊳ <u>⊔</u> y, y ∈ ∑*, and |y| = ceil (square-root(k)).
The following TM claims to do the desired computation. Check it, and correct the mistake (if any).



Sol: Do the following modifications:

a) From last R^2, the transition of a^2 should go to R^1 (and not R^3). Also, from last R^2, transition should go to first [move left until a blank is found] on reading blank on tape-2.

Marking Scheme: 3M for each of two modifications

[15M]