74+4
100

This entrance exam booklet is being re-used for mid-sem exams.
Please ignore the printed material on the sheets.

CHENNAI MATHEMATICAL INSTITUTE
https://www.cmi.ac.in

Mid Semester Examination, Aug–Nov 2023

Name: Andrā Mazumder	Roll Number: MCS202304
Date:	Subject: TOC
Course & Year: MSc CS1 2023	Total No. of Pages:

	Points	Remarks
Part A		
Part B		
Total		

	Points	Remarks
B1		
B2		
B3		
B4		
B5		
B6		
Total		

Answers to part B of 2023 Entrance Examination for BSc Programmes at CM

Carefully read the instructions on the question paper.

B1. Write your solution to B1 below.

(1) ~~You have to~~ L' is regular.

we will prove this using a DFA M' that will accept L'

$M' = \{Q', \Sigma', \delta', q_0', F'\}$ where $M = \{Q, \Sigma, \delta, q_0, F\}$ and $L(M) = L$

Now, $Q' = Q \cup \{T\}$ trap state

$\Sigma' = \Sigma$

$\delta'(q, a) =$ if $q \in Q$ and ~~$q \in F$~~ $q \notin F$ then $\delta(q, a)$ else T

$q_0 = q_0$

$F' = F$

10

Basically we will remove all the transitions from final state so that they can't reach any other final state

Proof: $L' \subseteq L(M')$

~~Contradiction~~

If $w \in L'$ w can not be divided into zy s.t. $z \in L'$.

So, as it hasn't seen any final state till now, it will be accepted by M' as it only matches transition after final state

$L(M') \subseteq L'$

Let $w \in L(M')$

as $w \in L(M')$ no way it can go to another state as every transition from there is to trap state

So, no other words will have prefix as it.

2

Solution to B1 continued

we will prove this using Myhill-Nerode theorem.

Let take two general words of the form ab^i and ab^j and $i \neq j$

In this case, if $i \neq j$

then adding c^i to the end of ab^i will make it accepted

while " c^i to the " of ab^j will not

So, for $\forall i, j$ pair.

ab^i and ab^j exists in different equivalence classes.

10+5

So, this language has infinite index. So, it's not regular

But, we have to use pumping lemma after proving it ?

B2. Write your solution to B2 below.

~~Now I want to pick a string~~

I want to prove that language L not regular.

~~But~~, Adversary will say that it is regular and he/she will design a p state DFA for me

$M = (Q, \Sigma, \delta, q_0, F)$

Now, as L is regular

every string $\in L$ will be accepted by that DFA.

So, $a b^p c^p$ will be accepted by that ~~Language~~ DFA

So, say before reading P, we are at state

$q \in Q$.

But after reading all the bs I have passed through

$p+1$ states. including q.

So, by Pigeonhole principle as there are p states only in the DFA. there must be a state which I have passed two times

Say, q' be such a state, and

$$\delta(q', b^k) = q'$$

and $k \leq p$

So, I will once again go through the loop and then normally read c^p.

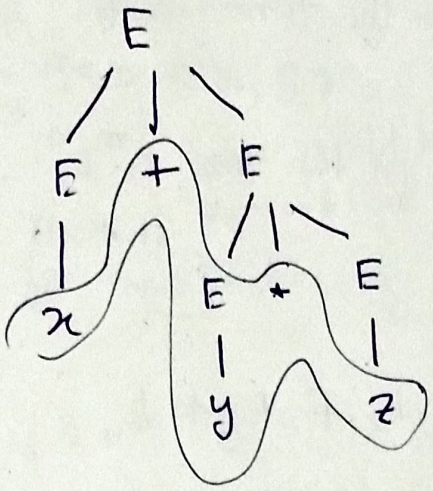
But, then $a b^{p+k} c^p$ is also accepted by the ~~string~~ DFA, Hence adversary was wrong. So No DFA exists.

(4) The grammar shown here is ambiguous

So, there will be two different leftmost parse trees for some strings

Let $w = x + y * z$

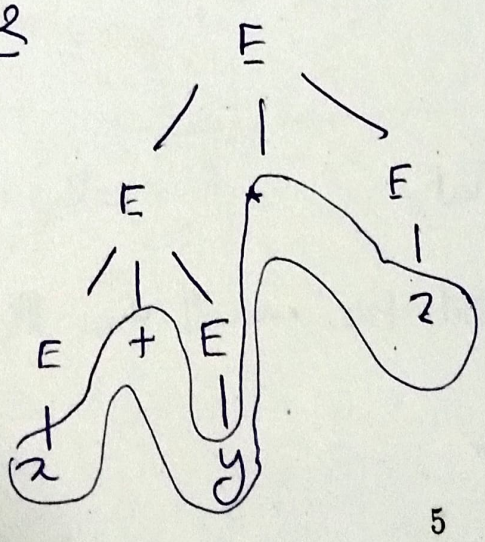
Parse tree 1



= x+y*z

10

Parse tree 2



= x+y*z

B3. Write your solution to B3 below.

⑤ $\bar{L}$ complement will be union and concatenation of several languages

we will pick them carefully.

let L_1 = set of all strings such that it starts with b. $-\{\varepsilon\}$

let L_2 = set of all strings start with a $-\{\varepsilon\}$

L_3 = set of all strings in the form $a^n b^m$ st $m > n$.

L_4 = set of all strings in the form $a^m b^n$ st $m > n$

So, $\bar{L} = L_1 + L_3 \cdot L_2 + L_4 \cdot L_2 + L_3 + L_4$

(•) Concatinate

(+) union

Why? An argument is required.

Now we have to show that,

each of them is context-free. we will show that by giving grammars for them

Solution to B3 continued

$L_1 \Rightarrow$ $S \to b \mid sa \mid sb$

$L_2 \Rightarrow$ $S \to a \mid sb \mid sa$

$L_3 \Rightarrow$ will be concatenation of $a^n b^{nH}$ with b^+

$$L_3 \to \left\{ \begin{array}{l} S \to S_1 S_2 \\ S_1 \to a S_1 b \mid b \\ S_2 \to \epsilon \mid b S_2 \end{array} \right\}$$

Similarly,

$L_4 \Rightarrow$ will be concatenation of a^+ with $a^{nH} b^n$.

$L_4 \Rightarrow$ $S \to S_3 S_4$

$S_3 \to a S_3 \mid \epsilon$

$S_4 \to a S_4 b \mid a$

So, it is regular ?

8

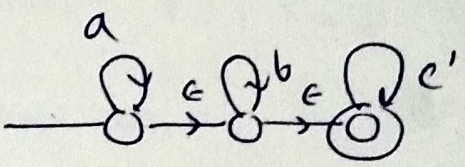
7). we know from the closure property of CFL

Intersection of CFL with regular language is CFL

now, $L = \{ w \mid w \text{ has equal nos of } a,b,c \}$

let, L_R be a regular language

$L_R = \{ a^* b^* c^* \}$



So, $L_N = L_R \cap L$

$= \{ a^n b^n c^n \mid n \geq 0 \}$

Claim:

now, we will prove that. L_n is not context free. using Pumpi

Lemma

let p be the pumping length

So, $z = a^p b^p c^p$

$= uvwxy$

such that, $|vwx| \leq p$

and $|vx| \geq 1$

B5. Write your solution to B5 below.

- If vwx is contained inside a^p, b^p or c^p

we are done. on choosing any $i \neq 1$, will give a string not in L.

Subclaim: vwx cannot span over all $a^p b^p c^p$

Proof:
as $|vwx| \leq p$. and $|b^p|$ has length p

we can't do that

So, in all the cases, ~~we can't change~~ we can always change the length of $|a^p|$ and $|b^p|$ but not $|c^p|$. WLOG

So, not context free. claim Proved

10

Now. CFL ∩ Regular ⇒ CFL

But as the resulting language is not CFL,

we can ~~always~~ say that either the first language is not CFL or 2nd one not regular. 10 But we proved 2nd one is regular.

So, The given Language is not context free

B6. Write your solution to B6 below

6 $S \to \epsilon \mid bA \mid aB$

$\{ A \to aS \mid bAA \mid a \}$

$\{ B \to bS \mid aBB \mid b \}$

we will take these production and remove S from them

$S \to \epsilon \mid bA \mid aB$

$\{ A \to a \mid abA \mid aaB \mid bAA \mid a \}$ one of terminal in the greibach Normal form

$\{ B \to b \mid bbA \mid baB \mid aBB \mid b \}$

If $A \overset{*}{\Rightarrow} \alpha$ then α has ...

Now

Claim { A has $\#A + \#a - \#B - \#b = 1$

B has $\#B + \#b - \#a - \#A = 1$ → length of derivation

we will prove this using induction on no of derivations of a sentence

Base case $A \to a \mid aBA \mid aaB \mid bAA \mid a$

$B \to b \mid bbA \mid baB \mid aBB \mid b$

Base case true

Now Induction hypothesis

$B \to b \mid bbA \mid baB \mid aBB$

$A \to a \mid abA \mid aaB \mid bAA$

both A and B produce strings of length k

But we know that A and B can be derived in k+1

Solution to B6 continued.

(k+1) total step

We have to show that after one more step it will still be satisfied

	$\#(A)-\#(B) + \#a - \#b = R$
$B \to b$	-1
$B \to bbA$	-1
$B \to bAB$	-1
$B \to aBB$	-1

Similarly for, A will give one accen of R.

Hence proved.

8

Now, $S \to \epsilon \mid bA \mid aB$.

Case 1: $S \to \epsilon$

and it has equal no of a's and b's = 0

Case 2: $S \to bA$

for b / for A

So, $\#a + \#A - \#b - \#B = 1 + -1$

$= 0$

If necessary, continue solution to any problem on blank pages →

But we know, whatever S produces ultimately should

ly terminal string

So, $\#A = \#B = 0$

Blank page for rough work/continuation of part B solutions

8) We will create a DFA using product construction and keeping an extra counter to store what language we are now reading letter from.

$M'' = \{Q'', \Sigma'', \delta'', q_0'', F''\}$

Let $L(M') = L'$

$L(M) = L$

So, $Q'' = Q \times Q' \times \{0, 1\} \cup T$

$\Sigma'' = \Sigma \cup \Sigma'$

$\delta''(\langle q_1, q_2', k \rangle, a) =$ if $k = 0$, $\langle \delta(q_1, a), q_2', 1-k \rangle$

$a \in (\Sigma \cup \Sigma')$ else $\langle q_1, \delta'(q_2', a), 1-k \rangle$

$q_0'' = \{q_0, q_0', 0\}$

$F'' = \{\langle q_i, q_j', k \rangle \mid q_i \in F \text{ and } q_j \in F'\}$ (k = 0 or 1) ✗ has to be on (q, q', 0) states.

5

$\delta''(T, \Sigma) = T$

if $\delta(q_1, a)$ or $\delta'(q_2', a)$ is not defined (like $a \notin \Sigma$) then we will send them to trap state ?

15

If some case is missed to be handled, we can create an NFA to handle many of them automatically

8(a) ?

Blank page for rough work/continuation of part B solutions

8. Both L and L' are regular

So, we can construct a DFA for it

Before that we will use homomorphism

$$\begin{Bmatrix} h(a) = a \\ h(b) = \epsilon \end{Bmatrix}$$

So, as Regular languages are closed under homomorphism

both L_H and L'_H are regular

What is L_H? L'_H?

So, L_H only consists of w such that $w \in a^*$

$$L_H = \{ w \mid \text{if } w' = b^* a b^* \quad b^* \text{ such that } h(w') = w \in L \}$$

What is w' ? quantifier?

Same goes for L'_H

So, we can create DFA for them

Then, the problem is now reduced to produc construction to check if in both the words $\in L_H$ and L has same number of as.

4

$L'' = L(M'')$ will be such language.

$M'' = \{ Q \times Q', \{a\}, \delta'', q_0'', F'' \}$

$q_0'' = \{ q_0, q_0' \}$

$F'' = \{ q_i, q_j' \mid q_i \in F \text{ and } q_j' \in F' \}$

$\delta''(\langle q_i, q_j' \rangle, a) = \langle \delta(q_i, a), \delta'(q_j, j) \rangle$

Now we have to use inverse homomorphism to construct the actual language.

Now, we have created a DFA that accepts only those a^* which has a same number of a as in w such that $w \in L$.

So, we will again use product construction to accept only those strings such that the a^* with number of as ... is accepted by L''.

Blank page for rough work/continuation of part B solutions

$Q''' = (Q'' \times Q)$

Transition $\delta'''(\langle q_i'', q_1 \rangle, a)$

$= \langle \delta''(q_i'', a), \delta(q_1, a) \rangle$

$\delta'''(\langle q_i'', q_1 \rangle, b)$

$= \langle q_i'', \delta(q_1, b) \rangle$

Final state $= F_1 \wedge F_2$

Start state $= \langle q_0'', q_{10} \rangle$

4

Reexamine

This is wrong! later it is done

Redundant

Blank page for rough work/continuation of part B solutions

9. (a) Let $n = 2^k$ where k is the number of Variables in the Grammar.

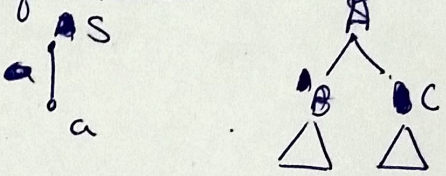
So, we can convert every grammar to corresponding CFL in Chomsky Normal Form.

where, $\begin{cases} A \to BC \\ A \to a \end{cases}$ all productions are in following form.

Derivation tree of

Claim: string with greater than $\geq 2^{K}$ length has path length atle $(K+1)$

Induction on no of derivations.



Base case: Its true. as $K=0$, $2^K = 1$

String length is 1

and path length is 1

Inductive hypothesis: Its true for derivation trees of production of $(A \to BC)$.

So, Both B and C has string length $\geq 2^{K-1}$

and they have path length $\geq K$

Redundant

Blank page for rough work/continuation of part B solutions

Induction step:

Now string length is $\geq 2^k + 2^k$

$= 2^{k+1}$

and path length is $\max(\geq k+1, \geq k+1) + 1$

$= k+2$

Hence proved

Let us take any string that has length at least $n \in 2^k$?

So, the derivation tree must have path length $(k+1)$.

So, number of vertices is atleast $k+2$.

we know that, the leaf of tree is a terminal

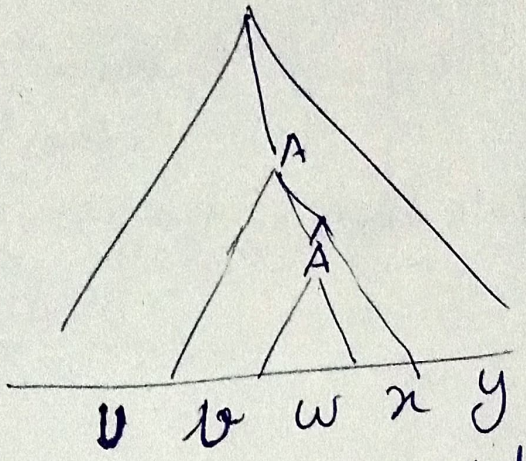
So, no of variables in the longest path is $k+1$

But k is the number of variables in the grammar

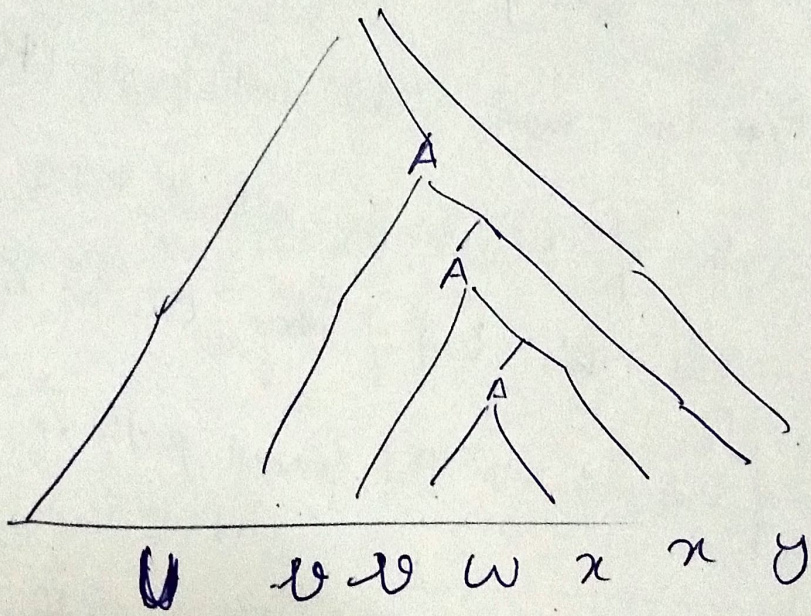
So, there must be a variable which is repeated

Redundant

Blank page for rough work/continuation of part B solutions



if this is the case, a similar valid tree will be



$\therefore uv^2wx^2y \in L$.

or remove it

we can increasingly add A into tree A, so

$uv^iwx^iz \in L$ for all i.

Blank page for rough work/continuation of part B solutions

Let G' be the grammar after removing unit productions

(a) (a)

Let m be defined as the max length of a production and K be the number of variable

if $A \to \alpha B \beta$

then $m = \max |\alpha\beta|$ such that $A \to \alpha B \beta$ is a production (all the prods)

claim: Strings with length $\geq n = mK$ has path length $\geq K$

Induction on derivation steps

Base case: $l \leq m$

S
|
a

Induction hypothesis.

S
|
α B β

B has length and K−1 steps $> m(K-1)$

Induction:

So, now. $|\alpha\beta| \geq m$

$\geq mK$

So, has path length 23

and K steps is needed

So, let $n > mk$ and such a string exists with length(n)

So, there must be $(k+1)$ steps in deriving that string

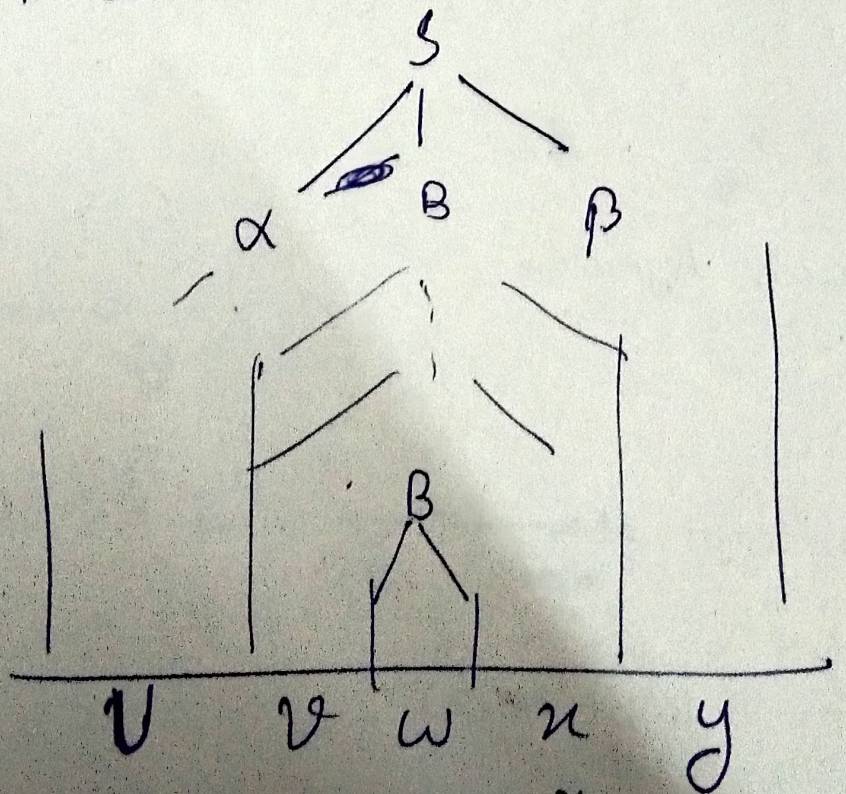
But, path length of $(k+1)$ has vertices count $(k+2)$

and the leaf is a terminal.

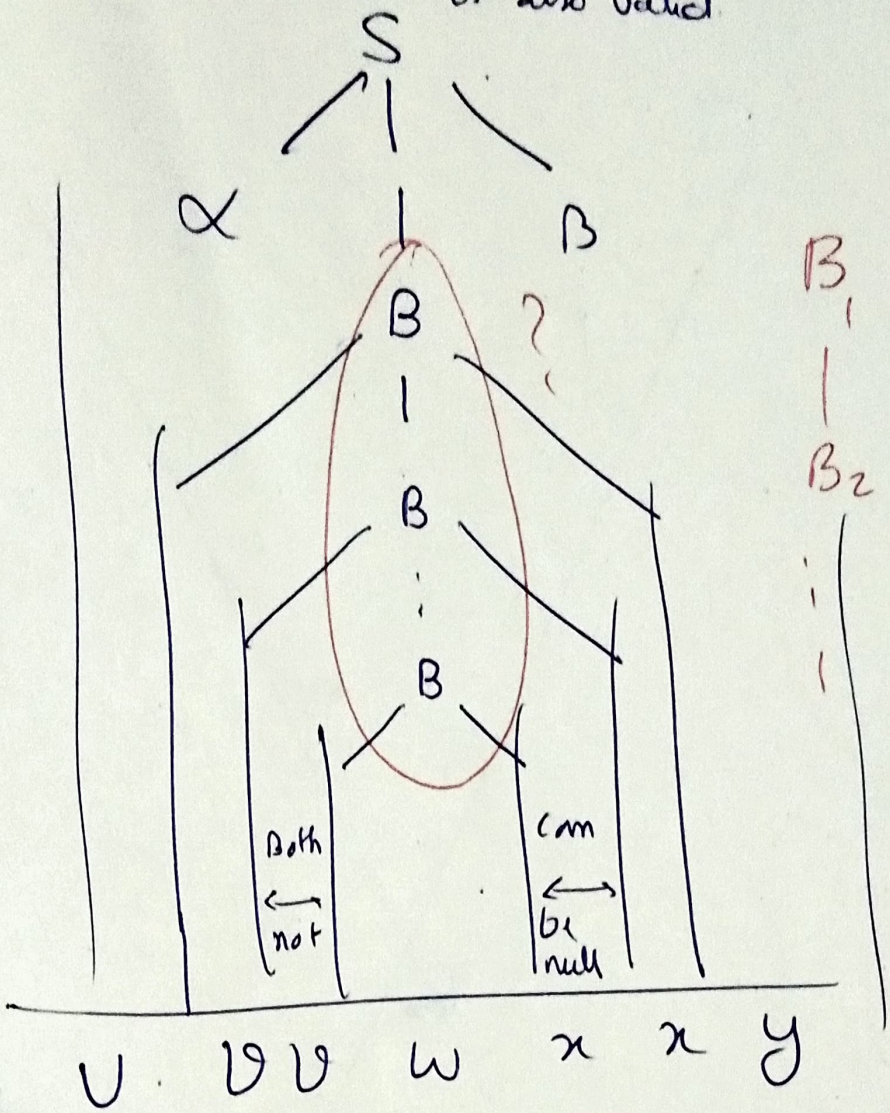
So, $(k+1)$ variable must exists in that path

and there will be ~~atleast~~ one variable ~~that~~ is repeated

So, the derivation tree for s



But this derivation tree is also valid



8

∴ We can infinitely increase B or completely remove it

$$uv^i wx^i y \in L$$

we know that, Both α, β of (B → αCB) can not be null as we have removed unit production

So, $|\alpha\beta| \geq 1$

∴ So, $|vx| \geq 1$

But we also know,

$|\alpha\beta|$ is every step is almost m.

So, sum of all such $|\alpha\beta|$ will be mk

and $n > mk$

So, $|uvxy| \leq k|\alpha\beta|$

$\leq mk$

$\leq n.$

So, Both the condition are satisfied.

Why is this written here?