

Ques - $\{a^n b^n c^{2n} \mid n \geq 0\}$ is a CFL?

↑
CFL $\cap$ regular = Yes it is a CFL.

Never do merge the language, try to go for closure.

Ques - $DCFL \uparrow \cup CFL \Rightarrow$ so push smaller one
 $CFL \cup CFL = CFL$

Ques $DCFL \cdot reg \uparrow$

$DCFL \cdot DCFL = X$

↓
 $DCFL \uparrow \cdot DCFL \uparrow = CFL \cdot CFL = \checkmark$

Try to give the closest Answer

Ques $(a^n b^n c^{2n}) \cap (a^* b (a+b)^* b a^*) = CFL \cap reg = CFL$

(a) CFL

(b) DCFL

(c) regular

(d) CSL

↓
DCFL

Ques - $L = \{a^n b^n\}$ $R = \Sigma^*$

$L \cap L \cup R = ?$ $L \cup R \Rightarrow CFL$

$L \Rightarrow CFL$ $CFL \cap reg$

but $\{a^n b^n\} \cup \Sigma^*$ is Σ^*

$L \cup R$

$L \cap R$

↓ regular

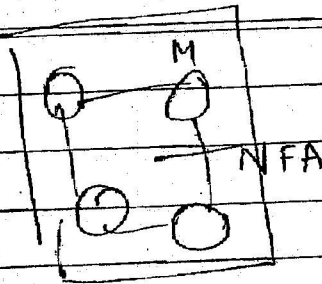
$\therefore$ regular & non-regular

q. - Difference is not commutative.

Page No.

Date : / /

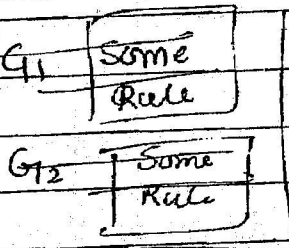
Ques



$\overline{L(M)}$ is regular or not?

Yes, it is regular.

Ques



Ques

$\overline{RE} = ?$

May not may not be RE

Ques

$L_1 - DCFL$ $L_2 - CFL$

$$L_1 - L_2 = L_1 \cap \overline{L_2}$$

$$DCFL \cap CFL \uparrow$$

$$= DCFL \cap CSL$$

$$= CSL$$

$L_2 - L_1$

$$= \overline{L_2} \cap L_1$$

$$= CFL \cap DCFL$$

$$= CFL \cap CFL \uparrow$$

$$= \underline{CSL}$$

Ques

$$L_1 - L_2 = L_1 \cap \overline{L_2}$$

$$\text{XOR } L_1 \oplus L_2 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$$

$$\text{NAND } L_1 \uparrow L_2 = L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

$$\text{XNOR } L_1 \downarrow L_2 = \overline{L_1 \cap L_2} = \overline{L_1} \cup \overline{L_2}$$

$$L_1 \Rightarrow L_2$$

$$\text{XNOR } L_1 \ominus L_2 = (L_1 \cap L_2) \cup (\overline{L_1} \cap \overline{L_2})$$

Ques

$$L - \text{Reg} = L$$

$$\text{Reg} - L = \text{may not be } L$$

$$L = CFL$$

$$\text{Reg} - CFL = \text{Reg} \cap \overline{CFL}$$

$$= \text{Reg} \cap CSL$$

$$= \underline{CSL}$$

EXCELLENT

Ques $L_1 = \text{DCFL}$

$L_2 = \text{Reg}$

$$\begin{aligned} L_1 \oplus L_2 &= (L_1 \cap \bar{L}_2) \cup (\bar{L}_1 \cap L_2) \\ &= (\text{DCFL} \cap \text{Reg}) \cup (\text{Reg} \cap \text{DCFL}) \\ &\quad \text{CFL} \quad \cup \quad \text{CFL} \\ &= \text{CFL} \end{aligned}$$

~~Ques~~ ~~Ques~~ ~~Ques~~

precedence - $() \geq \text{'complement'} \geq \cap \geq \cup \geq \Rightarrow \geq \Leftrightarrow$

Ques - $L_1 = \text{DCFL}$

$L_2 \rightarrow \text{DCFL}$

$(L_1 \cap L_2) =$ first $\cap$ then comp^{united} will generate weak answer so use de Morgan law

$$\begin{aligned} \overline{(L_1 \cap L_2)} &= \bar{L}_1 \cup \bar{L}_2 \\ &\quad \text{DCFL} \cup \text{CSL} \\ &= \text{CSL} \end{aligned}$$

- For a language to be closed under $\oplus, -, \uparrow, \downarrow, \Rightarrow, \Leftrightarrow$, it has to be closed under $\cup, \cap, \neg$ complement.

$\cap$ ^{can} also be converted to $\cup$ by de Morgan law

$\therefore$ So

for a language to be closed under $\oplus, -, \uparrow, \downarrow, \Rightarrow, \Leftrightarrow$, it must be closed under $\cup$, complement. ^{secondary opⁿ}

- Reg, CSL, Rec are closed on secondary opⁿ.

● DCFL - Union - May or may not be DCFL

Ex - $L_1 = \{a^n b^n c^m \mid m, n \geq 0\}$ - DCFL

~~$L_2 = \{a\}$~~

~~$L_2 = \{a b^n c^n\}$~~

● A CFL is DCFL if

1. the push to pop has to be cleared i.e. no confusion i.e. when to push when to pop

2. How much the push has to be cleared. ^{or how much} pop it is also ^{also} ^{becl}

3. $\{a^m b^n c^p \mid m=n \text{ or } m=p\}$ - For DCFL, language must not have it.

Ex - $\{w w^R \mid w \in \{0,1\}^*\}$ - It is a CFL not a DCFL as push: pop is not cleared

as 101101

how system will know when to push & when to pop
but NDPA will do it by splitting of machine & storing each & every state

● Union - Not closed

$L_1 = \{b^n c^n \mid n \geq 0\}$ DPDA

$L_2 = \{b^n c^{2n} \mid n \geq 0\}$ DPDA

$L_1 \cup L_2 = b^n c^n \text{ or } b^n c^{2n}$ Not a DPDA or DCFL

as it is confused how much to push or much to pop

bbcc

bbcccc

● double comp. on a single variable is allowed with 'OR' cond. in CFL or PDA

$\therefore \{a^m b^n c^p \mid m=n \text{ or } m=p\}$

↓ CFL

but not DCFL

EXCELLENT

$\{w \alpha w^R \mid w \in (0,1)^*, \alpha \in (0,1)\}$ - CFL but not DCFL

ques

I - $\{w \alpha w^R \mid w \in (0,1)^*, \alpha \in (0,1)\}$

II $\{w \alpha w^R \mid w \in (0,1)^* \alpha \in (0,1)^*\}$



Which one is DCFL

Ans - II as each RL is DCFL

Intersection - not closed

$L_1 = \{a^n b^n c^m\}$

$L_2 = \{a^n b^m c^m\}$

$\rightarrow$ CFL
~~DCFL~~ DCFL

$L_1 \cap L_2 = \{a^n b^n c^n\}$ - not a DCFL

Ques

$\{a^n b^n c^n\} \cup \{d b^n c^{2n}\} \rightarrow$ DCFL

as no confusion

when 'd' came push 1 'd'

when 'd' came push 2, 'd'

$\{c^n b^n a\} \cup \{c^{2n} b^n d\} =$ Not a DCFL

CFL - Intersection - not closed

$L_1 = \{a^n b^n c^m\}$

$L_2 = \{a^n b^m c^m\}$

$L_1 \cap L_2 = \{a^n b^n c^n\}$ - CSL

for Regular language-

Union - $L_1 \rightarrow \text{regular} \rightarrow \exists a_1 e_1$

$L_2 \rightarrow \text{regular} \rightarrow \exists a_2 e_2$

$\therefore L_1 \cup L_2 = a_1 e_1 + a_2 e_2 = \underline{a_1 e_1} \therefore$

$L_1 \cup L_2 = a_1 e_1 \therefore$ it is regular

• Kleen theorem for Generation Machine & Grammar guarantee
Right linear regular grammar.

$L_1 \rightarrow a_1 e_1$

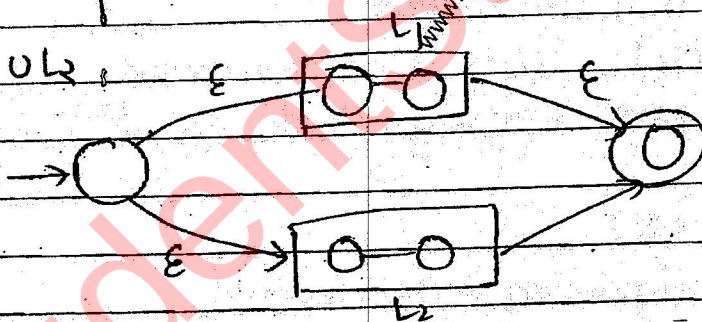
$L_2 \rightarrow a_2 e_2$

$L_1 \cup L_2 \rightarrow$ draw Machine using Theorem

$\therefore$ it is also generate grammar which is RLR or.

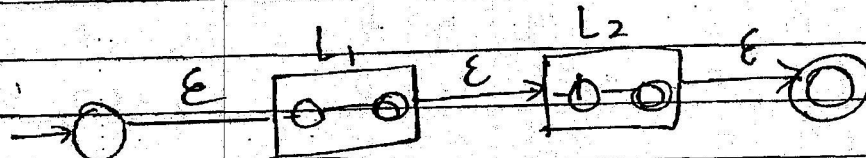
$\therefore$ it is regular.

for $L_1 \cup L_2$,

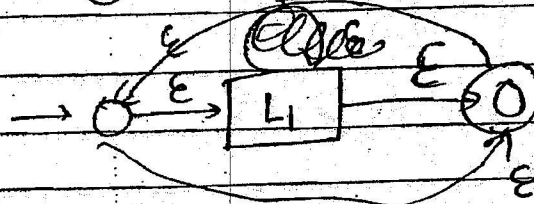


$L_1 \cap L_2$

$L_1 \cap L_2$



L_1^*



for complement - for each r.l we can draw a fa
 each fa is ~~can~~ can be converted to DFA
 & we can ~~comple~~ obtain complement of DFA.
 $\therefore$ it is closed.

for Intersection- if L_1 is regular & L_2 is also regular
 $\therefore$ i.e. M_1, M_2 corresponds to L_1, L_2
 Machine

$$\text{then } L(M_1 \times M_2) = L(M_1) \cap L(M_2)$$

↑ product Automata

$M_1 \times M_2$ can be obtained from DFA's only, on same alphabet set

Product Automata- if $M_1 - m$ state
 $M_2 - n$ state

$M_1 \times M_2$ $m \times n$ state

$$L(M_1 \times M_2) = L(M_2 \times M_1) \quad // \text{ it is commutative.}$$

they will accept same language but M/c ~~may~~ will be different.

$$\therefore L(M_1 \times M_2) = L(M_2 \times M_1)$$

but $M_1 \times M_2$ configuration $\neq M_2 \times M_1$ confi.

Ex -

M_1	0	1	M_2	0	1
$\rightarrow A$	B	A	$\rightarrow C$	D	D
(B)	A	B	(D)	C	D

$M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$
 $M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

$$M_1 \times M_2 = (Q, \Sigma, \delta, q_0, F)$$

$$Q = Q_1 \times Q_2$$

$$\Sigma = \Sigma_1 = \Sigma_2$$

$$q_0 = \{q_{01}, q_{02}\}$$

$$F = \{F_1 \times F_2\}$$

↓ it will accept all those state

EXCELLENT

- $F_1 \times F_2$ will allow the o/p which is common to both & this is the property of intersection.

Page No.

Date : / /

$$\Phi = \{ \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\} \}$$

$$\delta((X, Y), a) = (\delta_1(X, a), \delta_2(Y, a)) \quad L_1 \cap L_2$$

where

$$(X, Y) \in \Phi$$

$$X \in \Phi_1$$

$$a \in \Sigma$$

$$Y \in \Phi_2$$

$$F \text{ for } L_1 - L_2 = \{BD\}$$

X Y $M_1 \times M_2$	0	1
$\rightarrow AC$	BD	AD
AD	BC	AD
<u>BC</u>	AD	BD
BD	AC	BD

for $L_1 \cup L_2$

$$F = \{AC, BC, BD\}$$

$$\text{for } L_1 \oplus L_2 = \{AC, BD\}$$

for its conversion for requirement of answer

if given

$\rightarrow A$	A	B
<u>B</u>	B	C
C	C	D
D	D	A

initially take initial state & final state & check whether it matches

$$A \rightarrow AC$$

$$B \rightarrow BC$$

you can check it intuitively

A	
-	B
B	
-	A

• $L_1 \times L_2$ can be used to create any secondary op machine

1. If $L_1 \cup L_2$ is required - construct $L_1 \times L_2$

$$\text{and } F = \{(X, Y) \text{ where } X \in F_1 \text{ or } Y \in F_2\}$$

2. if $L_1 - L_2$ is required - construct $L_1 - L_2$

$$\Rightarrow F = \{F_1 \times (\text{non final state of } L_2)\}$$

$$F = \{F_1 \times \bar{F}_2\}$$

3. $L_1 \oplus L_2$ - $F = \{X \in F_1 \text{ or } Y \in F_2 \text{ but } X, Y \notin F_1, F_2\}$

4. ~~L1 - L2~~

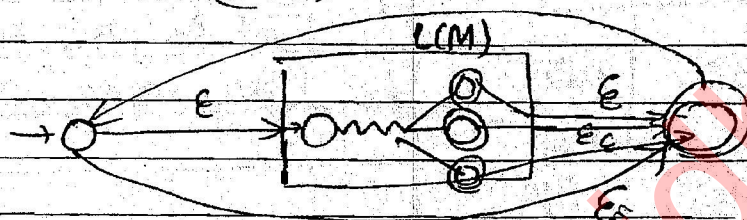
EXCELLENT

It is possible to have a Grammar which is not regular but language generated is regular.

Concatenation - $L_1 \rightarrow a^*e$
 $L_2 \rightarrow a^*e$
 $L_1 L_2 \Rightarrow a^*e$

closure - $L - RL$
 $L \rightarrow a^*e$
 $L^* = (a^*e)^*$ is also regular

To convert $L(M)$ to $(L(M))^*$



Reversal =

$S \rightarrow AB$
 $A \rightarrow aA \mid \epsilon$
 $B \rightarrow bB \mid \epsilon$

Non regular grammar generate RL

$L = a^*b^*$ bcoz CFL can also generate R.E

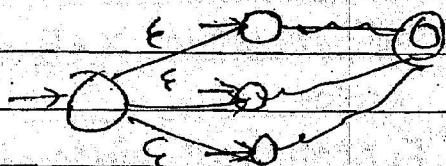
• Type 3 cat

• but RG can not be written for Non-Regular language.

Reversal - 1. Exchange the final state & initial state

2. Reverse all the arrows

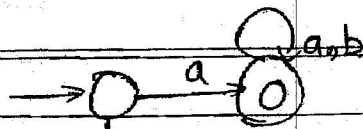
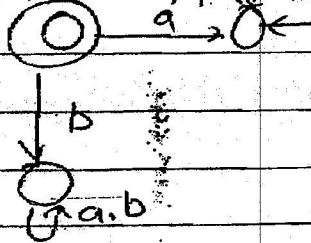
3. In case you have more than initial state convert into NFA by new initial state



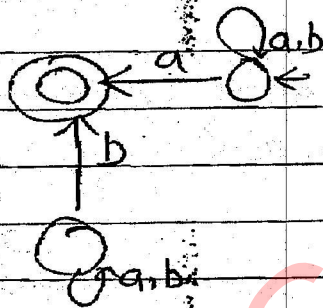
• this algo may or may not give minimized Mach.

EXCELLENT

Ex -

 $a(a+b)^*$ 1) exchange if & f.f. $Q_{a,b}$ 

2) Reverse arrow:

 $(a+b)^*a$

Homomorphism of Language - if L is language on Σ
then $h(L)$ will be L' on Σ'

$$\Sigma' = \{a, b\}$$

as $L = \{01, 100\}$ on $\Sigma = \{0, 1\}$

$$\Sigma \rightarrow \Sigma^*$$

$$\{\Sigma \rightarrow \Sigma^*\}$$

Let $h(0) = a$ $h(1) = b$ $\therefore h(L) = \{a^2b, ba^2a^2\}$

- It is closed as L is RE $\therefore$ and $h(L)$ is also RE

- $h(L)$ is of finite L is always L .

- $h(L)$ of Infinite L may be finite or infinite

 $h(L)$ is many to one mapping.

$$L = (01+100)^*100$$

$$h(L) = \epsilon$$

$$h(0) = \epsilon$$

$$h(1) = \epsilon$$

EXCELLENT

$$\Sigma = \{a, b, c\}$$

$$\Gamma = \{0, 1\}$$

$$h(a) = 1$$

$$h(b) = 0$$

$$h(c) = 01$$

$$h(L) = \{1001\}$$

$\therefore h^{-1}(L) =$ each string which can form this language on Σ
 $\therefore \{abba, abc\}$

$$\text{If } h(L) = \{1010\} \quad h(a)=1, h(b)=0, h(c)=10 \\ = \{abab, abc, cab, cc\}$$

- cardinality of $h(L)$ can be more than or less than or equal to L as it only depends on Mapping $h(a)$
 $\therefore (h(L) \leq |L|)$

Converse of the closure properties-

Union-

- If $L_1 \cup L_2$ is regular then it may or may not be possible that L_1, L_2 are not regular.
- It is only 1 way regular theorem

$$L_1 \text{ is reg \& } L_2 \text{ is reg} \Rightarrow L_1 \cup L_2 \text{ is reg.}$$

as if

$$L_1 = a^n b^n \rightarrow \text{Non Reg.}$$

$$L_2 = (a+b)^* \rightarrow \text{Reg}$$

$$L_1 \cup L_2 = (a+b)^* = \Sigma^*$$

and $L_1 = a^n b^n$

$$L_2 = \overline{a^n b^n}$$

$$L_1 \cup L_2 = a^n b^n \cup \overline{a^n b^n} \\ = (a+b)^*$$

• Complement & Reversal of an ^{Regular} language is regular

• It is two way theorem

∴ If L is regular

then $\bar{L}$ is also regular

& if L is Non Regular then $\bar{L}$ is also Non regular.

Intersection - if $L_1 \cap L_2$ is regular, then L_1, L_2 may or may not be regular

ex-

$$\begin{array}{ccc}
 \emptyset \cap a^n b^n = \emptyset \\
 L_1 \quad L_2 \quad L_1 \cap L_2 \\
 \uparrow \quad \quad \uparrow \\
 \text{regular} \quad \text{Non reg}
 \end{array}$$

$$\begin{array}{ccc}
 \overline{a^n b^n} \cap a^n b^n = \emptyset \\
 L_1 \quad L_2 \quad L_1 \cap L_2 \\
 \underbrace{\quad \quad \quad}_{\text{JNR}} \quad \quad \quad \uparrow \\
 \quad \quad \quad \text{IR}
 \end{array}$$

ques - if $L_1 \cup L_2$ is Non regular

• both are NR

• 1 is NR

• atleast 1 is NR

ques - If $L_1 \cup L_2$ is Non regular & L_2 is reg

$L_1 = ?$

= Must be non regular

for Complement-

L is reg. $\Leftrightarrow \bar{L}$ is regular

i.e. if L is reg. then $\bar{L}$ is regular

if $\bar{L}$ is reg. then L is regular

if L is NR then $\bar{L}$ is also non Regular

if $\bar{L}$ is NR then also L is regular

if L & $\bar{L}$ are two language which is not possible-

(a) both are DCFL's

(b) both are regular

(c) both are RE

→ because if one is RE if it is REC as REC is also RE

✓ but one is REC, other is Not RE

$\left\{ \begin{array}{l} L = \text{rec (also reg)} \\ \bar{L} = \text{rec} \end{array} \right\} \rightarrow \text{possible}$

for concatenation-

• L_1 is regular & L_2 is reg. $\Rightarrow L_1 L_2$ is regular

• if $L_1 L_2$ is regular then atleast one has to be regular.

$$\emptyset \cdot a^n b^n = \emptyset \quad (L_1 \cdot L_2)$$

$L_1 \quad L_2 \quad \text{non reg}$

Then regular

• if $L_1 L_2$ is not regular atleast one has to non Regular.

Ques

$L_1 L_2 = \text{regular} \quad L_1 = \text{regular}$

$L_2 = \text{may or may not be regular}$

for * L is reg $\Rightarrow L^*$ is regular• if L^* is regular, L may or may not be regular

$$\text{if } L^* = \{a^n b^n + a + b\}^* = (a+b)^*$$

$$\left\{ \begin{array}{l} L = \{a^n b^n + a + b\} \\ \downarrow \text{non regular} \end{array} \right.$$

$$\text{if } L = \{a^{n^2} \mid n \geq 0\}$$

$$\begin{aligned} L^* &\text{ is } \{\epsilon + a + aa + a^4 + a^9 + \dots\}^* \\ &= (a^*) \\ &= \text{regular} \end{aligned}$$

It is possible that closure of non regular language is regular.

$$L = \{a^n \mid n \text{ is prime}\}$$

$$\begin{aligned} &= \{a^2 + a^5 + a^7 + a^{11} + a^{13} + a^{15} + \dots\} \\ &= \underline{a^2} a^* = a a a^* = a^* - \{a\} \\ &= \text{regular} \end{aligned}$$

for Reversal L is reverse regular $\Leftrightarrow L^R$ is also regular L is CFL $\Leftrightarrow L^R$ is also CFLfor h(L)if $\Phi(L)$ is regular $\Rightarrow h(L)$ is also regularif $h(L)$ is regular $\Rightarrow L$ may or may not be regular

23-Ab91

Page No.

Date : / /

DECIDABILITY - for a problem to be decidable, an algorithm exist which solve problem in finite amount of time & it must halt.

Undecidable - No algorithm exist
if algorithm exist, it will give answer but for some input, it may go to hang.

A problem with answer 'Yes' or 'No' are decision problem

13 Problem - Decidable

27 ——— - Undecidable

A RLPG can be converted to LLRG or vice versa.

Trivial Decidable problem - O/P = Input

[i.e, problem of converting DFA to NFA
as each DFA itself a NFA]

Proper Decidable (like NFA to DFA) (13/40)

Decidable

(doing Nothing) (or always the same)

Trivial Decidable (like DFA to NFA)

Undecidable

Proper UD (No logic exist for this type of problem) (27/40)

Trivial Undecidable

(like NDPA $\rightarrow$ DPDA)

as these are UD because of power of NDPA is more than DPDA we can't convert it.

$\therefore$ there are problem which are TUD
i.e which are UD because of power)

EXCELLENT

Ex- DPDA to NPDA $\rightarrow$ Trivial Decidable.

NPDA to DPDA $\rightarrow$ Trivial Undecidable

All the decision problem is

Page No.

Date. / /

The complement of
decidable problem
is always decidable
or vice-versa

1. NPDA has more power than DPDA

2. LBA's & TM are going on to research.

(FA
Type 3)
RE

(PDA
Type 2)

(Type 1
LBA)

(HTM)

(Type 0
TM)

REG

CFL

CSL

REC

RE

1. MEMBERSHIP ($w \in L$?)

✓

✓

✓

✓

×

2. EMPTINESS ($L = \emptyset$?)

✓

✓

×

×

×

3. FINITENESS ($L = \text{Finite}$?)

✓

✓

×

×

×

4. EQUIVALENCE ($L_1 = L_2$?)

✓

×

×

×

×

5. REGULARITY ($L = \text{Regular}$?)

✓

×

×

×

×

(of language)
6. AMBIGUITY ($L = \text{Ambiguous}$?)

✓

×

×

×

×

7. UNIVERSALITY ($L = \Sigma^*$?)

✓

×

×

×

×

8. DISJOINTNESS ($L_1 \cap L_2 = \emptyset$?)

✓

×

×

×

×

undecidable
because they are
not REC ha

undecidable
because they are
not REC ha
HTM.

A problem is Decidable if and only if $L(P)$ is REC.

So if ques asked a problem is REC or not, check whether
it is decidable or not.

if a problem is Undecidable then it is Not REC i.e RE or
Not RE

EXCELLENT

Membership-

• $w \in L(G)$ iff there exist an algo derivation of w using production

or

$w \in L(M)$ iff M accept w

or

$w \in L(G)$, iff G generate w .

• PDA, HTM, FA, LBA Halts on every input.

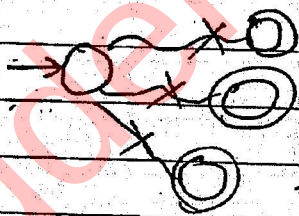
• So Machine can tell in finite time where w is member of L or not.

• for RE, it may or may not halt, so membership is undecidable as if you give some input, it goes to loop so it is undecidable.

Emptiness -

$L(M) = \emptyset$

iff there is no directed path to reach final state



for Reg ~~set~~ is ~~undecidable~~ because if A is q_0 & $F = \{B, C, D\}$

I will use graph theory to check whether there any path b/w A, B, A, C, A, D it is decidable.

if A, B, A, C, A, D is no path b/w them $\therefore$ it is empty.

For CFL - CFL can make list of useful & useless elements or variables.

If the list of useless variable has start symbol then it accept empty language.

for CFL

$$L(G) = \emptyset \text{ if } S \in \text{start symbol, belongs to useless variable}$$

Emptiness is mainly discussed by grammar

CYK Algorithm - to check whether $w \in L(G)$ for G of type complexity $= O(n^3)$

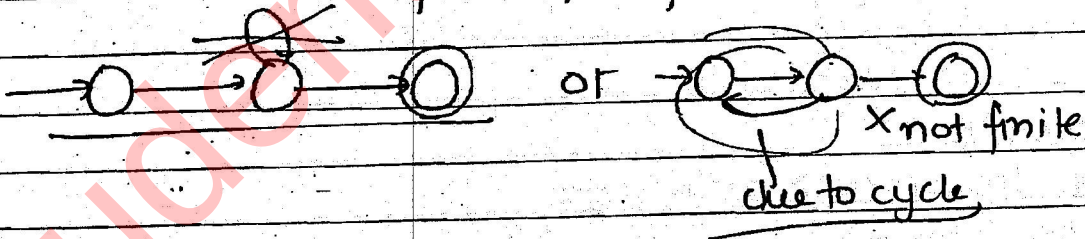
n is the length of word given as input to the algorithm i.e. (length of w)

Most efficient algorithm.

Finite Ness - for Finite Automata

for M , $L(M)$ is finite if and only if there is no cycle b/w directed between path from q_i

in directed path b/w final & initial state



Ques - FA has loop

1) it is finite

2) it is infinite

3) May or may not be finite

FA has loop in Path b/w q_0

1) it is surely infinite.

FA has loop but not in path of q_0 & F

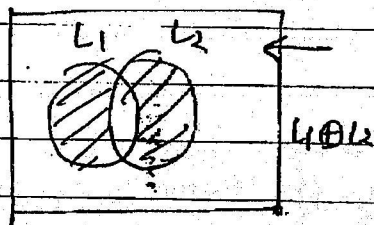
1) it is surely finite

it is decidable because graph theory can tell about cycle.

for CFG - bcoz there exist an algorithm to check about recursion & fruitful recursion.

EQUIVALENCE - Only two grs are equivalent
one gr & one pda are not equivalent
Type 0, 1, 2 are not decidable for equivalent

L_1, L_2 are equal iff $L_1 \oplus L_2 = \phi$



if $L_1 = L_2$



$$L_1 \bar{L}_2 + L_2 \bar{L}_1$$

$\downarrow$
 ϕ

$$(L_1 - L_2) \cup (L_2 - L_1)$$

$\downarrow$
 ϕ

as $L_2 \neq L_1$
 $\therefore L_2 \neq \bar{L}_1$
 $L_1 \bar{L}_2$

- develop M_1, M_2
- $M_1 \oplus M_2$ by product automata
- if $M_1 \oplus M_2$ produces ϕ , then (it is decidable as it is having an algorithm to solve)

for REC, Regular, CSL are closed under $\cup$ & $\bar{}$ so you can decide for $M_1 \oplus M_2$

but Regular is not decidable under emptiness only
but CSL, REC not so it is undecidable.

Halting problem of TM is undecidable & all undecidable & fundamental problem can be reduced by it. Page: / / Date: / /

• REGULARITY if L is Regular $\Leftrightarrow$ a finite automata for L

↓ It is a trivial decidability problem for regular language.

if a problem P_1 is reducible to another problem P_2

$$\therefore P_1 \leq P_2$$

• if P_1 is undecidable then P_2 is also undecidable.

• it is only one way - theorem.

if P_2 is undecidable, P_1 may or may not be undecidable

• if P_2 is decidable surely P_1 is also decidable by (contrapositive)

• if P_1 is decidable, P_2 may or may not be decidable

~~AMBIG~~

whether

• It is impossible to predict when a system will halt or not

Ambiguity - decidable for Regular language.

"A language is REC if the problem is decidable"

<u>Ques</u>	$I = L = \{ \langle w, M \rangle \mid w \in L(M), M \text{ is FA} \}$	A problem is undecidable if it has infinite no. of countable truth.
<u>Ans</u>	↓ REC as Membership is decidable for Regular language	

II. $L = \{ M \mid L(M) = \emptyset, M \text{ is TM} \}$ → Not REC as it is not decidable

$L = \{ M \mid L(M) = \emptyset, M \text{ is HTM} \}$ - Not REC

EXCELLENT

IV $L = \{ (G_1, L_2) \mid G_1 = G_2 \text{ if } M \text{ is FA } G_1 \text{ is Type 3} \}$

✓ REC as
it is decidable.

V $L = \{ (M_1, M_2) \mid M_1 = PDA, M_2 = NFA, M_1 = M_2 \}$

NP - NOT REC

VI $L = \{ (M_1, M_2) \mid M_1 = PDA, M_2 = NFA, M_1 \neq M_2 \}$

NP - NOT REC

as it is the complement problem.

VII $L = \{ (M_1, M_2) \mid M_1 = PDA, M_2 = NFA, M_1 \neq M_2 \}$

1. L is REC

2. L is RE but not REC

3. L is not RE

both can be the answer so choose 2 option as
it is very close to answer.

as No of 'not RE' problem is very less as compared
to 'RE but not REC' problem.