Register No: 19CSR057

Coursecode: 18CST57

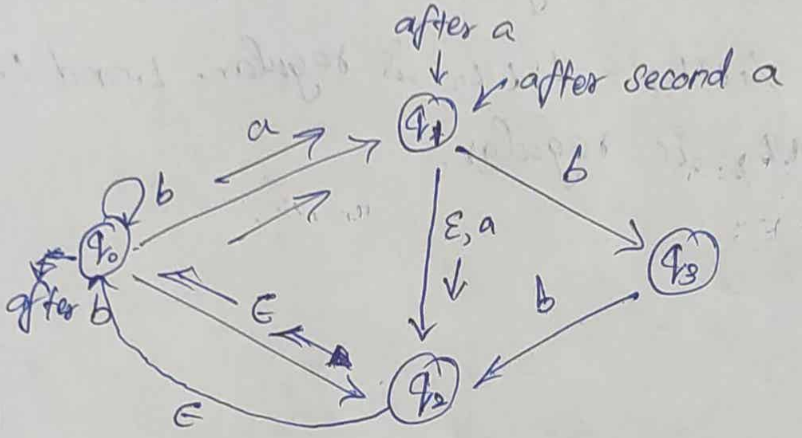
Course Name: Theory of Computation

Total No of Pages: 25.

END SEMESTER EXAMINATIONS.

PART-A

a)



Transition of $\delta^{\wedge}(q_2, abu)$ is $\{q_0, q_1, q_2\}$

b) formal definition of context free grammer.

A formal grammer is 'context free grammer' if Its Production rules can be applied regardless of a Context of a ~~nonterminal~~ non-terminal. No matter which Symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side.

4) Regular Languages are closed under intersection

Reason:-

Regular languages are closed under intersection i.e. if L_1 and L_2 are regular then $L_1 \cap L_2$ is also regular. Hence $L_1 \cap L_2 = L_1 \cup L_2$ is regular. L_1 and L_2 are regular. $L_1 \cup L_2$ is regular.

PART-B

14) a) i)

CFG into GNF

$S \rightarrow AB$
$A \rightarrow BS \mid b$
$B \rightarrow SA \mid a$

Solution

Step:1 The given grammer is already in CNF

Step:2 Renaming non-terminals.

Non-terminals in the grammer includes

S, A, B

∴ S is renamed as A_1
A is renamed as A_2

S is renamed as A_3

Step 3 :- Modifying the productions includes

$A_1 \rightarrow A_2 A_3$

$A_2 \rightarrow A_3 A_1 / b$

$A_3 \rightarrow A_1 A_2 / a$

Step: 4

In (i) $A_1 \rightarrow A_2 A_3$

$i = 1, j = 2$

$i < j$

So No Modification

(ii) $A_2 \rightarrow A_3 A_1 / b$

$i = 2, j = 3$

$i < j$

∴ No Modification

(iii) $A_3 \rightarrow A_1 A_2 / a$

$i = 3, j = 1$

$i \nless j$

∴ Modifications should be performed.

$A_3 \rightarrow A_1 A_2 / a$ is substituted as

$A_3 \rightarrow A_2 A_3 A_2 / a$

$i = 3$; $j = 2$

$i > j$

So substitude again

$A_3 \rightarrow A_3 A_1 A_3 A_2 \mid bA_3 A_2 \mid a$

$i = 3$. $j = 3$ $i = j$

So, stop the process

Here $A = A_3$

$\alpha = A_1 A_3 A_2$

$\beta_1 = bA_3 A_2$

$\beta_2 = a$

So having $A \rightarrow \beta i$

$A = \beta i\, x$

$x \rightarrow \alpha i$

$x \rightarrow \alpha i\, x$

Here x is taken as B_1

The new productions include

$A_3 \rightarrow bA_3 A_2 \mid a$

$A_3 \rightarrow bA_3 A_2 B_1 \mid aB_1$

$B_1 \rightarrow A_1 A_3 A_2$

$B_1 \rightarrow A_1 A_3 A_2 B_1$

$A_1 \rightarrow A_2 A_3$

$A_2 \rightarrow A_3 A_1 \mid b$

a) ii) Pumping Lemma for CFL:-

Theorem

Let L be any CFL, then there is a constant n depending only on L such that if z is in L, and $|z| \geq n$ then we may write $z = uvwxy$ such that

i) $|vx| \geq 1$
ii) $|vwx| \leq n$ and
iii) for all $i \geq 0$, uv^iwx^iy is in L.

$L = \{a^m b^m c^m : m \geq 0\}$

Let n be a pumping lemma constant where $z = a^m b^m c^m$ with $|z| = 3n \geq n$.

Split z into $uvwxy$

$$z = a^{m-i}\, a^i\, b^{m-i}\, b^i\, c^m$$

Pump v and x for 2 times such that

$$z = a^{m-i}\, a^i a^i\, b^{m-i}\, b^i b^i\, c^m$$
$$z = a^{m+i}\, b^{m+i}\, c^m \neq a^m b^m c^m$$

Since it is not ~~satisfaction~~ satisfies the pumping lemma property. It is not a CFL.

(8) b) i)

$S \to AS / \epsilon$

$A \to 0A_1 / A_1 / 01$

$L = \{0^n b^n / n \geq 1\}$ $G = \{N, T, P, S)$ $N = \{S\}$,

$T = \{0, 1\}$ $P = \{S \to AS/\epsilon, A \to 0A_1 / A_1 / 01\}$ $S = \{S\}$

Rule :1

$S \to AS$ $\delta(q_0, \epsilon, S) = (q_0, SA)$

$S \to \epsilon$ $\delta(q_0, \epsilon, \epsilon) = (q_0, \epsilon)$

$A \to 0A1$ $\delta(q_0, \epsilon, A) = (q_0, 1A0)$

$A \to A_1$ $\delta(q_0, \epsilon, A) = (q_0, 1A)$

$A \to 01$ $\delta(q_0, \epsilon, 01) = (q_0, 10)$

Rule 2 :-

$\delta(q_0, 0, 0) = (q_0, \epsilon)$

$\delta(q_0, 1, 1) = (q_0, \epsilon)$

$w = 0011$

$q_0, 0011, S \vdash (q_0, 0011, S)$

$\vdash (q_0, 0011, SA)$ $S \to AS$

$\vdash (q_0, 0011, 1A0)$ $A \to 0A1$

$\vdash (q_0, 011, 1A)$ (POP)

$\vdash (q_0, 011, 110)$ $A \to 01$

$\vdash (q_0, 11, 11)$ (pop)

$\vdash (q_0, 1, 1)$ (pop)

$\vdash (q_0, \varepsilon, \varepsilon)$ (pop)

$\therefore$ The given string is accepted.

iii) $a^n b^{2n} \mid a \geq 1$

$S \to asbb \mid abb$ $G = \{N, T, P, S\}$

$N = \{S\}$, $T = \{a, b\}$, $S = S$, P $S \to asbb \mid abb$

PDA $\Rightarrow M = (Q, \Sigma, \delta, \Gamma, Z_0, q_0, F)$

$Q = \{q_0\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, S\}$, $Z_0 = S$, q_0, $F = \phi$

Rule 1 $A \to \alpha$

$S \to asbb$ $\delta(q_0, \epsilon, S) = (q_0, bbSa)$

$S \to abb$ $\delta(q_0, \varepsilon, S) = (q_0, bba)$

Rule 2 terminal a, b

$\delta(q_0, a, a) = (q_0, \epsilon)$

$\delta(q_0, b, b) = (q_0, \epsilon)$

Eg:- aabbbb

q_0, aabbbb, S ⊢ (q_0, aabbbb, S) S → aSbb

⊢ (q_0, aabbbb, bbSa)

⊢ (q_0, abbbb, bbS) S → abb

⊢ (q_0, abbbb, bbbba)

⊢ (q_0, bbbb, bbbb)

⊢ (q_0, bbb, bbb)

⊢ (q_0, bb, bb)

⊢ (q_0, b, b)

⊢ (q_0, ε, ε) ∴ String & Stack is empty. ~~ss~~ accepted.

Eg: abbb

⊢ (q_0, abbb, S) S → a~~S~~bb

⊢ (q_0, abbb, bba)

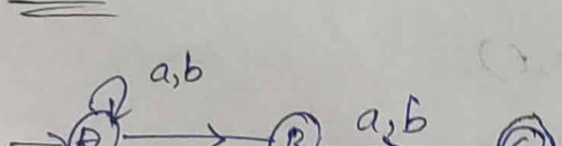
⊢ (q_0, bbb, bb)

⊢ (q_0, bb, b) ∴ Stack is Empty but

⊢ (q_0, b, ε) not the String rejected.

PART-A

8)



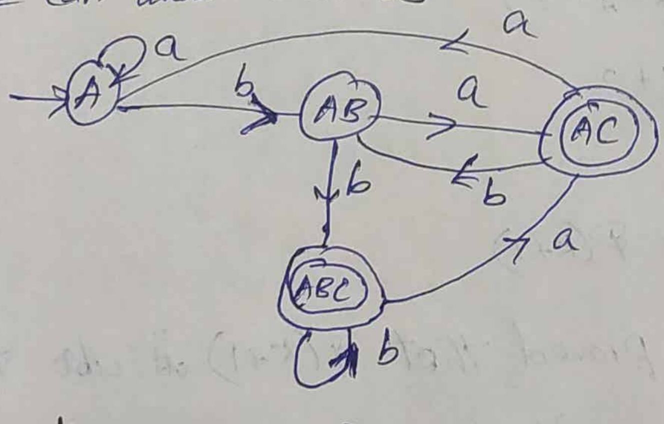
State transition table for NFA.

State	input a	input b
A	A	{A,B}
B	C	C
*C	{}	{}

Convert this table into table for DFA.

State	Input a	Input b
A	[A]	[AB]
[AB]	[AC]	[ABC]
*[AC]	[A]	[AB]
*[ABC]	[AC]	[ABC]

we can draw DFA as



Here no state is redundant

So states in minimal DFA = 4

11) a) i) $2+6+10+\dots+(4n-2)=2n^2$

Solution:

From the statement formula

when $n=1$ or $P(1)$

$LHS = 2$

$RHS = 2 \times 1^2 = 2$

So $P(1)$ is true.

Now we assume that $P(k)$ is true or $2+6+10+\dots+(4k-2)$
$= 2k^2$

For $P(k+1)$

$LHS = 2+6+10+\dots+(4k-2)+(4(k+1)-2)$

$= 2k^2+(4k+4-2)$

$= 2k^2+4k+2$

$= (k+1)^2$

$=$ RHS for $P(k+1)$

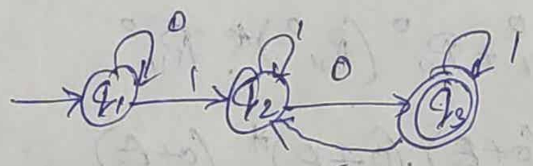
Hence proved that $P(k+1)$ is also true for the all positive integers n.

11) a) (ii)

Prove that if x is sum of the square of 4 +ve integers then $2^x \geq x^2$

Statement	Justification
1) $x = a^2 + b^2 + c^2 + d^2$	Given (4 integers a, b, c, d)
2) $a \geq 1, b \geq 1, c \geq 1, d \geq 1$	Given (+ve integers)
3) $a^2 \geq 1, b^2 \geq 1, c^2 \geq 1, d^2 \geq 1$	from ②, mathematical property of arthimetic.
4) $x \geq 4$	① & ③ and Property of arthimetic
5) $2^x \geq x^2$	④ & theorem

12) b) i)



$$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{(k-1)} (R_{kk}^{k-1})^* R_{kj}^{(k-1)}$$

k=0 iteration ①

$R_{11}^{0} = 0 + \epsilon$ $\quad R_{21}^{0} = \phi$

$R_{12}^{0} = 1$ $\quad R_{22}^{0} = 1 + \epsilon$

$R_{13}^{0} = \phi$ $\quad R_{23} = 0$

k=1 iteration ②

$$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{k-1}$$

$R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*} R_{11}^{0}$

$i=1, j=1$ $= 0+\epsilon + (0+\epsilon)(0+\epsilon)^{*}(0+\epsilon)$

$= (0+\epsilon)(0+\epsilon)^{*} = 0^{*}$

$i=1, j=2$ $R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}(R_{12})^{0}$

$= 1+0+\epsilon\,(0+\epsilon)^{*}.1$

$= 1[\epsilon + 0.\epsilon(0.\epsilon)^{*}) = 1.0^{*}$

$R_{13}^{1} = R_{13}^{0} + R_{11}^{0}(R_{11}^{0})^{*} R_{13}^{0}$

$i=1, j=3$ $= \phi$

$R_{ij}^{k} = R_{ij}^{k-1} + R_{jk}^{k-1}(R_{kk}^{k-1})^{*} R_{kj}^{k-1}$

$i=2, j=1$ $R_{21}^{k} = R_{21}^{0} + R_{11}^{0}(R_{11}^{0})^{*}.R_{11}^{0}$

$= \phi + (0+\epsilon).(0+\epsilon)^{*}(0+\epsilon)$

$= 0^{*}$

$i=2, j=2$ $R_{22}^{1} = R_{22}^{1} + R_{21}^{0}(R_{11}^{0})^{*} R_{12}^{0}$

$= 1+\epsilon$

$i=2, j=3$ $R_{23}^{1} = R_{23}^{0} + R_{31}^{0}(R_{11}^{0})^{*} R_{13}^{0}$

$= 0+\phi = 0$

$i=3, j=1$ ~~$R_{31}^{1} = R_{31}^{0} + R_{11}^{0}(R_{11}^{0})^{*} R_{11}^{0}$~~

$i=3$, $j=1$ $\quad R_{31}^{1} = R_{31}^{0} + R_{11}^{0}(R_{11}^{0})^{*}.R_{11}^{0}$

$= \phi + (1+\epsilon)(1+\epsilon)^{*}(1+\epsilon)$

$= 1^{*}$

$k=2$, iteration (2)

$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}(R_{22}^{1})^{*}.R_{21}^{1}$

$= 0^{*} + 1.0^{*}(1+\epsilon)^{*}.0^{*}$

$= 0^{*}[\epsilon + 1(1+\epsilon)^{*}] = 0^{*}.1^{*}$

$i=1$, $j=2$ $\quad R_{12}^{2} = R_{12}^{1} + R_{22}^{1}(R_{22}^{1})^{*}.R_{22}^{1}$

$= 1.0^{*} + 1+\epsilon\,(1+\epsilon)^{*}.(1+\epsilon)$

$= 1.0^{*} + 1^{*}$

$i=1$, $j=3$ $\quad R_{13}^{2} = R_{13}^{1} + R_{32}^{1}(R_{22}^{1})^{*}.R_{23}^{1}$

$= \phi + 0.(1+\epsilon)^{*}.0 = 0.1^{*}$

$R_{32}^{1} = R_{32}^{0} + R_{21}^{0}(R_{11}^{0})^{*}.R_{12}^{0}$

$= 0 + \phi = 0$

$R_{33}^{1} = R_{33}^{0} + R_{31}^{0}(R_{11}^{0})^{*}.R_{13}^{0}$

$= 1+\epsilon$

$i=2$, $j=1$ $\quad R_{21}^{2} = R_{21}^{1} + R_{12}^{1}(R_{22}^{1})^{*}.R_{21}^{1}$

$= 0^* + 1.0^*(1+\epsilon)^*. 0^*$

$= 0^* [\epsilon + 1.1^*.\epsilon) = 0^* 1^*$

$R_{22}^* = \cancel{R_{22}} + R_{22}^1 + R_{22}^1 (R_{22}^1)^*. R_{22}^1$

$= 1+\epsilon + (1+\epsilon)(1+\epsilon)^*.(1+\epsilon)$

$= (1+\epsilon)[\epsilon + (1+\epsilon)^*]$

$= 1.1^* = 1^*$

$R_{23}^2 = R_{23}^1 + R_{32}^1 (R_{22}^1)^*. R_{23}^1$

$i=2$
$j=3$

$= 0+0(1+\epsilon)^*.0$

$= 0[\epsilon + (1+\epsilon)^*].0.1^*$

$R_{31}^2 = R_{31}^1 + R_{32}^1 (R_{22}^1)^* R_{21}^1$

$= 1^* + 0(1+\epsilon)^* 0^*$

$= 1^* + 0^* 1^* = 1^* [\cancel{\epsilon} + 0^*] = 1^* 0^*$

$R_{32}^2 = R_{32}^1 + R_{32}^1 (R_{22}^1)^* R_{22}^1$

$= 0+0(1+\epsilon)^* 1+\epsilon$

$= 0[1^*] = 0.1^*$

$k=3$, final iteration

$R_{13}^3 = R_{13}^2 + R_{13}^2 (R_{33}^2)^*. R_{33}^2$

$= 0^*.1 + (0.1^*).(1+0.1^*.0)^*.(1+0.1^*.0)$

$= (0^*.1) [\epsilon + \epsilon.(1+0.1^*.0)^*]$

$= 0^*.1.1^* + (0.1^*.0)^*$

$= 0^*.1.1^* + (0.1^*.0)^* \Rightarrow 0^*.1.1^* + (0.1^*.0)^*$

15] b) i)

S.T TSP is NP-complete

The travelling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one and returning to the same city. The challenge of the problem is that the travelling salesman wants to minimize the total length of the trip.

~~Proff~~ Proof

To prove TSP is NP-complete, first we have to prove that TSP belongs to NP. In TSP we find a tour and check that the tour contains each vertex once. Then the total cost of the edges of the tour is calculated. finally we check if the cost is ~~maxim~~ minimum. This can be completed in polynomial time. Thus TSP belongs to NP.

Secondly, we have to prove that TSP is NP-hard. To prove this, one way is to show that Hamiltonian cycle $\leq_p$ TSP (as we know that the Hamiltonian cycle problem is NP complete

Assume $G=(V,E)$ to be an instance of Hamiltonian cycle.

Hence, an Instance of TSP is Constructed. We Create the Complete graph $G' = (V, E')$ where

$$E' = \{(i,j) : i,j \in V \text{ and } i \neq j\}$$

Thus, the cost function is defined as follows.

$$t(i,j) = \begin{cases} 0 & \text{if } (i,j) \in E \\ 1 & \text{otherwise} \end{cases}$$

Now, suppose that a Hamiltonian cycle h exists in G. It is clear that the cost of each edge in h is 0 in G' as each edge belongs to E therefore h has a cost of 0 in G'. Thus, if graph G has a Hamiltonian Cycle, then graph G' has a tour of 0 cost.

Conversely, we assume that G' has a tour h' of cost at most 0. The cost of edges in E' are 0 and 1 by definition. Hence, each edge must have a cost of 0 as the cost of h' is 0. We therefore conclude that h' contains only edges in E.

We have thus proven that G has a Hamiltonian cycle, if and only if G' has a tour of cost at most 0. TSP is NP complete.

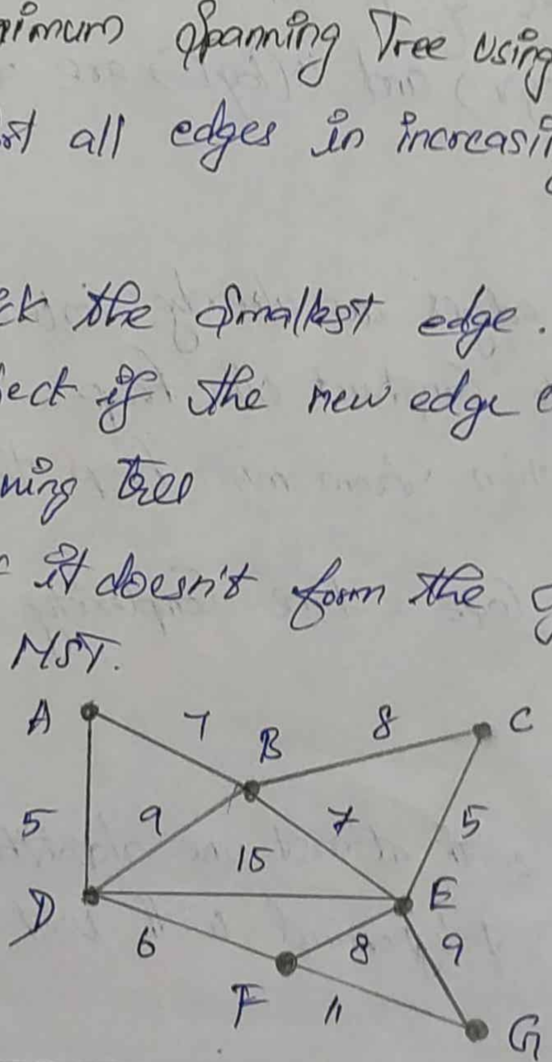
15) b) ii) Creating minimum spanning Tree using Kruskal Algorithm

- Step 1:- Sort all edges in increasing order of their edge weights

Step 2: Pick the smallest edge.

Step 3: Check if the new edge creates a cycle or loop in a spanning tree

Step 4: If it doesn't form the cycle, then include that edge in MST.



worst case time Complexity of kruskal's Algorithm
$= O(E\log V)$ or $O(E\log E)$

Analysis

* The edges are maintained as min heap
* The next edge can be obtained in $O(\log E)$ time if graph has E edges
* Reconstruction of heap takes $O(E)$ time
* So, kruskal's Algorithm takes $O(E\log E)$ time
* The value of E can be at most $O(V^2)$
* So, $O(\log V)$ and $O(\log E)$ are same

Special case:-

* If the edges are already sorted then there is no needs to Construct min heap
* So, deletion from min-heap time is saved
* In this case, time Complexity of kruskal's Algorithm $= O(E+V)$

Problems with atleast one algorithm that solves the problem in polynomial with wrt to Input size.

Polynomial time

The number of steps needed relates polynomially to the size of the Input

$O(n^2)$, $O(n^9)$, $O(n^c)$, where c is a Constant but Not $O(n!)$, $O(2^n)$, where n is the size of the Input.

Problems solvable in polynomial time using a Deterministic Turing Machine (DTM) belong to the class ~~Polyn~~ P

Polynomial time

* The no. of moves needed relates polynomially to the size of the input

$n^2, 17n^3, 9n^4$ but Not 2^n.

* The MWST problem belongs to the P class of Problems, since there is an algorithm that solves it in polynomial time

* Kruskal's algorithm $O(n^2)$

12) b) iii)

$L = \{0^P \mid P \text{ is a prime number}\}$

Let L be a RL and Let n be a pumping lemma Constant

$w = a^P \rightarrow P \geq n$

$|w| = P \geq n$

$w = xyz$ such that $y \neq \epsilon$, $|y| > 0$

$|xy| \leq n$

$\therefore 1 \leq m \leq n$

Let $y = a^m$ for some m,

$|xy^kz| = \underbrace{|x| + |y| + |z|}_{w} + |y^{k-1}|$

$= P + (k-1)(y)$

$= P + (k-1)m$

$= P + P.m$

$= P(1+m) \Rightarrow$ Not a prime Number

$z = uv^P wx^P y$

$= |uvwxy| + |v|^{P-1} + |x|^{P-1}$

$= P + (P-1)m + (P-1)n$

$= P + Pm - m + Pn - n$

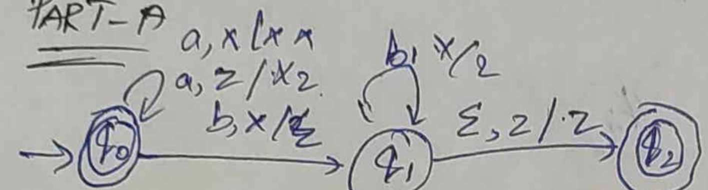
$z = P(1+m+n) - m - n$

It is not a prime

since $(P(1+m))$ is not a prime, $xy^k z \notin L$

Hence L is not a RL.

PART-A

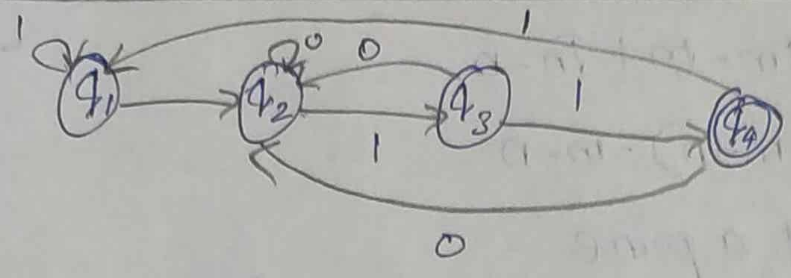
5) 

The transitions of PDA are

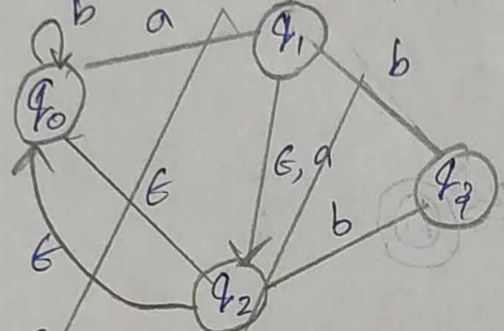
1) $(q_1, a, z) \rightarrow (q_1, xz)$
2) $(q_1, a, x) \rightarrow (q_1, xx)$
3) $(q_1, b, x) \rightarrow (q_2, \epsilon)$
4) $(q_2, b, x) \rightarrow (q_2, \epsilon)$
5) $(q_2, \epsilon, z) \rightarrow (q_3, z)$

Hence the language accepted by PDA is

$L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is deterministic Context-free (CFG).

1) 

$L = \{011, 00011, 1011, \ldots\}$

2) 

$\{q_0, q_1, q_2\} \rightarrow \delta^\wedge(q_2, aba)$

4)

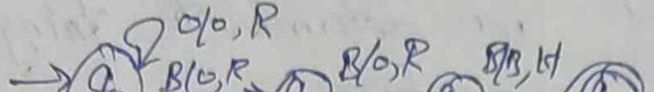
i)

ii) Undecidable since context free languages are not closed under complementation

iii) Decidable problems since regular languages are closed under complementation

(iv) Decidable ~~problems~~ problems since Recursive languages are closed under complementation.

9)



Input String Verification

w = 000BBB

↳ 000BBB

↳ 000BBB

↳ 000BBB

↳ 0000BB

↳ 00000B

↳ 00000B

As w halts, the given string is accepted by turing machine

Function of turing machine

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

10)

Class P	Class NP
P is the Complexity class containing decision problems which can be solved by a deterministic turing machine	In computational complexity theory NP is the set of decision problems solvable in polynomial time on a

using a polynomial amount of computation time, or polynomial time.	non-deterministic turing machine.
* P is often taken to the class of computational problems which are ~~effectively~~ efficiently solvable" or "tractable".	* It is the set of problems that can be "verified" by a deterministic turing machine in polynomial time.
* There exist problems in P which are interactable in practical terms for example some require at least $n^{1000000}$	* All the problems ~~that~~ in in this class have the property that their solutions can ~~so~~ be checked effectively

7) 1) To eliminate $A \to \epsilon$

$S \to ABAC \mid BAC \mid ABC \mid BC$

$A \to aA \mid a$

$B \to bB \mid \epsilon$

$C \to d$

2) To eliminate $B \to \epsilon$

$S \to ABAC \mid BA \mid ABC \mid BC \mid AAC \mid AC \mid C$

$A \to aA \mid a$

$B \rightarrow bB / b$

$C \rightarrow d$