Upayan De, CSE UG3, JU

Formal Languages and Automata Theory PYQs

(b) Prove that regular languages are closed under reversal.

Ans.

- Assume that a given Regular Language **L** is defined by a Regular Expression **E.**
- We will show that there exists a Regular Expression $\mathbf{E^R}$ such that $\mathbf{L(E^R) = (L(E))^R}$
- This can be done by Mathematical Induction.
- **Basis:** If E is **ϵ, ∅ or a** where a is any symbol of the alphabet. In this case, $\mathbf{E^R = E}$. This is because $\{\epsilon\}^R = \{\epsilon\}$, $\{\}^R = \{\}$ and $\{a\}^R = \{a\}$.
- **Induction:** There are three cases depending on the form of E:
  1. $\mathbf{E = E_1 + E_2 \Rightarrow E^R = E_1^R + E_2^R}$
  2. $\mathbf{E = E_1E_2 \Rightarrow E^R = E_2^RE_1^R}$
  3. $\mathbf{E = (E_1)^* \Rightarrow E^R = (E_1^R)^*}$
- Thus, by Mathematical Induction, L^R is Regular.

3 (a) State the Pumping Lemma for regular languages.

Ans.

- Let **L** be a Regular Language. Then,
- **There exists** a constant **n** (which depends on L), such that
- **For every** string **w** in L with $\mathbf{|w| \geq n}$,
- **There exists** a partition of w into three strings, **w = xyz** such that
  - $\mathbf{y \neq \epsilon}$
  - $\mathbf{|xy| \leq n}$
  - **For all** $\mathbf{k \geq 0}$, xy^kz is also in L.

(b) Consider the following language L over the alphabet Σ={a,b,c}.
$L=\{a^ib^jc^k \mid i,j,k\geq 0$ and if $i=1$ then $j=k\}$
Prove that L is not regular.

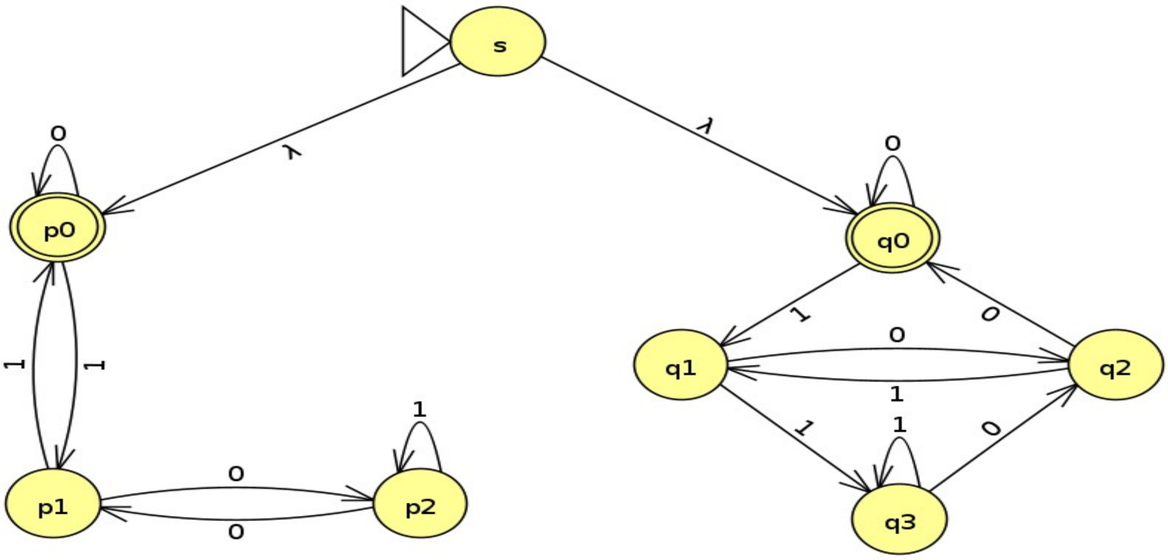
Ans.

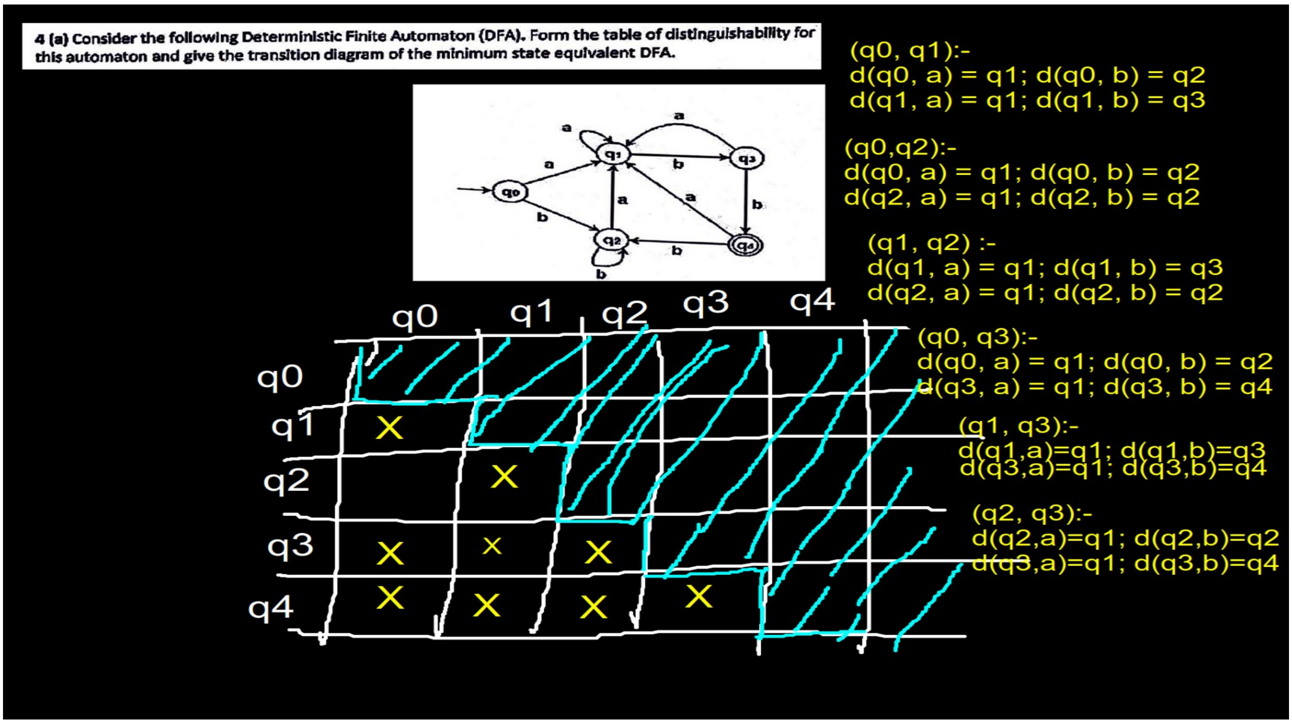
- We assume **L** is Regular.
- Let **F** be the Regular Language defined by the Regular expression $\mathbf{a(b + c)^*}$
- Let $\mathbf{L' = L \cap F}$. It is clear that $\mathbf{L' = \{ab^kc^k, k \geq 0\}}$.
- Since the intersection of two Regular Languages is Regular, **L' must be Regular.**
- Thus, there exists a constant **n** satisfying the conditions of the Pumping Lemma for L'.
- Let $\mathbf{w = ab^nc^n}$. It is clear that w is in L', and $|w| \geq n$.
- By the Pumping Lemma, we can break **w = xyz,** such that $\mathbf{y \neq \epsilon}$ and $\mathbf{|xy| \leq n}$.
- Since $|xy| \leq n$, it is clear that **xy contains only a and b's, and does not contain c.**
- Case 1: ($\mathbf{x = ab^p, y = b^q}$)
  - Here $p \geq 0$ and $q > 0$.
  - Consider the string $\mathbf{xy^0z}$. By the Pumping Lemma, this string must belong to L'.
  - However, this string is of the form $\mathbf{ab^{n-q}c^n}$, since y was removed.

  - $y \neq \epsilon \Rightarrow q \neq 0 \Rightarrow n - q \neq n$. So the given string does not belong to L'.
- Case 2: ($\mathbf{x = \epsilon, y = ab^p}$)
  - Here $p \geq 0$.
  - Consider the string $\mathbf{xy^0z}$. By the Pumping Lemma, this string must belong to L'.
  - However, this string is of the form $\mathbf{b^{n-p}c^n}$, which does not contain **a**.
  - Clearly, this string does not belong to L'.
- Thus, we arrive at a contradiction, which proves that our initial assumption was false. So **L** is not Regular.

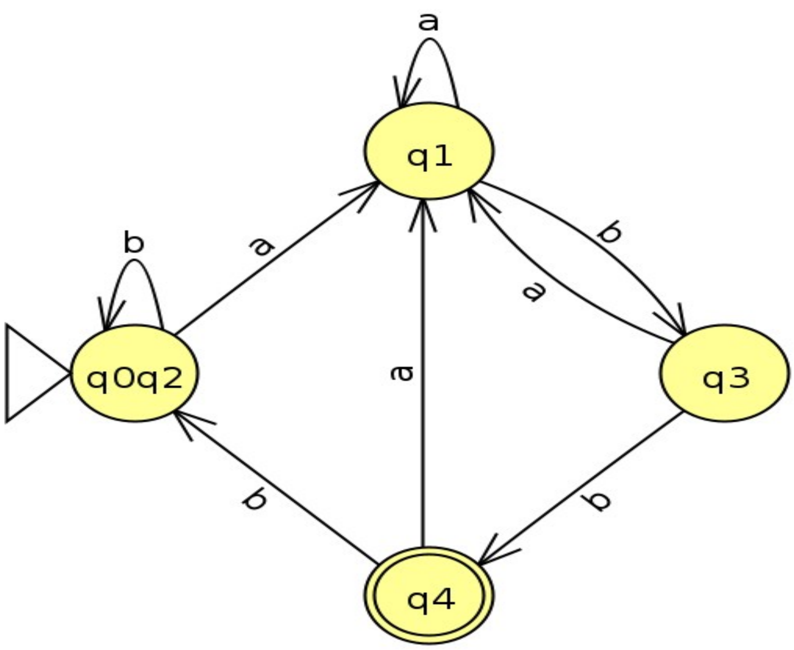
2 (a) Give the transition diagram of a Non-Deterministic Finite Automaton (NFA) that accepts all binary integers divisible by 3 or 4.

Ans.





	a	b
q0q2	q1	q0q2
q1	q1	q3
q3	q1	q4
*q4	q1	q0q2



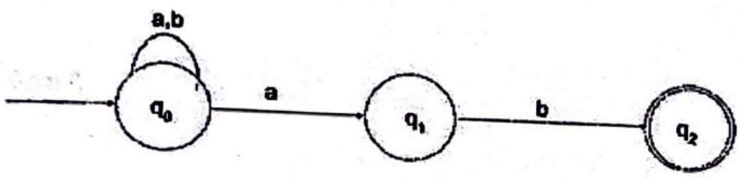
(b) Prove that no other equivalent DFA with lesser number of states than the one obtained by your method can exist. (5+7)

Ans.

- Let the minimised DFA obtained by the table-filling algorithm be **M.** It follows that every state in M is distinguishable from every other state in M.
- Let us assume that there exists a DFA **N**, such that **L(M) = L(N),** and **N has fewer states than M**.
- The start states of M and N are indistinguishable, since L(M) = L(N).

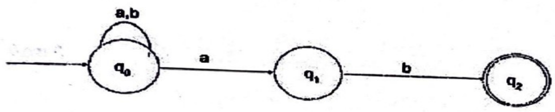
- Now, if {p, q} are indistinguishable, then their successors on any one input symbol are also indistinguishable. If not, we could distinguish p from q.
- Neither M nor N have an inaccessible state. If so, we could simply remove that state and obtain an even smaller DFA for the same language.
- Now, we claim that **every state in M is indistinguishable from at least one state in N.**
- Proof:
  - Let **p** be a state of M.
  - Then there is some string $\mathbf{a_1 a_2 a_3 ... a_n}$ which takes the start state of M to state p.
  - The above string takes the start state of N to some state. Let this state be **q**.
  - Since the start states of M and N are indistinguishable, it follows that their respective successors on input a_1 are also indistinguishable.
  - The successors of these states on input a_2 are also indistinguishable, and so on until we conclude that **p and q are indistinguishable**.
- Since M has fewer states than N, it follows that there are at least **two states in M which are indistinguishable from the same state in N,** by the **Pigeonhole Principle.**
- In that case, these two states in M are indistinguishable from each other.
- However, this is a contradiction due to the **design** of the DFA M.
- Thus, our assumption was wrong, and such a DFA N cannot exist.

5 (a) Consider the following Non Deterministic Finite Automaton (NFA). Give the transition diagram of an equivalent DFA. (5+7)



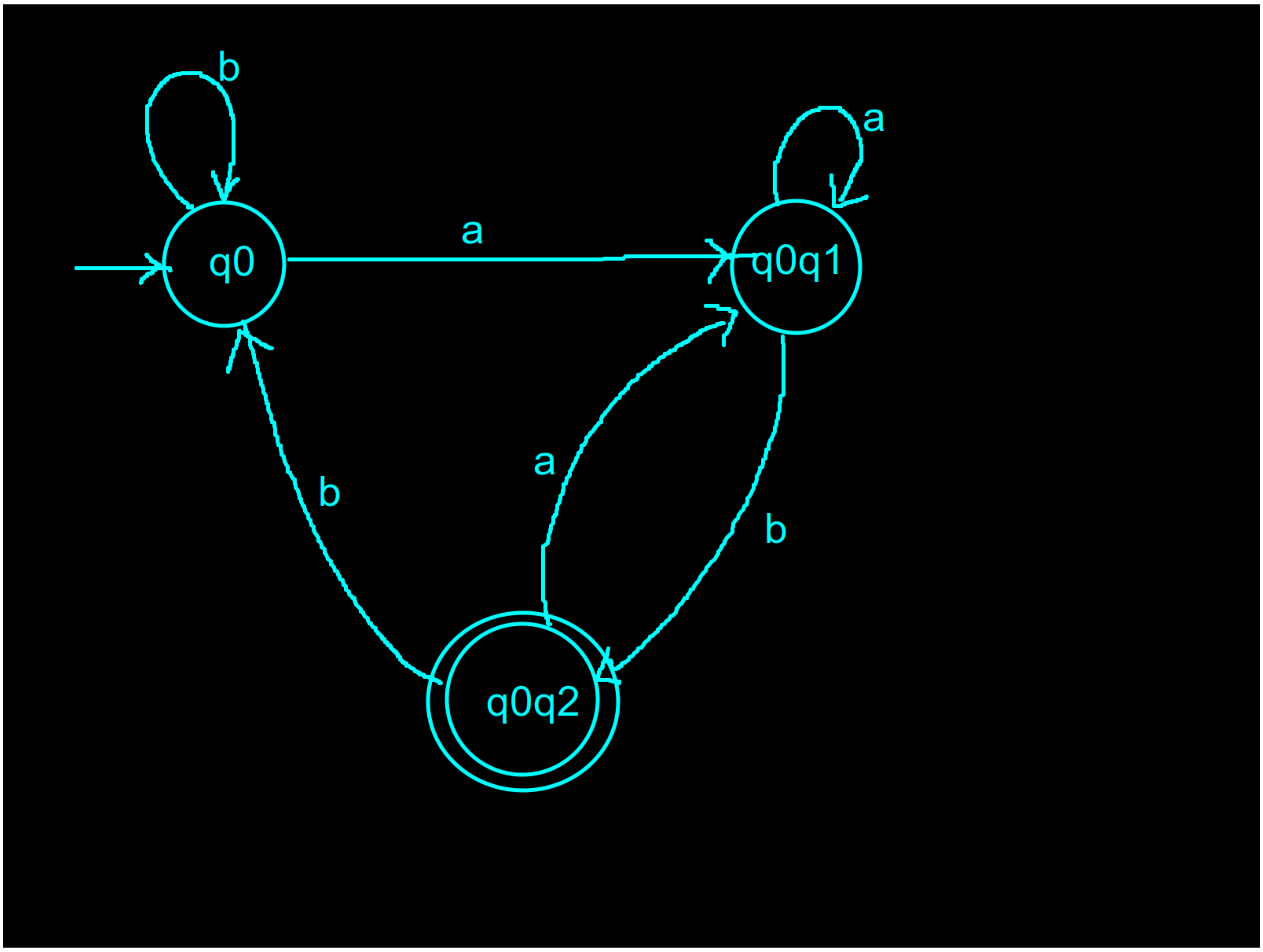
Ans

5 (a) Consider the following Non Deterministic Finite Automaton (NFA). Give the transition diagram of an equivalent DFA. (5+7)



	a	b
q0	{q0, q1}	{q0}
q1	{}	{q2}
*q2	{}	{}

	a	b
q0	q0q1	q0
q0q1	q0q1	q0q2
*q0q2	q0q1	q0



(b) Let Σ={a,b}. Given a language L ⊆Σ* and a string w∈Σ*, consider the following two languages:
Extend(L,w):={xw|x∈L} , Shrink(L,w):={x|xw∈L}

Show that if L is regular, both Extend(L,w) and Shrink(L,w) are regular. 4+(4+4)

Ans

- Let **R** be the RE which defines the language **L.**
- Then, the RE **Rw** defines the language Extend(L, w).
- This is because the language defined by the RE w is {w}, as w consists of symbols of the alphabet only, and no operator.
- Hence, Extend(L, w) is regular.

6(a) State the Pumping Lemma for context free languages.

Ans.

- Let **L** be a Context-Free Language. Then,
- **There exists** a constant **n** (which depends on L), such that
- **For every** string **z** in L with |**z**| ≥ **n**,
- **There exists** a partition of z into five strings, **z=uvwxy,** such that
  - **vx** $\neq$ **ϵ**
  - **|vwx| ≤ n**
  - **For all i ≥ 0,** uv^iwx^iy is also in L.

(b) Let $\Sigma=\{a,b\}$, $L=\{a^n b^j a^n b^j \mid n,j \geq 0\}$. Prove that L is not a context free language.

Ans.

- Assume that **L** is a CFL.
- Thus, there exists a constant **n** satisfying the conditions of the Pumping Lemma for CFLs
- Let **$z = a^n b^n a^n b^n$.** It is clear than z is in L, and $|z| \geq n$.
- By the Pumping Lemma, we can break **z=uvwxy** such that **$vx \neq \epsilon$** and **$|vwx| \leq n$**
- Since $|vwx| \leq n$, there can only be two cases:
- Case 1: vwx consists of only one kind of symbol
  - Consider the string **uv^0xw^0y.** By the Pumping Lemma, this string must be in L.
  - However, the above string has **n occurrences of three symbols** and **less than n of the fourth symbol,** since $vx \neq \epsilon$.
  - Thus, the string cannot belong to L.
- Case 2: vwx straddles two different symbols
  - Assume that vwx straddles the second block of b's and the first block of a's.
  - Consider the string **uv^0xw^0y.** By the Pumping Lemma, this string must be in L.
  - However, the above string is missing **either some a's or some b's or both,** since $vx \neq \epsilon$.
  - If the string is missing a's, then it cannot be in L because the second block of a's is of length n.
  - If the string is missing b's, then it cannot be in L because the first block of b's is of length n.
  - Similar arguments can be made for all other cases where vwx straddles two symbols.
- Thus, our assumption was wrong and L is not a CFL.

7(a) Let G be a GFG in Chomsky normal form.
Show that for any string $w \in L(G)$ of length $n > 1$, exactly $2n - 1$ steps are required for any derivation of w

Ans.

- We will prove this using **Induction** on the length of the string **|w|**.
- **Basis:**
  - When $|w| = 1$, the derivation must be of the form $G \Rightarrow w$.
  - Since there are no ϵ-derivations in a CNF grammar, so if the first derivation is of the form $G \Rightarrow AB$, then the length of the terminal string will exceed 1.
  - The number of steps in the derivation = 1 = 2(1) – 1. So the basis is proved.
- **Induction:**
  - Assume that the hypothesis holds for all strings of length < n.
  - Let $|w| = n$. Let $w = a_1a_2...a_n$.
  - The derivation must be of the form $G \Rightarrow AB \Rightarrow a_1B$ where $B \overset{*}{\Rightarrow} a_2a_3...a_n$
  - The length of the string derived by B = n – 1. Therefore, by the Induction hypothesis, this derivation takes **exactly 2(n – 1) – 1 = 2n – 3** steps.
  - Since **two more** steps are required to derive w, total number of steps to derive w = **2n – 3 + 2 = 2n – 1**.
  - Thus, the induction is proved.

... steps are required for any derivat...

(b) Consider the language $L = \{0^i1^j2^j3^{3i} \mid i,j >= 1\}$. Give a context free grammar for the language L

Ans

- The CFG for the language L is given by **G = (V, T, P, S)** where,
- V = {S, A}
- T = {0, 1, 2, 3}
- P is the set of productions as follows:

$$\mathbf{S \rightarrow 0S333 \mid 0A333}$$
$$\mathbf{A \rightarrow 1A2 \mid 12}$$

- S is the start state.

ii) Which binary strings does the regular expression $(0+10)^*(\epsilon+1)$ represent ?

Ans. All binary strings that do not contain two consecutive 1's.

vii) Which of the following statements about regular languages is NOT true?

(A) Every language has a regular superset.
(B) Every language has a regular subset.
(C) Every subset of a regular language is regular.
(D) Every subset of a finite language is regular.

...uages over the alphabet {a,b,c,d} ... $L=\{a^nb^mc^md^n \mid n,m\geq 0\}$.

Ans. C

- (A) Σ^* is a superset of every language
- (B) ϕ is a subset of every language
- (C) 0^n1^n is a subset of Σ^*
- (D) Finite languages are regular

8 (a) Prove that context free languages are not closed under set intersection.

Ans.

- Let us assume that CFLs are closed under set intersection.
- Consider the languages
  - $\mathbf{L_1 = \{a^xb^xc^y, x \geq 1, y \geq 1\}}$
  - $\mathbf{L_2 = \{a^xb^yc^y, x \geq 1, y \geq 1\}}$
- A Context-Free Grammar for L_1 is given by the productions

$$S \rightarrow AB$$
$$A \rightarrow aAb \mid ab$$
$$B \rightarrow Bc \mid c$$

  Thus, **L_1 is a CFL**.
- A Context-Free Grammar for L_2 is given by the productions

$$S \rightarrow AB$$

$$A \rightarrow aA \mid a$$
$$B \rightarrow bBc \mid bc$$

- The language $\mathbf{L = L_1 \cap L_2}$ is then given by $\mathbf{\{a^x b^x c^x,\ x \geq 1\}}$
- By our assumption, L must also be a CFL.
- However, L is **not a CFL,** which can be proved using the Pumping Lemma for CFLs.
- We arrive at a contradiction. So, our initial assumption is incorrect, and CFLs are not closed under set intersection.

<u>Proof that L is not a CFL</u>

- Assume **L** is a CFL.
- Thus, there exists a constant **n** (depending on L) satisfying the conditions of the Pumping Lemma for CFLs.
- Let $\mathbf{z = a^n b^n c^n}$. It is clear that w is in L and $|w| \leq n$.
- By the Pumping Lemma, we can break **z = uvwxy**, such that $\mathbf{vx \neq \epsilon}$ and $\mathbf{|vwx| \leq n}$.
- Since $|vwx| \leq n$, there are two cases:
- **Case 1**: vwx contains only one kind of symbol, say *a*
  - Consider the string $\mathbf{uv^0wx^0y}$. By the Pumping Lemma, this string must belong to L.
  - However, the string has n occurrences of *b* and *c* and < n occurrences of *a*.
  - $vx \neq \epsilon$, so at least one occurrence of *a* was dropped.
  - Thus the string cannot belong to L.
- **Case 2**: vwx straddles two symbols say *a* and *b*
  - Consider the string $\mathbf{uv^0wx^0y}$. By the Pumping Lemma, this string must belong to L.
  - However, the string has n occurrences of *c* and < n occurrences of either *a* or *b* or both.
  - $vx \neq \epsilon$, so at least one occurrence of either *a* or *b* was dropped.
  - Thus the string cannot belong to L.
- We see that, in all the cases we arrive at a contradiction.
- So, our initial assumption was wrong, and L is not a CFL.

(b) Let C=A context free language, R=A regular language. Determine if R-C is context free.

Ans.

- R – C = R ∩ C'
- We know that the complement of a CFL may or may not be a CFL.
- **Case 1:** C' is a CFL
  - We know that the intersection of a RL and a CFL is a CFL.
  - Thus R – C is a CFL.
- **Case 2:** C' is not a CFL
  - In this case, we cannot determine whether R – C is a CFL.

ANSW

9(a) State Halting problem for Turing machines.

Ans.

- Let **M** be a Turing Machine.
- Let **H(M)** denote the set of input strings **w** such that
  - M halts given input w, regardless of whether it accepts w or not.
- The **Halting Problem** is then the language defined by the set of **all pairs (M, w) such that w is in H(M).**
- This language is RE but not Recursive.

(b) Assuming undecidability of Halting problem for Turing machines, prove undecidability of Blank tape halting problem. **(2+5)**

Ans.

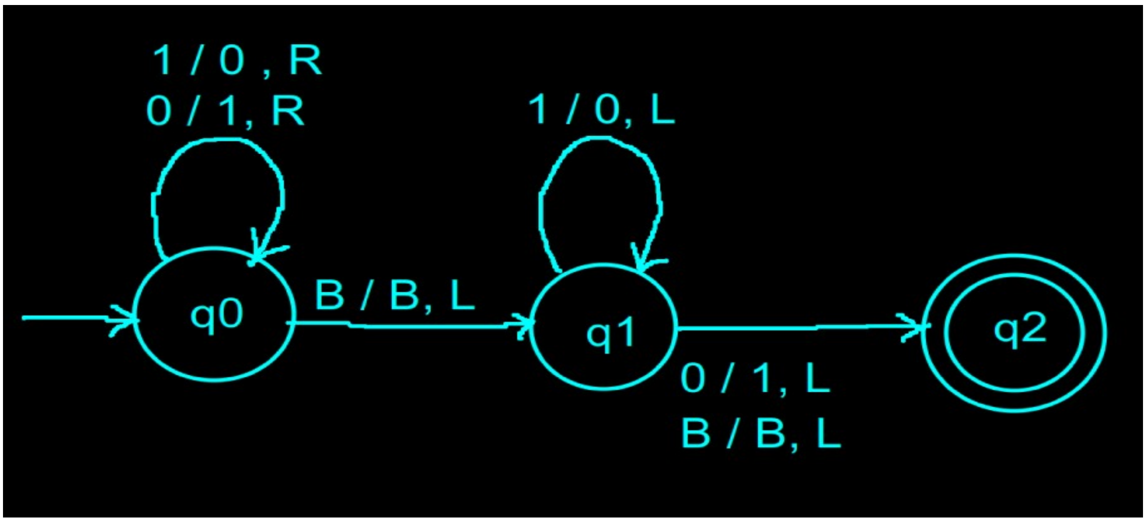
10) Define P, NP, NP hard and NP complete classes of problems.

Ans.

- **P:** We say a language **L** is in class P if
  - There exist some **deterministic** Turing Machine **M** and some polynomial **T(n)** such that
    - **L = L(M)**
    - When M is given **any** input of length **n,** there are no sequences of more than T(n) moves of M.
  - Eg: **Kruskal's Algorithm, Sorting Problem**
- **NP:** We say a language **L** is in class NP if
  - There exist some **non-deterministic** Turing Machine **M** and some polynomial **T(n)** such that
    - **L = L(M)**
    - When M is given **any** input of length **n**, there are no sequences of more than T(n) moves of M.
  - Eg: **Travelling Salesman Problem**
- **NP-Hard:** We say a language **L** is NP-hard if
  - For every language **L'** in NP, there is a polynomial time reduction of L' to L.
  - We cannot prove that L is in NP.
  - Eg: **Subset Sum Problem**
- **NP-Complete:** We say a language **L** is NP-complete if
  - For every language **L'** in NP, there is a polynomial time reduction of L' to L.
  - L is in NP.
  - Eg: **SAT (Boolean Satisfiability)**

11) Give the transition diagram of a Turing machine that produces 2's complement of a binary number. **(7)**

Ans.



- Here the Turing Machine is given by **$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_2\})$**
- δ is defined by the above transition diagram.

(b) How would you prove that a problem is in NP-hard? How would you prove that a problem is in NP-complete?

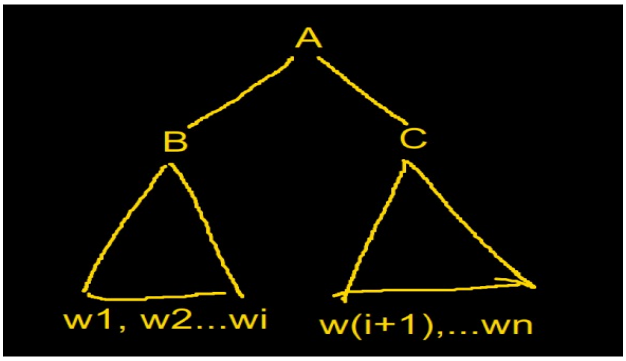
Ans.

- To prove a problem **H** is NP-hard, we show that there is a **polynomial time reduction** *from* a known NP-hard problem (such as SAT) *to* H.
- To prove that a problem **H** is NP-complete we
  - Prove that H is **NP-hard** as described above.
  - Show that H is **in NP**, by providing a suitable polynomial time non-deterministic algorithm to solve the problem.

2(a) Let **w** be the yield of a Parse tree formed by a grammar in Chomsky Normal Form. Also assume that the length of the longest path in the Parse tree is **n**. Then prove that **$|w| \leq 2^{n-1}$**.

Ans.

- We will prove this using Induction on n.
- **Basis: n = 1**
  - In this case, the Parse tree contains just the root and one leaf node.
  - The terminal symbol at the root node must be w. So $|w| = 1 \leq 2^{1-1}$.
- **Induction: $n \geq 2$**
  - Assume the inductive hypothesis holds for all parse trees with length of longest path < n.
  - The production at the root of the tree must be of the form $A \rightarrow BC$, since this is not the last level.



  - The longest path in the subtrees rooted at B and C $\leq n - 1$, since the edge from A to B or A to C is excluded.
  - Thus using the Inductive Hypothesis, the yields of the subtrees rooted at B and C have length $\leq \mathbf{2^{n-1-1} = 2^{n-2}}$.
  - Since the yield of the Parse tree rooted at A is a concatenation of the yields of subtrees B and C, $|w| \leq \mathbf{2^{n-2} + 2^{n-2} = 2^{n-1}}$.

(b) State the Pumping lemma for Context Free Languages (CFLs)

(c) Using the Pumping lemma, prove that

$$L=\{ww|w\epsilon\{0,1\}^*\}$$

is not a CFL.

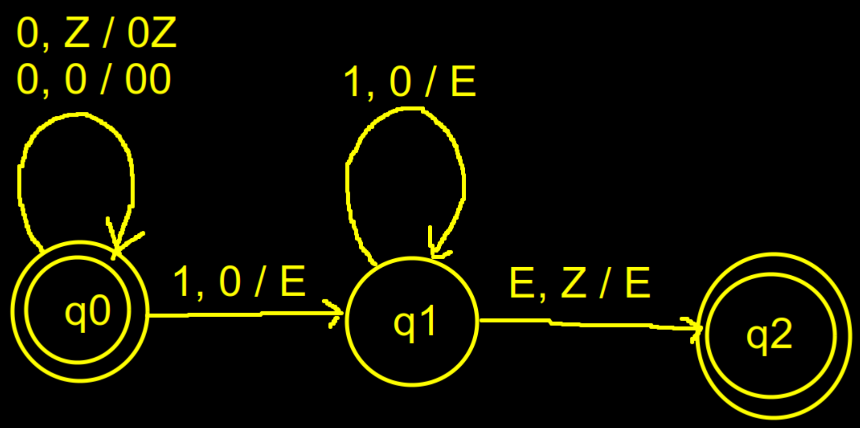
Ans.

- For every context free language **L**

- **There exists** a constant n (depending on L) such that
- **For every** string **z** in L, such that $|z| \geq n$,
- **There exists** a partition of z into five strings **uvwxy**, such that
  - $vx \neq \epsilon$
  - $|vwx| \leq n$
  - **For all** $i \geq 0$, uv^iwx^iy is in L.

- Assume L is a CFL.
- Thus there exists a constant n satisfying the conditions of the Pumping Lemma for CFLs.
- Let $z = 0^n1^n0^n1^n$. It is clear that z is in L, and $|z| \geq n$.
- By the Pumping Lemma, we can break z = uvwxy such that $vx \neq \epsilon$ and $|vwx| \leq n$.
- Also by the Pumping Lemma, uv^0wy^0x = uwy must be in L.
- Since $|vwx| \leq n$, so $|uwy| \geq 3n$. If uwy = tt, then $|tt| \geq 3n / 2$.
- There are many cases to consider depending on the position of vwx:
- **Case 1**: vwx is in the first block of 0's:
  - Let vx consist of k 0's, $k > 0$ as $vx \neq \epsilon$. Now, the string is of the form $0^{n-k}1^n0^n1^n$.
  - $|uwy| = 4n - k$. If uwy = tt, then $|t| = 2n - k / 2 > 2n - k$.
  - So, t does not end until after the first block of 1's.
  - So, t ends in a 0. However uwy ends in a 1. So uwy $\neq$ tt for any string t.
- **Case 2**: vwx straddles the first block of 0's and 1's
  - If vx contains only 0's ($x = \epsilon$), then the argument is the same as Case 1.
  - If vx contains at least one 1, then the first block of 1's has length < n.
  - uwy ends with a string of n 1's. If uwy = tt, then t must also end with n 1's.
  - But there is only one block on n 1's in uwy. So uwy $\neq$ tt for any t.
- **Case 3**: vwx is in the first block of 1's
  - The argument is the same as the second part of Case 2.
- **Case 4:** vwx straddles the first block of 1's and the second block of 0's
  - If vx contains only 1's ($x = \epsilon$), then the argument is the same as the second part of Case 2
  - If vx contains at least one 0, then the second block of 0's has length < n.
  - uwy begins with a block of n 0's. If uwy = tt, then t must also begin with n 0's.
  - But there is only one block of n 0's in uwy. So uwy $\neq$ tt, for any t.
- Symmetric arguments can be made for the remaining cases.
- Thus in all the cases, we arrive at a contradiction. So our assumption is wrong, and L is not a CFL.

3(a) Give the state diagram of a Deterministic Push Down Automaton (DPDA) to accept
$L=\{0^n|n\geq0\}\cup\{0^n1^n|n\geq0\}$



(b) Eliminate ϵ-productions from the following Context Free Grammar:

S→aXbY
X→aX| ϵ
Y→bY| ϵ

Eliminating X -> E,

S -> aXbY | abY
X -> aX
Y -> bY | E

Eliminating Y -> E,

S -> aXbY | abY | aXb | ab
X -> aX
Y -> bY

(c) Give a Context Free Grammar (CFG) for L={x∈{0,1}$^+$| symbol at position i is same as symbol at position i+2 and |x|≥2}

Strings generated = {00, 10, 01, 11, 001, 010, 101, 111, 0000, 1010, 01010, ...}

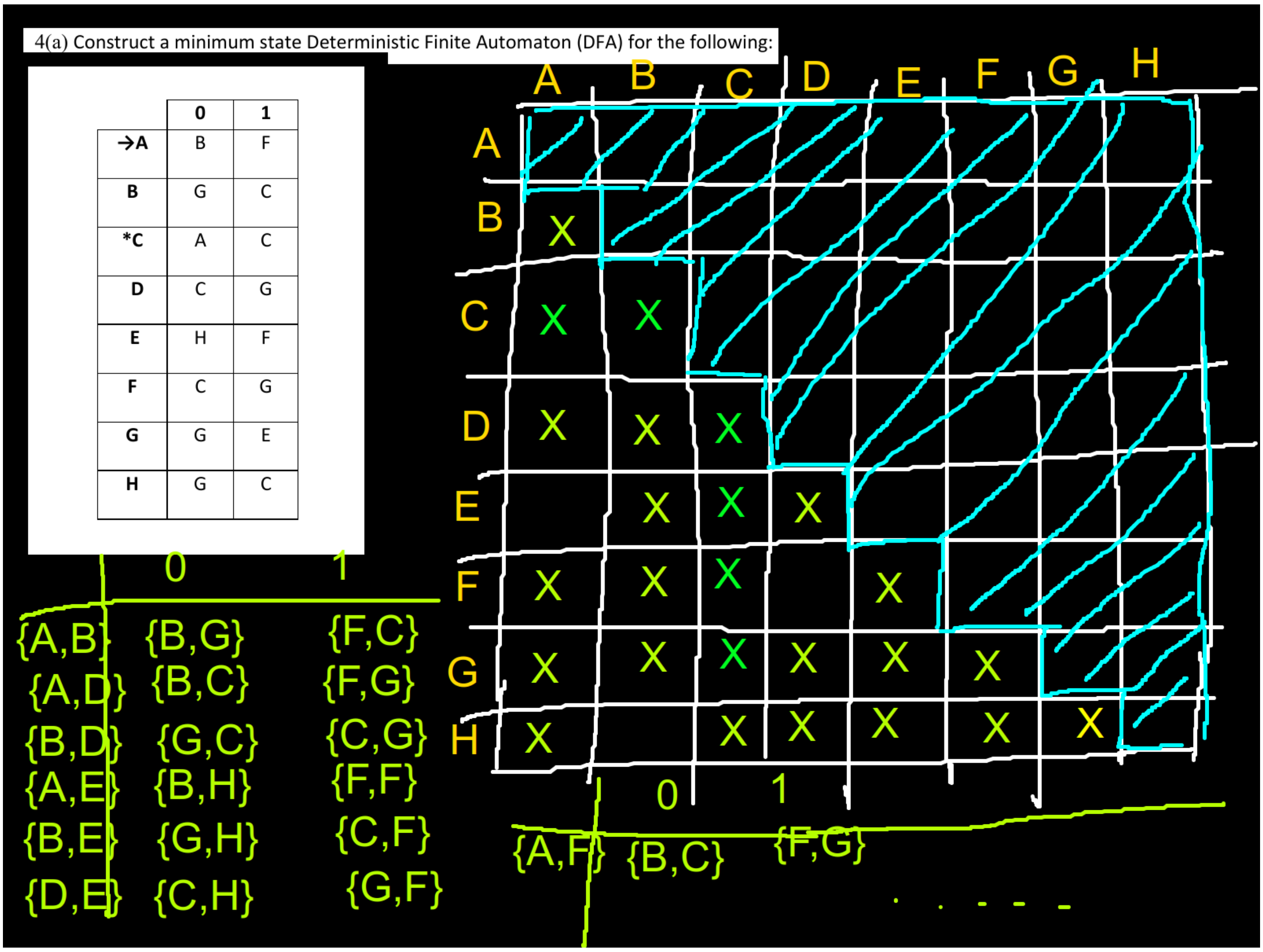
A 00

B 01

C 10

D 11

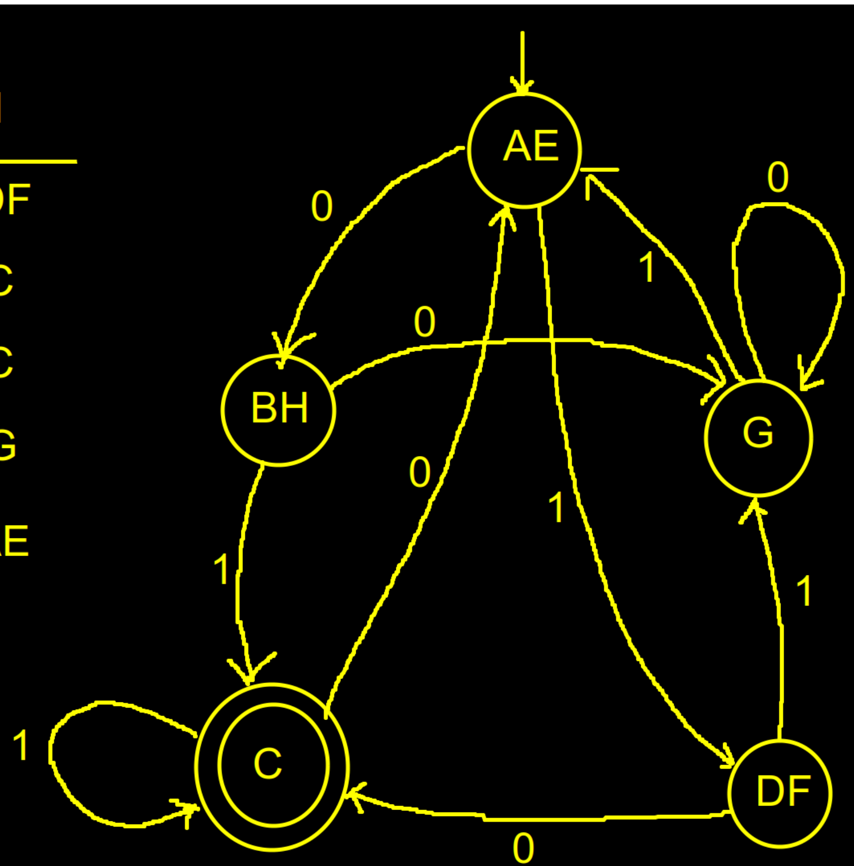
S -> 00A | 01B | 10C | 11D
A -> 00A | 0 | E
B -> 01B | 0 | E
C -> 10C | 1 | E
D -> 11D | 1 | E

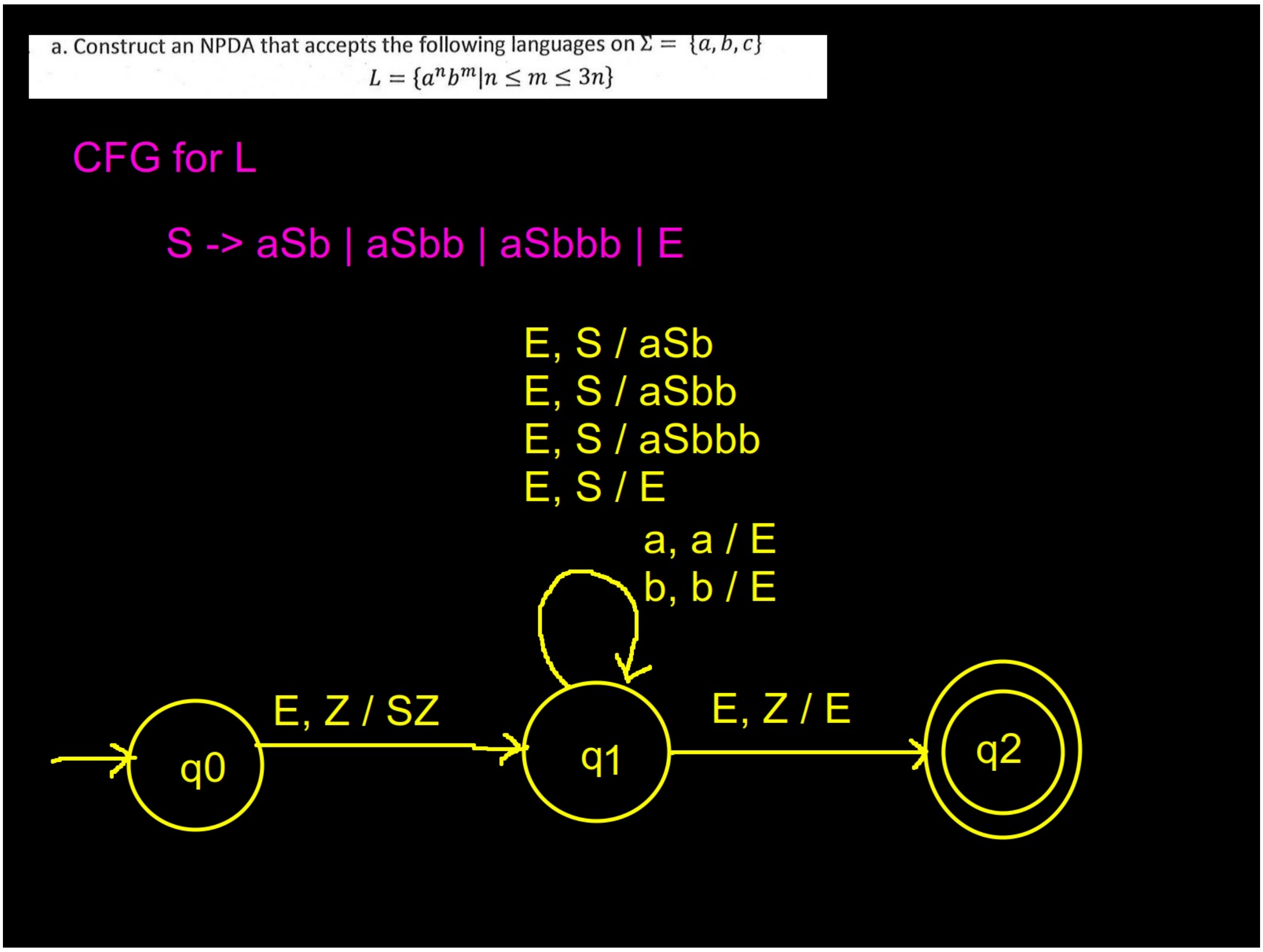


	0	1
→A	B	F
B	G	C
*C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

Minimised d:

	0	1
→AE	BH	DF
BH	G	C
*C	AE	C
DF	C	G
G	G	AE





(c) Using the Pumping lemma, prove that

L={$0^{n.m}$|n and m are prime numbers}

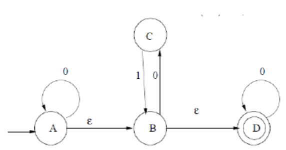
is not a CFL.

Ans.

- Assume L is a CFL.
- Thus, there exists a constant n (depending on L) satisfying the conditions of the Pumping Lemma for CFLs.
- Let p, q be **distinct** prime numbers $\geq$ n. We are guaranteed that such primes exist since there are infinitely many primes.
- Let z = $0^{p.q}$. Clearly, z is in L, and |z| = pq $\geq$ n.
- By the Pumping Lemma, we can break z = uvwxy such that vx $\neq \epsilon$ and |vwx| $\leq$ n.
- Let |vx| = m > 0.
- Let k = pq + 1. Consider the string t = uv^kwx^ky. By the Pumping Lemma, this string must be in L.
- Since the string consists of only 0's,

  $$|t| = |uwy| + k|vx| = pq - m + km = pq + (k - 1)m = pq(1 + m)$$
- Since, 1 + m > 1, |t| has **at least three factors p, q and (1 + m).** So, it cannot be the product of two primes.
- Hence t does not belong to L, which is a contradiction.
- So, our assumption was wrong and L is not a CFL.

(b) Given the NFA below for $0^*(01)^*0^*$, construct a Deterministic Finite Automaton (DFA)

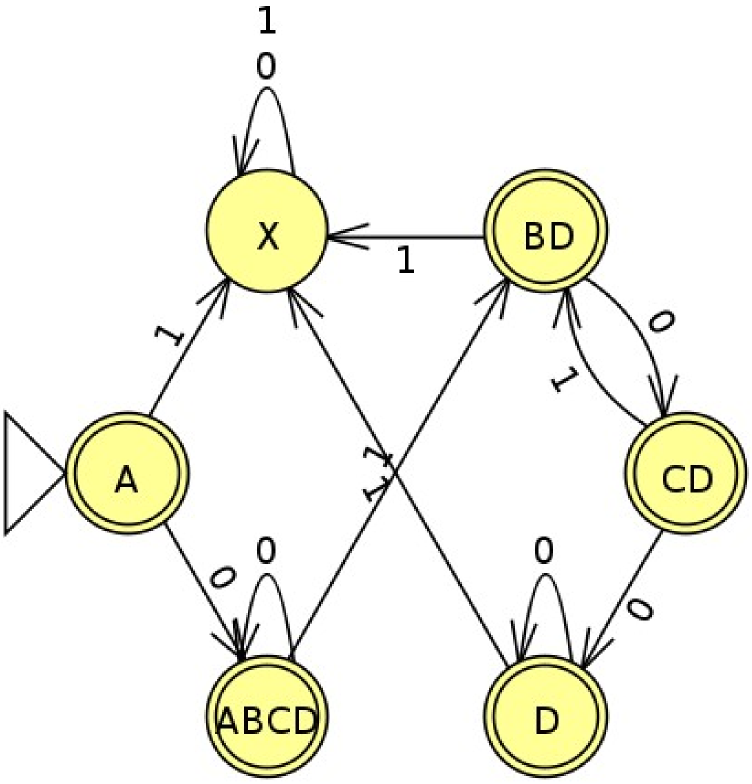


d_NFA

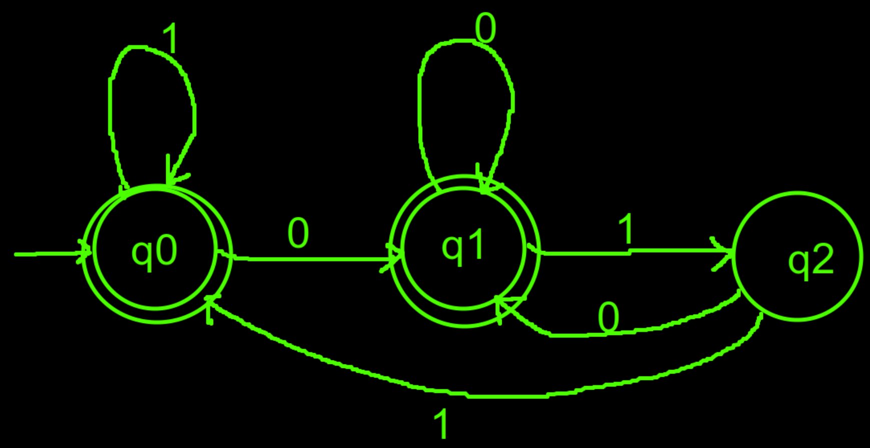
	E*	0	E*	E*	1	E*
→*A	{A, B, D}	{A,C,D}	{A,B,D,C}	{A,B,D}	{}	{}
*B	{B, D}	{C, D}	{C, D}	{B, D}	{}	{}
C	{C}	{}	{}	{C}	{B}	{B, D}
*D	{D}	{D}	{D}	{D}	{}	{}

Subset construction

	0	1
*A	ABCD	X
*ABCD	ABCD	BD
*BD	CD	X
*CD	D	BD
*D	D	X
X	X	X



(c) Give a Nondeterministic Finite Automaton (NFA) for the language of all strings over $\{0,1\}^*$ that do not end in 01.



5(a) How would you decide if a Regular language is empty?

Ans.

- If L is the language of a finite automaton
  - We check if there is any state **p** reachable from the start state such that p is a **final state.**
  - This can be done using a recursive **depth-first search algorithm** for graph reachability.
  - If **no such p** exists, L is empty. Otherwise L is non-empty.
- If L is given by a Regular Expression R.
  - We proceed by Induction.
  - **Basis:** ϕ is empty.
  - **Induction:**
    - $R = R_1 + R_2$ is empty iff R1 and R2 are both empty.
    - $R = R_1R_2$ is empty iff either R1 or R2 or both are empty.
    - $R = R_1*$ is never empty.
    - $R = (R_1)$ is empty iff R_1 is empty.

(b) Let L and M be two Regular languages and $[q_L,q_M]$ be a state of the product DFA produced out of the DFA for L and M. And also q_L and q_M are two states of the DFA for L and M respectively. How would you design the final/accept states of the product DFA to decide if L=M

Ans.

- We define a state (q_L, q_M) of the product DFA to be accepting iff **exactly one** of q_L and q_M is an accepting state.
- That is, **either q_L is final or q_M is final but not both**.
- If the language accepted by the above DFA is **empty**, we can conclude that L = M.

(c) Given **E=01*+10***, find $\mathbf{E^R}$ where R denotes Reversal of a Regular Expression.

Ans.

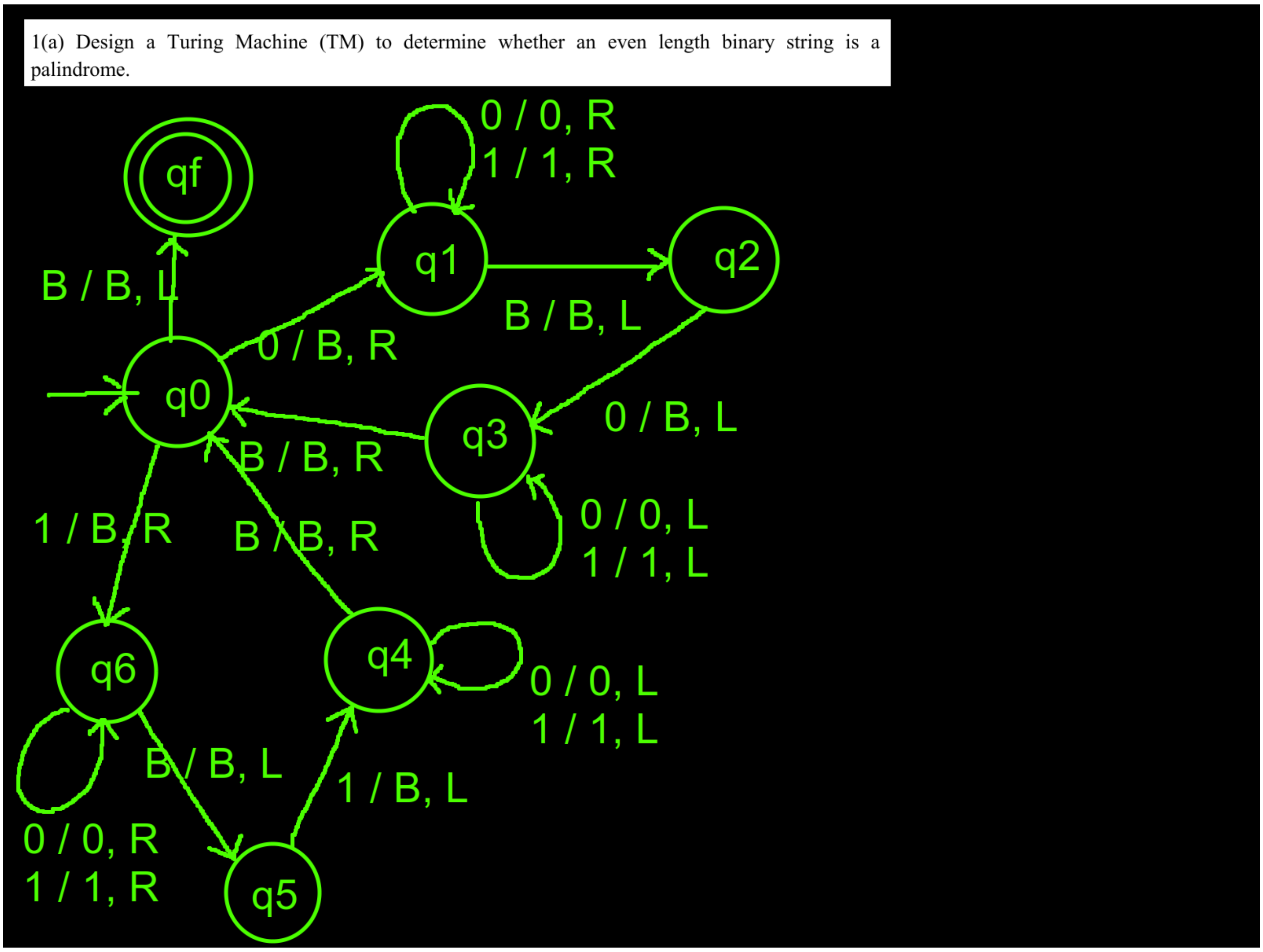
$E^R = (01* + 10*)^R = (01*)^R + (10*)^R = 1*0 + 0*1$

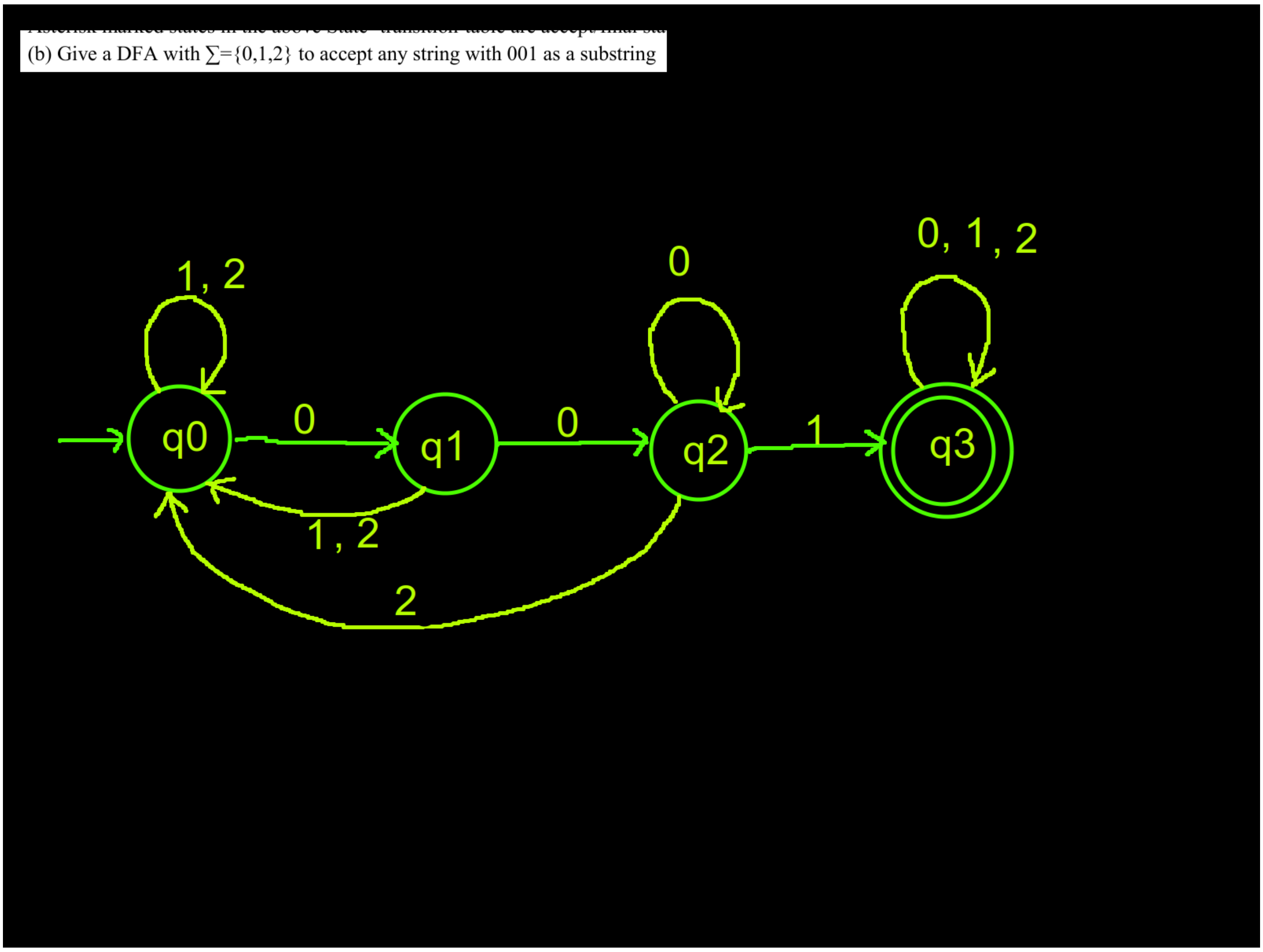
6. Let A be a Deterministic finite Automaton (DFA) with **n** states. Prove that if there is string of length at least n in L(A) then there is a string of length between n and 2n-1in L.

Ans.

- Given, there exists a string of length $\geq$ n in L(A).
- Let this string be w.
- If $n \leq |w| < 2n$, then the claim is trivially true.
- Let $|w| = m \geq 2n$.
- Let $w = a_1a_2...a_m$, where a_i is a symbol in Σ.
- Let p_0 be the start state. We define p_i to be the state $\delta(p_0, a_1a_2...a_i)$. That is p_i is the state of the DFA after processing the first i symbols of w.
- Clearly p_m is a final state, since w is in L.
- Now, the total number of states $p_0, p_1, ...\ p_m = m + 1 \geq 2n + 1 > n$.
- Since the DFA has only n states, by the Pigeonhole Principle, at least two of these states must be the same.
- Let p_i, p_j $(i < j)$ be two states such that $p_i = p_j$ and $(j – i)$ is the smallest possible.
- It is clear then that $p_i, p_{i+1}, ...\ p_{j-1}$ must be distinct states (otherwise we have a choice of i, j that gives a smaller $j – i$).
- Since the total number of states in the DFA = n, we conclude that $j – i \leq n$.

- Since $p_i = p_j$, we can say that the string $t = a_1a_2...a_ia_{j+1}a_{j+2}...a_m$ takes the start state of the DFA to the same final state p_m.
- $|t| = |m| - (j – i)$. Since $|m| \geq 2n$ and $0 < (j – i) \leq n$, we can say $n \leq |t| < |m|$.
- Thus, we have proved that **for every string w of length ≥ 2n in L, there exists a strictly shorter string t in L with |t| ≥ n.**
- We apply this process of shortening repeatedly starting from our initial string w, until we obtain a string with length in [n, 2n – 1], which terminates in a finite number of steps.





4. Assuming that L is a Regular language and h is a homomorphism on its alphabet, define h(L). 3

Ans.