Q1)

Consider the following two statements about regular languages:

- S1: Every infinite regular language contains an undecidable language as a subset.
- S2: Every finite language is regular.

Which one of the following choices is correct?

A) Only S1 is true
B) nly S2 is true
C) Both S1 and S2 are true
D) Neither S1 nor S2 is true

Q 2)

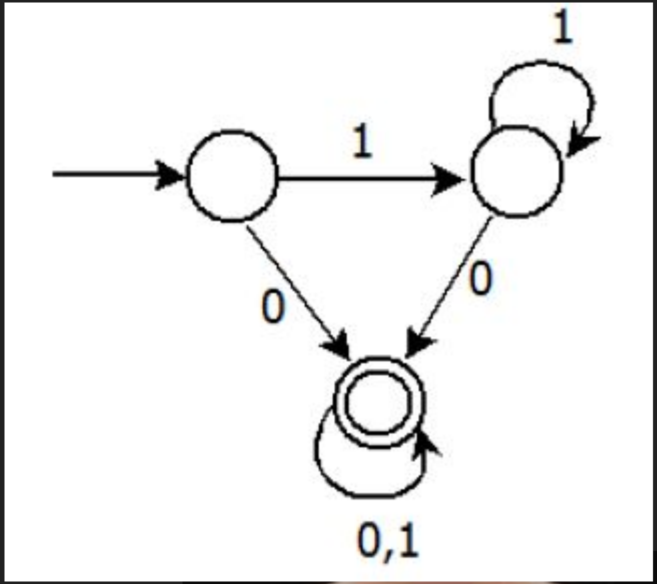
Which of the following are FALSE?

A) Complement of L(A) is regular

B) L(A) = L((11*0+0)(0 + 1)*0*1*)

C) For the language accepted by A, A is the minimal DFA

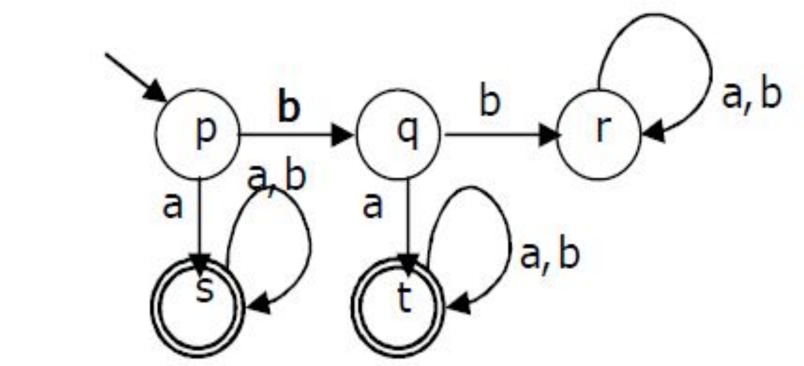
D) A accepts all strings over {0, 1} of length at least 2.



Q3)

A deterministic finite automata (DFA)D with alphabet {a,b} is given below

HOw many minimum states are there in the minimal DFA?



A) 4

B) 5

C) 3

D) 2

Q4)

Which one of the following is FALSE?

A) There is unique minimal DFA for every regular language

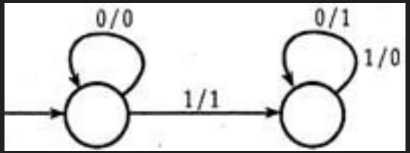
B) Every NFA can be converted to an equivalent PDA.

C) Complement of every context-free language is recursive.

D) Every nondeterministic PDA can be converted to an equivalent deterministic PDA.

Q5)

The following diagram represents a finite state machine which takes as input a binary number from the least significant bit Which one of the following is TRUE?



A) It computes 2's complement of the input number

B) It increments the input number

C) It computes 1's complement of the input number

D) None of these

Q6)

The smallest finite automata which accepts the language {x | length of x is divisible by 3}

as------------------states

Q 7)

Which one of the following regular expressions represents the set of all binary strings with an odd number of $1'$s?

A. $((0+1)^*1(0+1)^*1)^*10^*$
B. $(0^*10^*10^*)^*0^*1$
C. $10^*(0^*10^*10^*)^*$
D. None of these

Q 8)

Consider the following languages $L1 = \{ww | w \in \{a, b\}*\}$

$L2 = \{ww^R | w \in \{a, b\}*, w^R is\ the\ reverse\ of\ \ w\}$, $L3 = \{0^{2i} | i\ is\ an\ integer\}$

$L4 = \left\{0^{i^2} | i\ is\ an\ integer\right\}$ Which of the languages are regular?

A) Only L1 and L2

B) Only L2, L3 and L4

C) Only L3 and L4

D) Only L3

Q9)

Let L1, L2 be any two context-free languages and R be any regular language. Then which of the following is/are False? (I) $(L1)' \cup L2 \cup L1$ is context-free (II) $R' \cup L2$ is context-free (III) $R \cap L1 \cap L2$ is context-free (IV) $R \cap L2$ is context-free

A) I, II and IV only

B) I and III only

C) II and IV only

D) I only

Q10)

Which of the following regular expressions, each describing a language of binary numbers (MSB to LSB) that represents non-negative decimal values, does not include even values ?

A) 0*1+0*1*

B) 0*1*0+1*

C) 0*1*0*1+

D) 0+1*0*1*

Q 11)

Let L1 be regular language, L2 be a deterministic context free language and L3 a recursively enumerable language, but not recursive. Which one of the following statements is false?

A) L1 ∩ L2 is a deterministic CFL
B) L3 ∩ L1 is recursive
C) L1 ∪ L2 is context free
D) L1 ∩ L2 ∩ L3 is recursively enumerable

Q 12)

The number of strings of length 4 that are generated by the regular expression (0+1+|2+3+)*, where | is an alternation character and {+, *} are quantification characters, is:

A) 8

B) 9

C) 10

D) 12

Q13)

Which of the following languages is context-free?

A) $\{a^n b^n \mid n \geq 0\}$

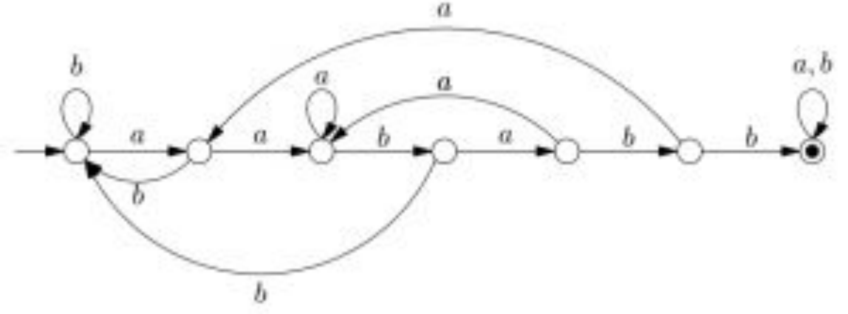
B) $\{a^n b^m c^n \mid n, m \geq 0\}$

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

D) $\{a^n \mid n \text{ is a prime number}\}$

Q14)

Consider the following Deterministic Finite Automata   Which of the following is true?



A) It only accepts strings with prefix as "aababb"

B) It only accepts strings with substring as "aababb"

C) It only accepts strings with suffix as "aababb"

D) None of the above.

Q 15)

Consider the following two statements:
I. If all states of an NFA are accepting states then the language accepted by the NFA is Σ^* .

II. There exists a regular language A such that for all languages B, A U B is regular.
Which one of the following is CORRECT?

A) Only I is true

B) Only II is true

C) Both I and II are true

D) Both I and II are false

Q 16)

Consider the Following regular expressions

$r1 = 1(0 + 1)^*$
$r2 = 1(1 + 0)^+$
$r3 = 11^*0$

WHich of the following is/are true?

A) $L(r1) \subseteq L(r2)$ and $L(r1) \subseteq L(r3)$

B) $L(r1) \supseteq L(r2)$ and $L(r2) \supseteq L(r3)$

C) $L(r1) \supseteq L(r2)$ and $L(r2) \subseteq L(r3)$

D) $L(r1) \supseteq L(r3)$ and $L(r2) \subseteq L(r1)$

Q 17)

Suppose M1 and M2 are two TM’s such that L(M1) = L(M2). Then

A) On every input on which M1 doesn’t halt, M2 doesn’t halt too.

B) On every i/p on which M1 halts, M2 halts too.

C) On every i/p which M1 accepts, M2 halts.

D) None of above

Q18)

If we consider an arbitrary NFA (non-deterministic finite automaton) with N states in total, the maximum number of states that are there in an equivalent DFA (minimised) is at least:

a. N!

b. 2N

c. 2^N

d. N^2

Q 19)

Consider the regular expression (0+1)(0+1)... (n-1) times. The minimum state finite automaton that recognizes the language represented by this regular expression contains:

A) n states
B) n+1 states
C) n+2 states
D) n-1 states

Q 20)

Let A and B be finite alphabets and let $\#$ be a symbol outside both A and B. Let f be a total function from A^* to B^*. We say f is *computable* if there exists a Turing machine M which given an input $x \in A^*$, always halts with $f(x)$ on its tape. Let L_f denote the language $\left\{x\#f(x) \mid x \in A^*\right\}$. Which of the following statements is true:

A. f is computable if and only if L_f is recursive.
B. f is computable if and only if L_f is recursively enumerable.
C. If f is computable then L_f is recursive, but not conversely.
D. If f is computable then L_f is recursively enumerable, but not conversely.

Q 21)

Let δ denote the transition function and α denoted the extended transition function of the ε-NFA whose transition table is given below:

δ	ϵ	a	b
$\rightarrow q_0$	$\{q_2\}$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_0\}$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\{q_2\}$

Which of the following option is correct?

A) α (q1,aba) is {q0, q2)
B) Null reachable states are {q0, q1, q2}
C) α (q3,bab) is {q0, q1, q2, q3}
D) None of these

Q 22)

Which of the following regular expressions, each describing a language of binary numbers (MSB to LSB) that represents non-negative decimal values, does not include even values ?

A) $0^*1^*+0^*1^*$

B) $0^*1^*0+1^*$

C) $0^*1^*0^*1+$

D) $0+1^*0^*1^*$

Q 23)

Let L1 be regular language, L2 be a deterministic context free language and L3 a recursively enumerable language, but not recursive. Which one of the following statements is false?

A) $L1 \cap L2$ is a deterministic CFL
B) $L3 \cap L1$ is recursive
C) $L1 \cup L2$ is context free
D) $L1 \cap L2 \cap L3$ is recursively enumerable

Q 24)

The number of states required by a Finite State Machine, to simulate the behavior of a computer with a memory capable of storing 'm' words, each of length 'n' bits is?

A) m x 2n

B) 2m+n

C) 2mn

D) m+n

Q 25)

Let Q be a context-free language, and P be a regular language such that Q is a subset of P. For example, let P be a language given by the regular expression a*b* and Q be a context-free language given by { anbn | n ϵ N }. Then which of the following is always regular?

a) P ∩ Q

b) P - Q

c) Σ* - P

d) Σ* - Q

Q 26)

Match the following

List - I		List - II	
(a)	$\{a^n b^n \mid n > 0\}$ is a deterministic context free language	(i)	but not recursive language
(b)	The complement of $\{a^n b^n a^n \mid n > 0\}$ is a context free language	(ii)	but not context free language
(c)	$\{a^n b^n a^n\}$ is context sensitive language	(iii)	but can not be accepted by a deterministic pushdown automation
(d)	L is a recursive language	(iv)	but not regular

Codes :

	(a)	**(b)**	**(c)**	**(d)**
(1)	(i)	(ii)	(iii)	(iv)
(2)	(i)	(ii)	(iv)	(iii)
(3)	(iv)	(iii)	(ii)	(i)
(4)	(iv)	(iii)	(i)	(ii)

A) 1

B) 2

C) 3

D) 4

Q 27)

Minimal deterministic finite automaton for the language $L = \{0^n \mid n \geq 0, n \neq 4\}$ will have how many final states?-------------------

A) 1

B) 2

C) 4

D) 5

Q 28)

The language $\{ww \mid w \in (0+1)^*\}$ is

A) not accepted by any Turing machine

B) accepted by some Turing machine, but by no pushdown automaton

C) accepted by some pushdown automaton, but not context-free

D) context-free, but not regular

Q 29)

Consider the following languages :

- $L1:=\{\langle M\rangle \mid M \text{ is a TM, and } M \text{ is the only TM that accepts } L(M)\}$.
- $L2:=\{\langle M\rangle \mid M \text{ is a TM, and } |M|<1000\}$.

Which of the above languages is Decidable?

A. Only L1
B. Only L2
C. Both
D. None

Q 30)

Consider the following instance of Post's Correspondence Problem:

Index	A	B
1	10	101
2	101	011
3	110	100

The above instance of Post's Correspondence Problem has :

A. Exactly one solution

B. At least one solution

C. Infinite solutions

D. No solution

Q 31)

The complement of the language L containing an equal number of a's , b's and c's is

A) Regular
B) Context free
C) Context sensitive but not context free
D) Recursive and not a CFL

Q 32)

The language $L= \{a^m b^n : m<2n<3m\}$ is —---------?

A) Context free language
B) Regular
C) Context sensitive
D) Recursive

Q 33)

Consider the following statements. I. If $L1 \cup L2$ is regular, then both L1 and L2 must be regular. II. The class of regular languages is closed under infinite union. Which of the above statements is/are TRUE ?

A) Ⅰ only

B) Ⅱ only

C) Both Ⅰ and Ⅱ

D) Neither I nor II

Q 34)

Which of the following pairs have DIFFERENT expressive power?

A) Deterministic finite automata (DFA) and Non-deterministic finite automata (NFA)
B) Deterministic push down automata (DPDA) and Non-deterministic push down automata (NPDA)
C) Deterministic single tape Turing machine and Non-deterministic single tape Turing machine
D) Single tape Turing machine and multi-tape Turing machine

Q 35)

regular expression for the set of binary strings where every 0 is immediately followed by exactly k 1's and preceded by at least k 1's (k is a fixed integer)

A) $1^*1^k(01^k)^*+1^*$

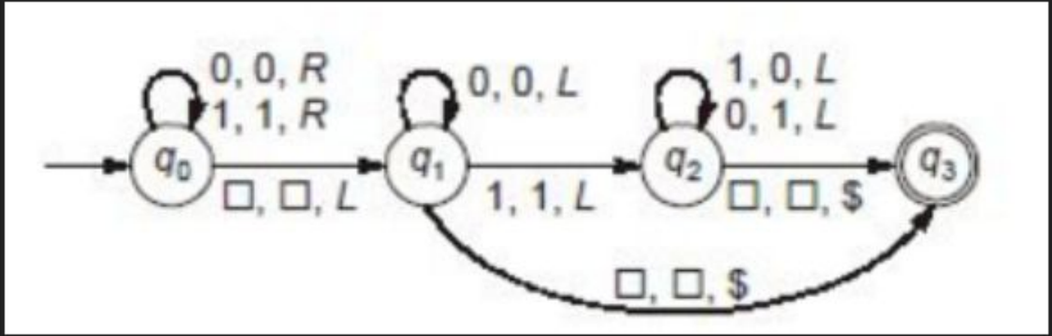
B) $1^k1^k(01^k)^*+1^*$

C) $1^*1^*(01^k)^*+1^*$

D) $1^*1^k(01^*)^*+1^k$

Q 36)

Consider the following turning machine (where,,$ is represent accept the string).



If the string is 01010 then what will be the output?

A) 10100
B) 10101
C) 10110
D) 10011

Q 37)

L1 is a recursively enumerable language over Σ. An algorithm A effectively enumerates its words as w1,w2,w3,…. Define another language L2 over $\Sigma \cup \{\#\}$ as $\{Wi\#Wj \mid Wi, Wj \in L1, i<j\}$. Here # is new symbol. Consider the following assertions.

- S1: L1 recursive implies L2 is recursive
- S2:L2 is recursive implies L1 is recursive

Which of the following statements is true?

A. Both S1 and S2 are true.
B. S1 is true but S2 is not necessarily true.
C. S2 is true but S1 is not necessarily true.
D. Neither is necessarily true.

Q 38)

For a string w, we define w^R to be the reverse of w. For example, if $w = 01101$ then $w^R = 10110$.

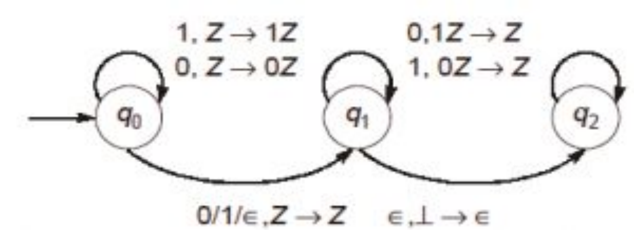
Which of the following languages is/are context-free?

A. $\{wxw^Rx^R \mid w, x \in \{0,1\}^*\}$
B. $\{ww^Rxx^R \mid w, x \in \{0,1\}^*\}$
C. $\{wxw^R \mid w, x \in \{0,1\}^*\}$
D. $\{wxx^Rw^R \mid w, x \in \{0,1\}^*\}$

ept

Q 39)

Consider the NPDA ⟨Q = {q0, q1, q2}, Σ = {0, 1}, Γ = {0, 1, ⊥}, δ, q0, ⊥, F = {q2}⟩, where (as per usual convention) Q is the set of states, Σ is the input alphabet, Γ is stack alphabet, δ is the state transition function, q0 is the initial state, ⊥ is the initial stack symbol, and F is the set of accepting states the state transition is as follows



Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automaton?

a) 10110
b) 10010
c) 01010

Q 40)

Which of the problems are unsolvable?

a) Halting problem

b) Boolean Satisfiability problem

c) Halting problem & Boolean Satisfiability problem

d) None of the mentioned

Q 41)

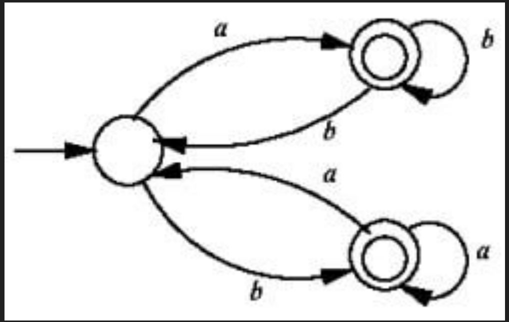
For any two languages L_1 and L_2 such that L_1 is context-free and L_2 is recursively enumerable but not recursive, which of the following is/are necessarily true?

I. $\bar{L}_1$ (Complement of L_1) is recursive
II. $\bar{L}_2$ (Complement of L_2) is recursive
III. $\bar{L}_1$ is context-free
IV. $\bar{L}_1 \cup L_2$ is recursively enumerable

A. I only
B. III only
C. III and IV only
D. I and IV only

Q 42)

Which of the following regular expression represents the following finite automaton?



A) ab*bab* + ba*aba*

B) (ab*b)*ab* + (ba*a)*ba*

C) (ab*b + ba*a)*(a* + b*)

D) (ba*a + ab*b)*(ab* + ba*)

Q 43)

Which of the following statements is/are FALSE?

1. For every non-deterministic Turing machine, there exists an equivalent deterministic Turing machine.
2. Turing recognizable languages are closed under union and complementation.
3. Turing decidable languages are closed under intersection and complementation.
4. Turing recognizable languages are closed under union and intersection.

A. 1 and 4 only
B. 1 and 3 only
C. 2 only
D. 3 only

Q44)

For $\Sigma=\{a,b\}$, let us consider the regular language $L=\{x \mid x=a^{2+3k}$ or $x=b^{10+12k}, k\geq 0\}$ what is the minimum pumping length of the given language?

A) 12

B) 24

C) 22

D) 5

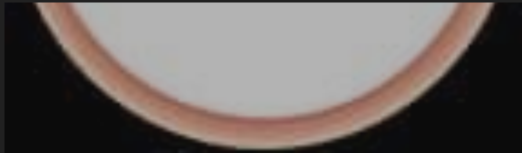
Q 45)

Consider the following languages:

I. $\{a^m b^n c^p d^q \mid m + p = n + q, \text{ where } m, n, p, q \geq 0\}$
II. $\{a^m b^n c^p d^q \mid m = n \text{ and } p = q, \text{ where } m, n, p, q \geq 0\}$
III. $\{a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q, \text{ where } m, n, p, q \geq 0\}$
IV. $\{a^m b^n c^p d^q \mid mn = p + q, \text{ where } m, n, p, q \geq 0\}$

Which of the above languages are context-free?

A. I and IV only
B. I and II only
C. II and III only
D. II and IV only



Q 46)

Let L(R) be the language represented by regular expression R. Let L(G) be the language generated by a context free grammar G. Let L (M) be the language accepted by a Turning machine M. Which of the following decision problems are undecidable ?

I. Given a regular expression R and a string w, is $w \in L(R)$?
II. Given a context-free grammar G, $L(G)=\phi$?
III. Given a context-free grammar G, is $L(G)=\Sigma^*$ for some alphabet Σ?
IV. Given a Turning machine M and a string w, is $w \in L(M)$?

A) I and IV only
B) II and III only
C) II, III and IV only
D) III and IV only

Q 47)

Let δ denote the transition function and δ denote the extended transition function of the ϵ-NFA whose transition table is given below:

δ	ϵ	a	b
$\rightarrow q_0$	$\{q_2\}$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_0\}$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\{q_2\}$

Then $\hat{\delta}(q_2, aba)$ is

A. $\emptyset$

B. $\{q_0, q_1, q_3\}$

C. $\{q_0, q_1, q_2\}$

D. $\{q_0, q_2, q_3\}$

Q 48)

Consider the following problem X.

Given a Turing machine M over the input alphabet Σ, any state q of M and a word $w \in \Sigma^*$, does the computation of M on w visit the state of q?

Which of the following statements about X is correct?

- A. X is decidable
- B. X is undecidable but partially decidable
- C. X is undecidable and not even partially decidable
- D. X is not a decision problem

Q 49)

Consider the following Recursive languages state which are true:

A. A proper superset of context free languages.
B. Always recognizable by pushdown automata.
C. Also called type 0 languages.
D. Recognizable by Turing machines.

Q 50)

Which of the following problems are undecidable?

A. Membership problem in context-free languages.
B. Whether a given context-free language is regular.
C. Whether a finite state automation halts on all inputs.
D. Membership problem for type 0 languages.

Q 51)

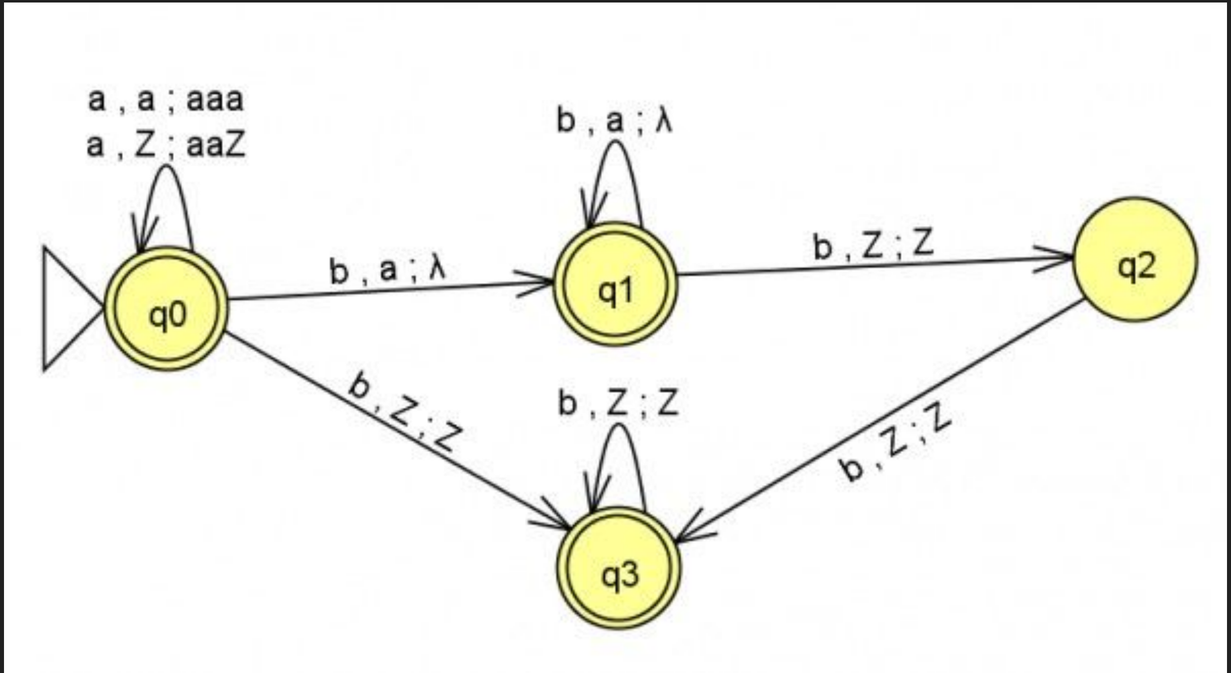
Which of the following conversions is not possible (algorithmically)?

A. Regular grammar to context free grammar
B. Non-deterministic FSA to deterministic FSA
C. Non-deterministic PDA to deterministic PDA
D. Non-deterministic Turing machine to deterministic Turing machine

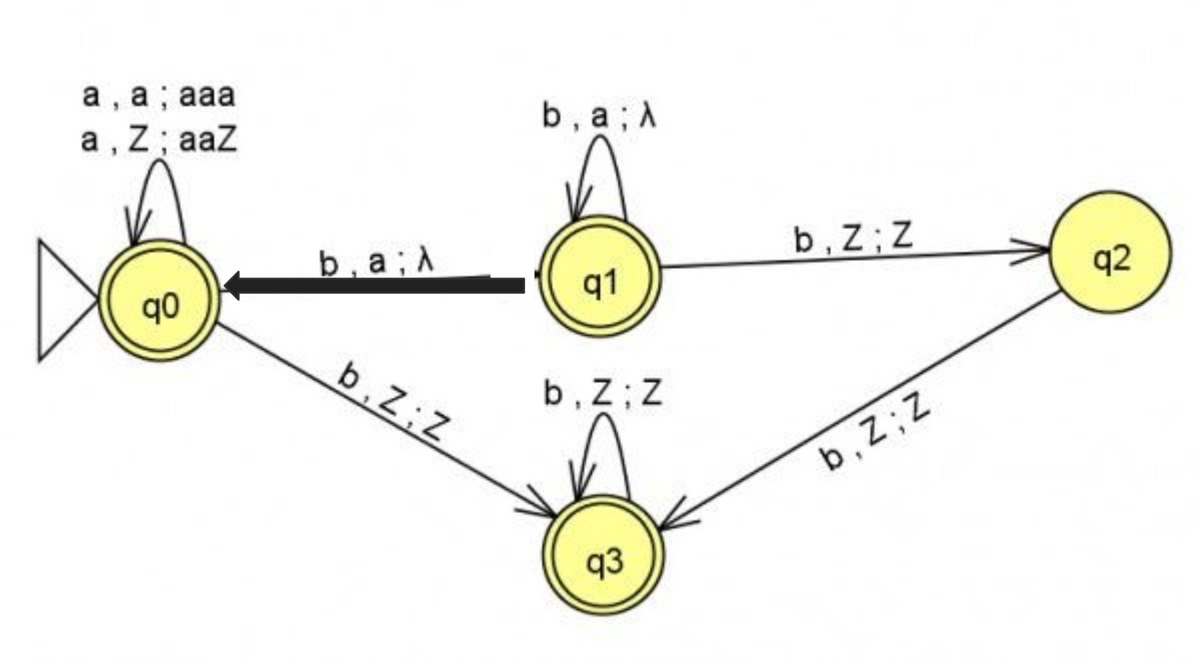
Q 52)

Find the pda for the given language L={a^i b^j | $i \neq 2j+1$}

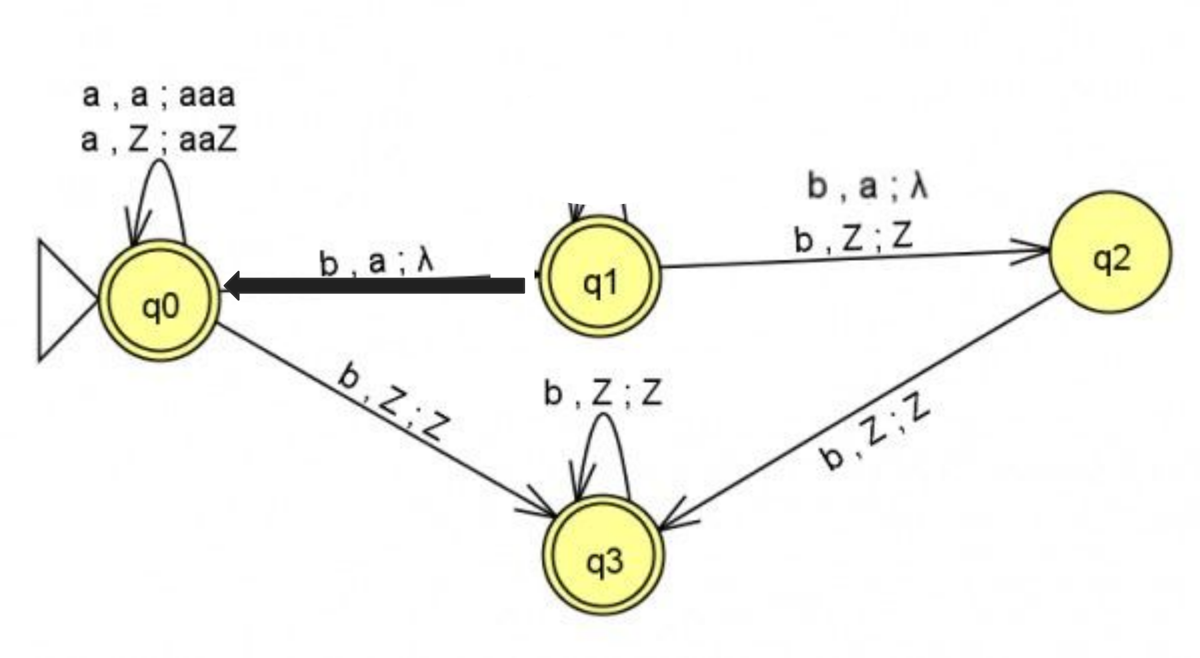
A)



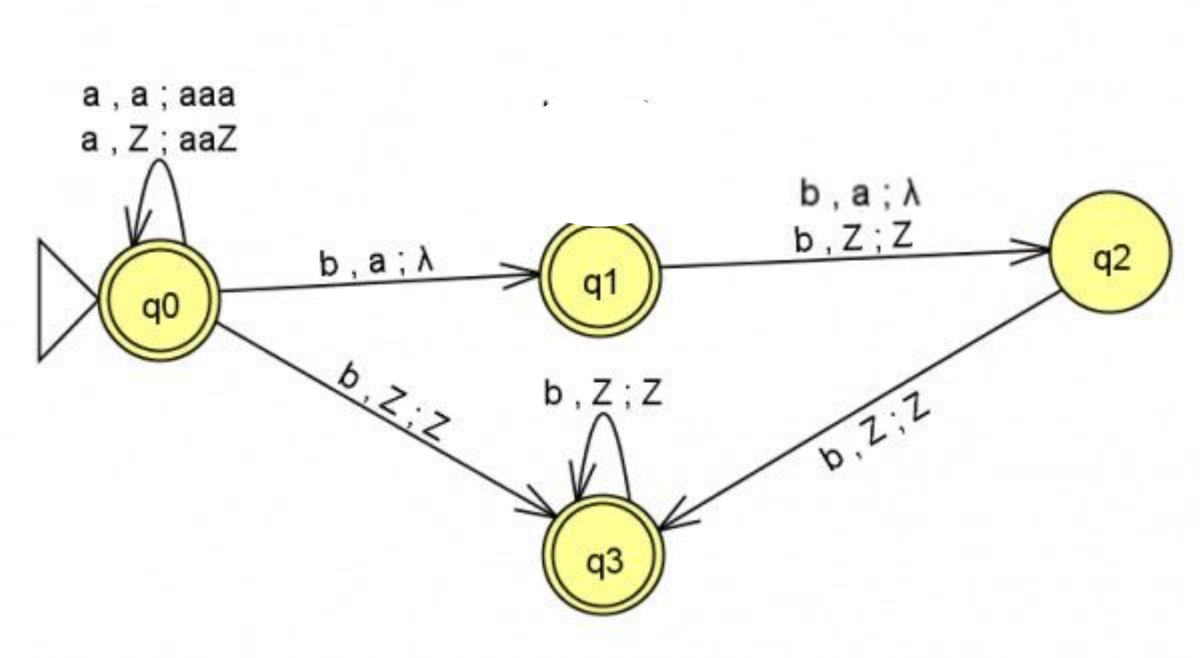
B)



c)



D)



Q 53)

Consider a push down automata (PDA) below which runs over the input alphabet (a, b). It has the stack alphabet $\{z_0, X\}$ where z_0 is the bottom of stack marker. The set of states of PDA is $\{q_0, q_1\}$ where q_0 is the start state.

$\delta\{q_0, b, z_0\} = \{(q_0, Xz_0)\}$
$\delta\{q_0, b, X\} = \{(q_0, XX)\}$
$\delta\{q_0, a, X\} = \{(q_1, X)\}$
$\delta\{q_0, \in, z_0\} = \{(q_0, \in)\}$
$\delta\{q_1, b, X\} = \{(q_1, \in)\}$
$\delta\{q_1, a, z_0\} = \{(q_0, z_0)\}$

The language accepted by PDA is

A) $\{(b^n a b^n a)^m | n, m \geq 0\}$

B) $\{(b^n a b^n a)^m | n, m \geq 0\} \cup \{b^n | n \geq 0\}$

C) $\{(b^n a b^n)^m a | n, m \geq 0\}$

D) None

Q 54)

Consider the following statement

S: $\{a^n b^{n+k} \mid n>=0, k>=1\} U \{a^{n+k} b^n \mid n>=0, k>=3\}$?

A) Regular
B) Deterministic context free
C) Context free
D) Context sensitive

Q 55)

Let $r = 1(1+0)^*$, $s = 11^*0$ and $t = 1^*0$ be three regular expressions. Which one of the following is true?

A. $L(s) \subseteq L(r)$ and $L(s) \subseteq L(t)$

B. $L(r) \subseteq L(s)$ and $L(s) \subseteq L(t)$

C. $L(s) \subseteq L(t)$ and $L(s) \subseteq L(r)$

D. $L(t) \subseteq L(s)$ and $L(s) \subseteq L(r)$

E. None of the above

Q 56)

Let be the encoding of a Turing machine as a string over $\sum$= {0, 1}. Let L = { |M is a Turing machine that accepts a string of length 2014 }. Then, L is

(A) decidable and recursively enumerable

(B) undecidable but recursively enumerable

(C) undecidable and not recursively enumerable

(D) decidable but not recursively enumerable

Q 57)

A single tape Turing Machine M has two states q0 and q1, of which q0 is the starting state. The tape alphabet of M is {0, 1, B} and its input alphabet is {0, 1}. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

	0	1	B
q0	q1,1,R	q1,1,R	Halt
q1	q1,1,R	q0,L,R	q0,B,L

The table is interpreted as illustrated below. The entry (q1, 1, R) in row q0 and column 1 signifies that if M is in state q0 and reads 1 on the current tape square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q1. Which of the following statements is true about M?

1. M does not halt on any string in (0 + 1)+
2. M does not halt on any string in (00 + 1)*
3. M halts on all string ending in a 0
4. M halts on all string ending in a 1

Q 58)

Set of binary strings represents Fibonacci Sequence over input alphabet {0,1} ?

A) Regular

B) Context free

C) Context sensitive

D) None of these

Q 59)

Which of the following is true regarding a Pushdown Automaton (PDA)?

A) PDA can recognize only regular languages.

B) PDA can recognize only context-free languages.

C) PDA can recognize both regular and context-free languages.

D) PDA can recognize any language.

Q 60)

Consider the following decision problems:

$(P1)$: Does a given finite state machine accept a given string?

$(P2)$: Does a given context free grammar generate an infinite number of strings?

Which of the following statements is true?

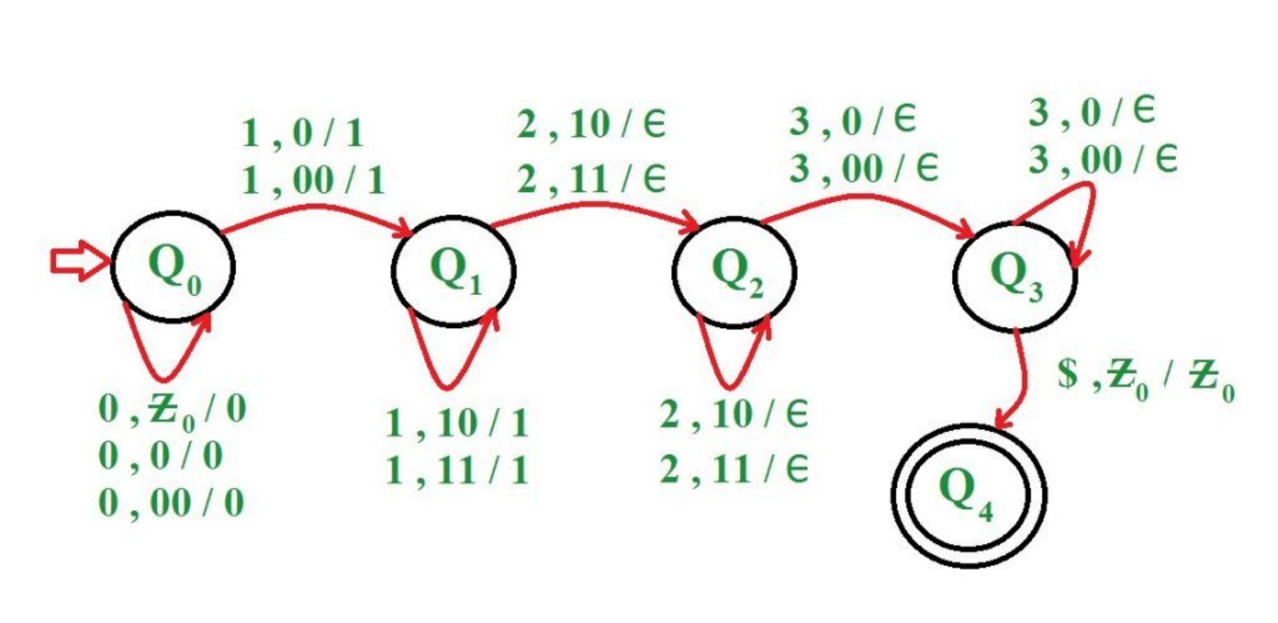
A. Both $(P1)$ and $(P2)$ are decidable
B. Neither $(P1)$ nor $(P2)$ is decidable
C. Only $(P1)$ is decidable
D. Only $(P2)$ is decidable

Q 61)

Consider the following PDA

Language generated by

The PDA is?



A) $L = \{0^n1^m2^m3^m \mid n>=1, m>=1\}$

B) $L = \{0^n1^m2^m3^n \mid n>=1, m>=1\}$

C) $L = \{0^n1^m2^n3^n \mid n>=1, m>=1\}$

D) $L = \{0^n1^n2^m3^n \mid n>=1, m>=1\}$

Q 62)

Let S and T be languages over $\Sigma = \{a, b\}$ represented by the regular expressions $(a + b^*)^*$ and $(a + b)^*$, respectively. Which of the following is true?

A. $S \subset T$
B. $T \subset S$
C. $S = T$
D. $S \cap T = \phi$

Q 63)

The ratio of number of input to the number of output in a mealy machine can be given as:

a) 1

b) n: n+1

c) n+1: n

d) none of the mentioned

Q 64)

The minimum number of states required to recognize an octal number divisible by 3 are/is

a)1

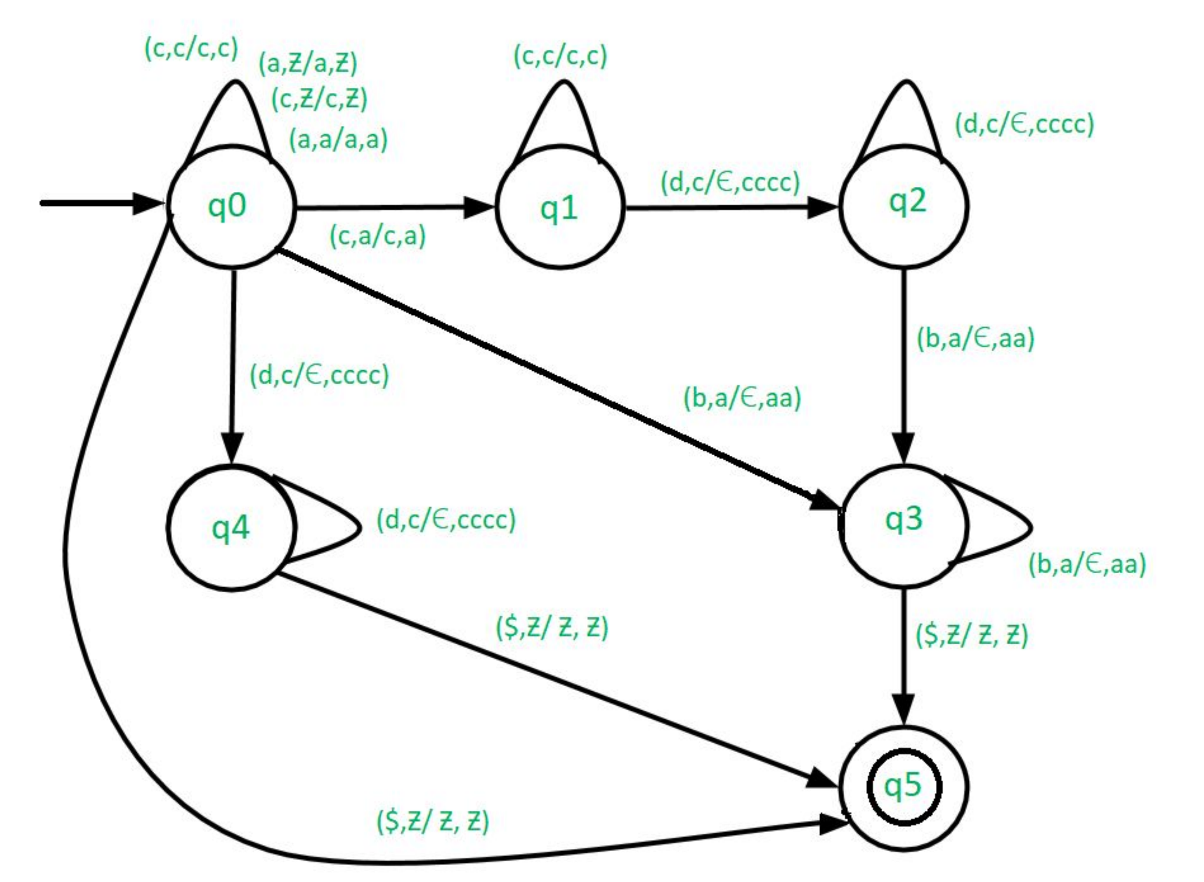
b)3

c)5

d)7

Q 65)

The Language accepted by given PDA



A) $\{a^{(2*m)}c(^{4*n)}d^{m}b^{m} \mid m,n \geq 0\}$

B) $\{a^{(2*m)}c(^{4*n)}d^{n}b^{m} \mid m,n \geq 1\}$

C) $\{a^{(4*m)}c(^{2*n)}d^{n}b^{m} \mid m,n \geq 0\}$

D) $\{a^{(2*m)}c(^{4*n)}d^{n}b^{m} \mid m,n \geq 0\}$

S 1) C

Explanation:

S_1: Every infinite regular language contains an undecidable language as a subset.

Cantor's theorem says that If S is any set then|s|<|p(s)|, where p(s) is the power set of S.So, from this we know that If we have any infinite set S then P(S) is definitely Uncountable (Because remember that a set A is countable if and only if |A|≤|N|, where N is the set of natural numbers).From this we can say that if S is countably infinite set, then P(S) is uncountable.

We know that Every language is countable. (Every language L is a subset of Σ∗, and Σ∗ is countable and subset of countable set is countable)If we have any infinite language L, then it means that we have uncountably many subsets of S. But we know that set of all RE languages is countable. So, due to this we have a subset of L which is Not RE. So, the following statements are true :

1. Every infinite regular language L has a subset S which is undecidable.
2. Every infinite regular language L has a subset S which is unrecognizable.
3. Every infinite language L has a subset S which is undecidable.
4. Every infinite language L has a subset S which is unrecognizable.

- Here, (2) implies (1) and (4) implies 3) as undecidable set is a proper superset of unrecognizable set. (By undecidable set, I mean Set of all undecidable languages. Similarly, for unrecognizable set.

S_2:

S_2 : **Every finite language is regular.**

It is true.

Proof:

Assume that we have a finite language $L = \{w_1, w_2, w_3, \ldots, w_n\}$

For L **we can write down:**

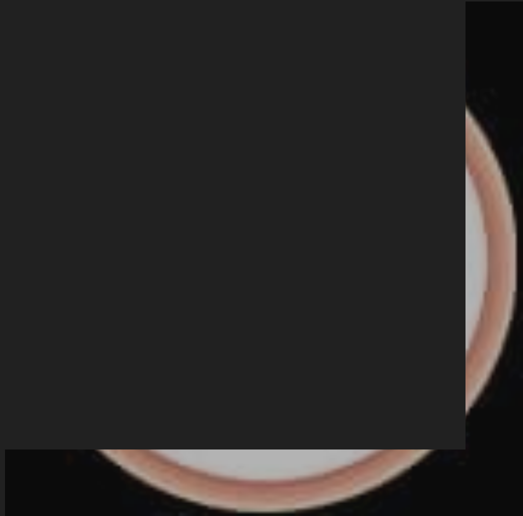
Regular grammar:

- $S \rightarrow w_1 \mid w_2 \mid w_3 \mid \ldots \mid w_n$

Regular expression:

- $w_1 + w_2 + w_3 + \ldots + w_n$

Hence, L **is regular.**



S2: C, D

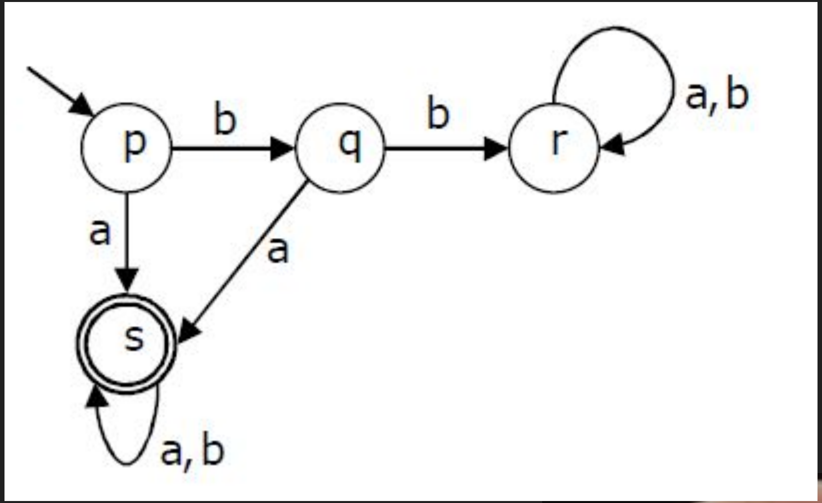
Explanation:

1 is true. L(A) is regular, its complement would also be regular. A regular language is also context free. 2 is true. 3 is false, the DFA can be minimized to two states. Where the second state is final state and we reach second state after a 0. 4 is clearly false as the DFA accepts a single 0.

S3: A

Explanation:

The minimal dfa of the given dfa is



Hence 4 states

S4: D

<u>Explanation:</u>

Power of Deterministic PDA is not same as the power of Non-deterministic PDA. Deterministic PDA cannot handle languages or grammars with ambiguity, but NDPDA can handle languages with ambiguity and any context-free grammar. So every non-deterministic PDA can not be converted to an equivalent deterministic PDA.

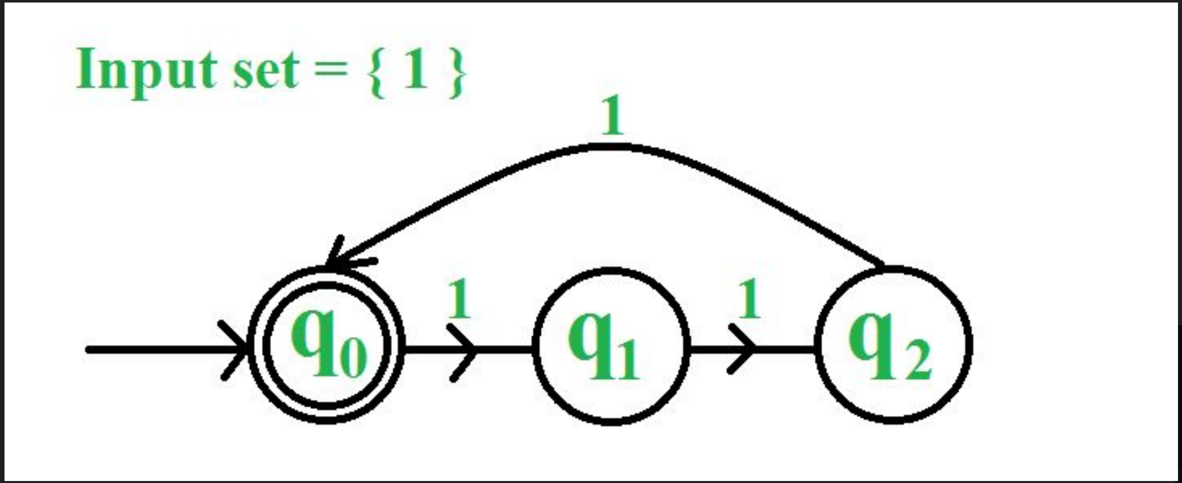
S5: A

Explanation:

The given finite state machine takes a binary number from LSB as input.
The given FSM remains unchanged till first ‘1’ . After that it takes 1’s complement of rest of the input string.
We assume the input string to be ‘110010’ . Thus, according to the FSM, output is ‘001110’ .
2’s complement of ‘110010’ = 1’s complement of ‘110010’ + 1 = 001101 + 1 = 001110 Thus, the FSM computes 2’s complement of the input string.

S6: 3

Explanation:



Thus, we require 3 states.

S7: D

Explanation:

Regular expression in option A cannot generate 001

Regular expression in option B cannot generate 100

Regular expression in option C cannot generate 001

Hence D is the answer

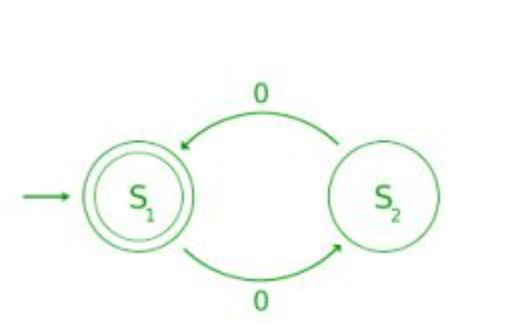
S8:

Explanation:

A language is known as regular language if there exists a finite automaton (no matter whether it is deterministic or non-deterministic) which recognizes it. So if for a given language, we can come up with an finite automaton, we can say that the language is regular. But sometimes, it is not quite obvious to design an automaton corresponding to a given language but it surely exists. In that case, we should not start thinking that the given language is not regular. We should use pumping lemma to decide whether the given language is regular or not. According to pumping lemma, “Suppose L is a regular language, then there exists a $l \geq 1$ such that for all string $s \in L$, where $|s| \geq l$, we can always split s (there exists at least one such splitting) in such a way that s can be written as xyz with $|xy| \leq l$ and $y \neq \varepsilon$ and for all $i \geq 0$, $xy^iz \in L$”. l is known as pumping length. Let’s rephrase the given Lemma for non regular languages. Suppose L is a language, if for all $l \geq 1$ there exist a string $s \in L$ with $|s| \geq l$ such that for all splitting (there doesn’t exists a single splitting which doesn’t follow this rule) of s in form of xyz such that $|xy| \geq l$ and $y \neq \varepsilon$, there exists an $i \geq 0$ such that $xy^iz \notin L$, then L is not regular. Notice that here we stress on finding such s if we want to prove that a language is not regular. Choice of the Questions: (a) In choice 1, Lets first consider w being of length n and containing only a. In this case the language represents a^na^n.

[Conti..]

The length of the string represented by language should be Even. Consider l = n, then xyz = anan with xy = a^n. lets assume y = a, then consider the membership of xy^iz with i = 0. This will simply be of odd length which doesn't belongs to L. So L is not regular. To discuss it in more detail, let's consider another example. Suppose w = apb, then string formed by L will be apbapb which is of length 2p + 2. Assume l = p, then xy = ap. suppose y = a, then consider the membership of xyiz with i = 0. This certainly doesn't belongs to L. So L is not regular. (b) In choice 2, The first example will work as above. In the second example, the string will be a^pbba^p, and there will be no changes in process for proving it to be non regular. (c) In choice 3, Assuming that we are considering integer from 0 and 02*n results in empty string, Which is also accepted, We can simply construct a DFA as given below. It simply accepts a string if it is either empty or contain even number of zeros. So the language is regular.



(d) In choice 4, We can simply assume that the pumping length l =i2/2. Now consider the xy = 0l with y = 0, Now if we check the membership of xy2z, we can find that this will represent 0i^2+1, and corresponding to which there exists no j such that j2 = i2 + 1 where i and j are integer except j = 1 and i = 0. But since i can't be zero. In Short, using pumping lemma, we can generate 0i^2+1 as well as 0^{i^2}-1, which won't be available in L. So L is not regular.

S9: B

Explanation:

Context free languages are not closed under complementary and intersection properties. Therefore, statements (I), (III) are false

S10: C

Explanation:

Regular expression for binary numbers that represents non negative odd numbers: It's LSB must be 1

S11: A

Explanation:

(A) This statement is true because deterministic context free languages are closed under intersection with regular languages.

(B) This statement is false because L1 is recursive and every recursive language is decidable. L3 is recursively enumerable but not recursive. So, L3 is undecidable. Intersection of recursive language and recursive enumerable language is recursively enumerable .

(C) This statement is true because L1 is regular. Every regular language is also a context free languages. L2 is a deterministic context free language and every DCFL is also a context free languages. Every context free language is closed under Union.

(D) This statement is true because L1 is regular hence it is also recursively enumerable. L2 is deterministic context free language so, it is also recursively enumerable . Recursively enumerable languages are closed under intersection.

Thus, problem mentioned in option (A) is undecidable.

S12: C

Explanation:

Possible strings of length 4 are: 0001, 0111, 0011, 0101, 0123, 2323, 2333, 2223, 2233, 2301. Total 10 strings are possible. So, option (C) is correct.

S13: A, B

Explanation:

A) $\{a^nb^n \mid n \geq 0\}$ —-------->equal no of a`s and b`s hence context free

B) $\{a^nb^mc^n \mid n, m \geq 0\}$ —-------->equal no of a`s and c`s and any number of b`s hence context free

C) $\{a^nb^nc^n \mid n \geq 0\}$ —------->more than one comparision exist hence not CFL

D) $\{a^n \mid n \text{ is a prime number}\}$ —--------->no regular repeatetion hence not CGL

S14: B

Explanation:

To reach the accepting state, any string will have to go through edges having aababb as labels in order. Though it might not be a continuous substring, but it sure will be a substring. There might be some cases where same substring always exists as a prefix or suffix for some DFA, but in this situation we don’t have to consider those cases, given this question has single choice answer. − > O − a− > O − a− > O − b− > O − a− > O − b− > O − b− > O Hence, correct answer should be B.

S15: B

Explanation:

Statement I : False, Since there is no mention of transition between states. There may be a case, where between two states there is no transition defined.
Statement II: True, Since any Complete language (i.e., A = sigma*) is regular and its intersection with any other language is Φ. Thus A U B is regular.

S16: B

Explanation:

Clearly r1 is a superset of both r2 and r3 as string 1 can not be generated by r2 and r3. r2 is a superset of r3 as string 11 is not present in L(r3) but in L(r2).

S17: C

Explanation:

M2 accepts strings in L(M1) so definitely it will halt. Other 2 are not guaranteed as Turing machine M does not accept w if it either rejects w by halting to a reject state or loops infinitely on w.

S18: C

Explanation:

For an arbitrary NFA with N states, the maximum number of states in an equivalent minimized DFA is 2^N

S19: D

Explanation:

To ensure the given string we need n-1 states

S20: A

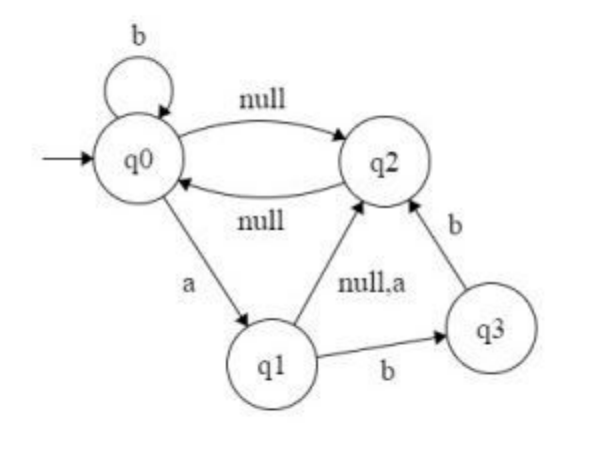
Explanation:

Since x belongs to A* is a total function, therefore, every alphabet in x will yield some alphabet from B*(in simple words f(x)) if given to a Turing machine.
The question itself says that f is computable if there exists a Turing machine which always halts with output f(x). If any Turing machine has to be always halting that means the language accepted by the Turing machine must be recursive.
In other words A) f is computable if and only if L(f) is recursive.

S21: C

Explanation:

Therefore, α (q1,aba) is {q0, q2, q3}, null reachable states are {q0, q2} and α (q3,bab) is {q0, q1, q2, q3}. Only option (C) is correct.



S22: C

Explanation:

Regular expression for binary numbers that represents non negative odd numbers: It's LSB must be 1. So (C) option is correct

S23: B

Explanation:

(A) This statement is true because deterministic context free languages are closed under intersection with regular languages.
(B) This statement is false because L1 is recursive and every recursive language is decidable. L3 is recursively enumerable but not recursive. So, L3 is undecidable. Intersection of recursive language and recursive enumerable language is recursively enumerable .
(C) This statement is true because L1 is regular. Every regular language is also a context free languages. L2 is a deterministic context free language and every DCFL is also a context free languages. Every context free language is closed under Union.
(D) This statement is true because L1 is regular hence it is also recursively enumerable. L2 is deterministic context free language so, it is also recursively enumerable . Recursively enumerable languages are closed under intersection.

Thus, problem mentioned in option (A) is undecidable.

S24: C

Explanation:

Given, 'm' words, each of length 'n' bits, so total number of bits = m*n A state of finite state automata can store 1 bit of data, and a state can have either 0 or 1. Therefore, total number of states required by a Finite State Machine = 2mn. So, option (C) is correct.

S25: C

Explanation:

complement of regular Language is regular.

S26: D

Explanation:

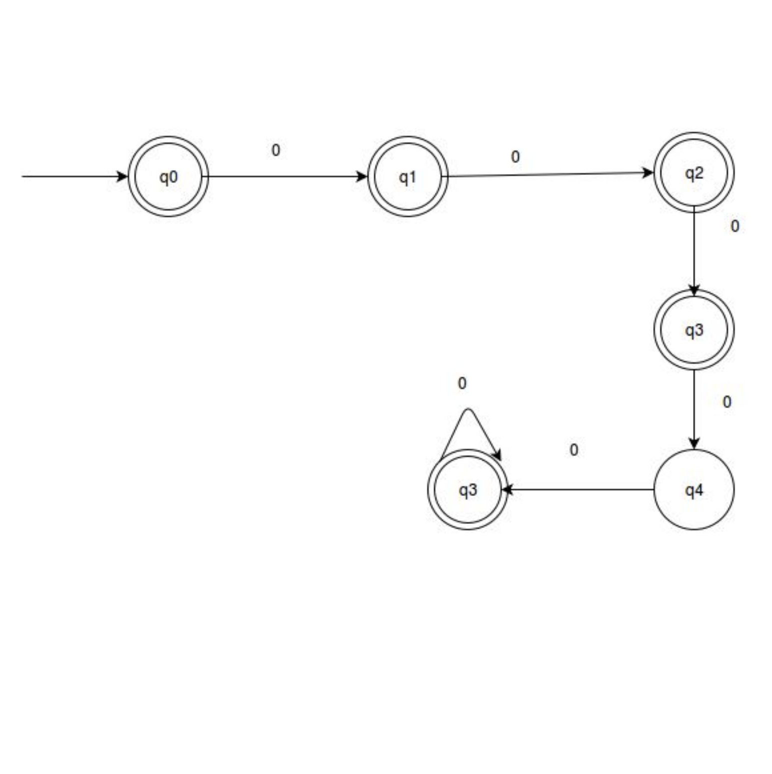
1. $\{a^n b^n \mid n > 0\}$ is a deterministic context free language but not regular language
2. The complement of $\{a^n b^n a^n \mid n > 0\}$ is context free language but not accepted by deterministic pushdown automata
3. $\{a^n b^n a^n\}$ is a context sensitive language but not recursive language
4. L is a recursive language but not a context free language

So, option (D) is correct.

S27: D

Explanation:

The minimal deterministic automata of given language is as follows



S28: B

Explanation:

Given language is ww and it is Non-CFL but CSL, so, we have a Turing machine which accepts it.

S29: C

Explanation:

L1 is decidable because This is the empty set, since every language has an infinite number of TMs that accept it.

L2 is decidable. In this question, we are talking about all the descriptions of Turing machines using a fixed alphabet (of finite size, of course), i.e., TM’s that are encoded as input to the universal TM. So, L2 is finite, and hence recursive

S30: D

Explanation:

The solution must start with index because choices and result in an immediate mismatch. But then, the is shorter than the , and there is no pair that allows the length of the to grow faster than that of the . Hence, we cannot create any solution to this problem.

S31: B

Explanation:

Equal No of a's ,b's and c's is not context-free, but it's complement is Context-free.

That is ,

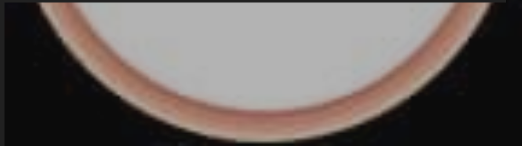
$$L = \{w \mid no_a(w) \neq no_b(w), w \in (a+b+c)^*\}$$

$$\cup$$

$$\{w \mid no_b(w) \neq no_c(w), w \in (a+b+c)^*\}$$

$$\cup$$

$$\{w \mid no_c(w) \neq no_a(w), w \in (a+b+c)^*\}$$



S32: A

Explanation:

we can create PDA for the given language.

first i find the relation between n & m.

given that m < 2n < 3m

i break this in two equation

1. 2n > m i.e [n > m/2] -----------eqn-1
2. 2n < 3m i.e [n < 1.5m] -------------eqn-2

from eqn 1 & 2 i found that [m >= n+1]for n,m >= 2 i.e if n=2 then m=3 n=3 then m=4 etc

now it is simple to construct PDA for the above language a^m b^n where m>=n+1 & m,n>=2

push all a into stack and when see 1st b leave it and when see 2nd b then start popping a from stack.

S33: D

Explanation:

Counter examples for given statements:

I. (anbn) ∪ (a*b*) = a*b*

where, a*b* is regular but (anbn) is not regular language.

II. Φ ∪ (ab) ∪ (a2b2) ∪ (a3b3) ..... (infinite union) = (anbn)

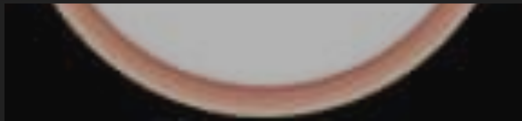
where, each language in left side are regular language, but language in right side (anbn) is not regular language. So, both statements are false. Option (D) is true.

S34: B

Explanation:

Expressing power of any machine can be defined as the maximum number of languages it can accept..if machine M_1 can accept more languages then M_2 then we can say that expressing power of M_1 is greater then M_2.

A. Languages accepted by NFA,will also be accepted by DFA because we can make DFA corresponding to NFA. So their expressing power is same.
B. In this case languages accepted by NPDA is more then DPDA, so expressing power of NPDA is more then DPDA
C. Both deterministic and non deterministic turing can accept same language.so there expressing power is same.
D. In turing machine if we increase the number of tape then also language accepted by that machine is same as single tape turing machine.so there expressing power is same.



S35: A

Explanation:

This is correct expression, this considering chance of not having any 0's (In that case string can also be empty string

S36: C

Explanation:

Initially tape has $b01010$

$\frac{input\ tape}{current\ state}$ and current R/W head position is in red, transition is used as given in TM .

$$\frac{b01010b}{q0} \rightarrow \frac{b01010b}{q0} \rightarrow \frac{b01010b}{q0} \rightarrow \frac{b01010b}{q0} \rightarrow \frac{b01010b}{q0} \rightarrow \frac{b01010b}{q0} \rightarrow \frac{b01010b}{q1} \rightarrow \frac{b01010b}{q1} -$$

10110 is the answer.

S37: A

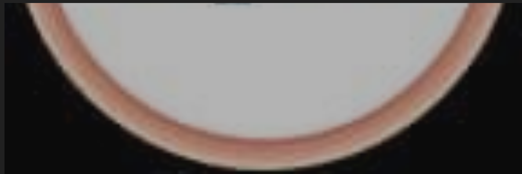
<u>**Explanation:**</u>

S_1 is TRUE.

If L_1 is recursive L_2 must also be recursive. Because to check if a word $w = w_i \# w_j$ belong to L_2, we can give w_i and w_j to the decider for L_1 and if both are accepted then w belong to L_1 and not otherwise.

S_2 is TRUE.

With a decider for L_2 we can make a decider for L_1 as follows. Let w_1 be the first string enumerated by algorithm A for L_1. Now, to check if a word w belongs to L_1, make a string $w' = w_1 \# w$ and give it to the decider for L_2 and if accepted, then w belongs to L_1 and not otherwise.



S38: B, C, D

Explanation:

Option A: Not context free, we have to match w with w^R but the top of the stack contains x, so not possible.

Option B: Context free, first match w with w^R and then x with x^R using stack.

Option C: Context free, it is just Σ^*

Option D: Content free, push w on stack then push x on stack match x with x^R and then w with w^R

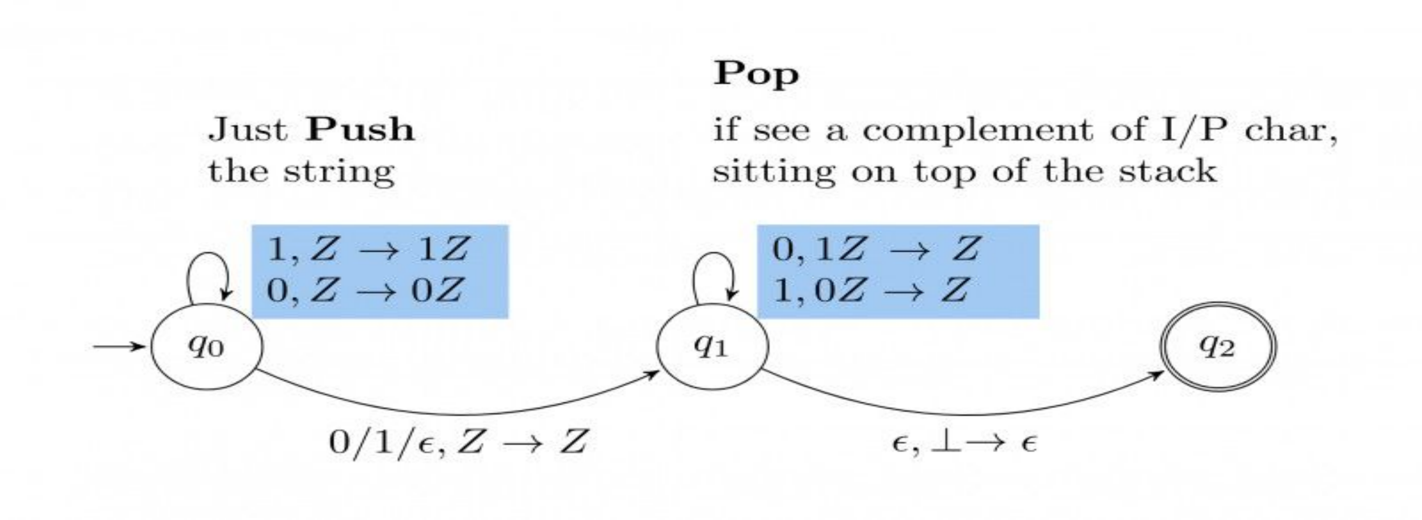
S39: B

Explanation:

Here, Z is used to represent the entire stack content except the top

Z is the string in Stack read from top to bottom. $1, Z \rightarrow 1Z$ means, on input symbol 1, the stack content changes from Z to $1Z$

In q_0 state, for '1', a '1' is pushed and for a '0' a '0' is pushed. In q_1 state, for a '0' a '1' is popped and for a '1' a '0' is popped. So, the given PDA is accepting all strings of of the form $x0x'_r$ or $x1x'_r$ or xx'_r, where x'_r is the reverse of the 1's complement of x. i.e.:



The given string 101100 has 6 letters and we are given 5 letter strings. So, $x0$ is done, with $x = 10110$. So, $x'_r = (01001)_r = 10010$.

S40: C

Explanation:

Alan turing proved in 1936 that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist.

S41: D

Explanation:

L_1 is context-free and hence recursive also. Recursive set being closed under complement, L_1' will be recursive.

L_1' being recursive it is also recursively enumerable and Recursively Enumerable set is closed under Union. So, $L_1' \cup L_2$ is recursively enumerable.

Context free languages are not closed under complement, so III is false

Recursive set is closed under complement. So, if L_2' is recursive, (L_2')' $= L_2$ is also recursive which is not the case here. So, II is also false.

S42: D

Explanation:

Option A :

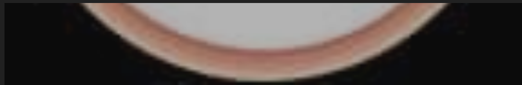
The Strings "a" and "b" are accepted by the given NFA BUT these strings are Not generated by the regular expression in Option A. So Option A is wrong.

Option B :

The String $abbbbaa$ is accepted by the given NFA BUT this string is Not generated by the regular expression in Option B. So Option B is wrong.

Option C :

The Empty String is NOT accepted by the given NFA BUT this string is generated by the regular expression in Option C. So Option C is wrong.



S43: C

Explanation:

Recursive enumerable languages are not closed under complement . while recursive languages are.

Both Recursive and Recursive enumerable languages are closed under intersection, union, and kleene star.

Non-Deterministic TM is equivalent to DTM

Only 2 is false. Option C is correct.

Note: Turing decidable language mean Recursive language and Turing recognizable language mean recursive enumerable language.

S44: A

Explanation:

$L = \{ a^2, a^5, a^8, a^{11}, a^{14}, \ldots\ldots\ldots, b^{10}, b^{22}, b^{34}, b^{46} \ldots\ldots\ldots \}$

since L=b10 has length ≥10 so, 10 must be pumped. But in b^{10} , If you take any non-empty substring and you remove that substring (Note that repeating substring y Zero times in w is equivalent to saying that remove y from the string w) then resulted string does not belong to the given language. So, Pumping length cannot be 10. And Since we know that if Minimum pumping length for a language is x then any number ≥x is also a Pumping length for the language. So, Since 10 is Not a pumping length for L, so, any number ≤10.

The minimum pumping length for the given language would be 12

S45: B

Explanation:

$I)\ \{a^m b^n c^p d^q \mid m + p = n + q,\ \text{where } m, n, p, q \geq 0\}$

Grammar :

$S \longrightarrow aSd \mid ABC \mid \in$
$A \longrightarrow aAb \mid ab \mid \in$
$B \longrightarrow bBc \mid bc \mid \in$
$C \longrightarrow cCd \mid cd \mid \in$

Above Grammar is CFG so it will generate CFL.

$II)\ \{a^m b^n c^p d^q \mid m = n \text{ and } p = q,\ \text{where } m, n, p, q \geq 0\}$

Grammar :

$S \longrightarrow AB \mid \in$
$A \longrightarrow aAb \mid ab$
$B \longrightarrow cBd \mid cd$

[Conti..]

Above Grammar is CFG so it will generate CFL.

$III)\ \{a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q,\ \text{where } m, n, p, q \geq 0\}$

More than one comparison so NOT CFL.

$IV)\ \{a^m b^n c^p d^q \mid mn = p + q,\ \text{where } m, n, p, q \geq 0\}$

It is CSL but NOT CFL.

S46: D

Explanation:

1^{st} statement is Membership problem of regular language $=$ **decidable**

2^{nd} statement is Emptyness problem of CFL $=$ **decidable**

3^{rd} statement is accept everthing problem of CFL $=$ **undecidable**

4^{th} statement is Membership problem of RE language $=$ **undecidable**

S47: C

Explanation:

Starting state : q_2 **and input string is "**aba**"**

- Step 1: **Find Epsilon closure of** $q_2 = \{q_2, q_0\}$
- Step 2: **Find transitions on** a **:**
  - $q_0 \rightarrow q_1$
  - $q_2 \rightarrow \emptyset$
- Step 3: **Find epsilon closure of** $q_1 = \{q_1, q_2, q_0\}$
- Step 4: **Find transitions on** b **:**
  - $q_1 \rightarrow q_3$
  - $q_0 \rightarrow q_0$
  - $q_2 \rightarrow \emptyset$
- Step 5: **Find epsilon closure of** $q_0 = \{q_0, q_2\}$ **UNION epsilon closure of** $q_3 = \{q_3\}, = \{q_0, q_2, q_3\}$
- Step 6: **Find transitions on** a **:**
  - $q_0 \rightarrow q_1$
  - $q_2 \rightarrow \emptyset$
  - $q_3 \rightarrow \emptyset$
- Step 7: **Find epsilon closure of** q_1 **:** $\{q_1, q_0, q_2\}$

S48: B

Explanation:

X is undecidable but partially decidable.

We have the TM M. Just make the state q the final state and make all other final states non-final and get a new TM M'. Give input w to M'. If w would have taken M to state q (yes case of the problem), our new TM M' would accept it. So, the given problem is partially decidable.

If M goes for an infinite loop and never reaches state q (no case for the problem), M' cannot output anything. This problem is the state entry problem, which like word accepting problem and halting problem is undecidable.

S49: A, D

Explanation:

A. A proper superset of context free languages. TRUE Since there are languages which are not CFL still Recursive
B. Always recognizable by pushdown automata. FALSE
C. Also called type 0 languages. FALSE Recursively Enumerable languages are actually type-0 languages.
D. Recognizable by Turing machines TRUE

S50: B, D

Explanation:

1. **Membership problem in context-free languages. is Decidable.**
2. **Whether a given context-free language is regular. Undecidable [Regularity is decidable till DCFL class]**
3. **Whether a finite state automation halts on all inputs. Decidable**
4. **Membership problem for type 0 languages. Undecidable [undecidable for RE or semi-decidable]**

S51: C

<u>**Explanation:**</u>

Because if that would have been possible then NPDA and DPDA must had same powers, which is not the case. You can take example of NFA and DFA. Both are convertible to each other and hence share the same power.

S52: A

Explanation:

The Proper PDA for the given language is given by option A

Input	Result
ab	Accept
abb	Accept
abbb	Reject
aab	Accept
aabb	Accept
aabbb	Accept
aabbbb	Accept
aabbbbb	Reject
aaab	Accept
aaa	Accept
bbbb	Accept
	Accept
bbbaaa	Reject

S53: D

Explanation:

The given PDA not accept any of the given language

S54: B

Explanation:

The given language first part b should greater than 3 and can be any number this can be done by DPDA similarly in second part of union a should greater than 3 of b and can be any number union of both also produce DCFL.

S55: A, C

Explanation:

A. $L(s) \subseteq L(r)$: strings generated by s are any numbers of 1's followed by one 0, i.e., $10, 110, 1110, 1110, \ldots$. Strings generated by r are is 1 followed by any combination of 0 or 1 , i.e., $1, 10, 11, 1110, 101, 110 \ldots$. This shows that all the strings that can be generated by s, can also be generated by r it means $L(s) \subseteq L(r)$ is true.
$L(s) \subseteq L(t)$: here strings generated by t are any numbers of 1 (here 1^* means we have strings as $\epsilon, 1, 11, 111, \ldots$) followed by only one 0, i.e., $0, 10, 110, 1110, \ldots$. So we can see that all the strings that are present in s can also be generated by t, hence $L(s) \subseteq L(t)$ which shows that option **A** is true.

B. $L(r) \subseteq L(s)$: this is false because string 1 which can be generated by r, cannot be generated by s.

C. Same as option **A.**

D. $L(t) \subseteq L(s)$: this is false because string 0 which can be generated by t,cannot be generated by s.

S56: B

<u>Explanation:</u>

There are finite number of strings of length ‘2014’. So, a turing machine will take the input string of length ‘2014’ and test it.

If, input string is present in the language then turing machine will halt in final state .

But, if turing machine is unable to accept the input string then it will halt in non-final state or go in an infinite loop and never halt.
Thus, ‘L’ is undecidable and recursively enumerable.

S57: A

Explanation:

Solution: Let us see whether machine halts on string ‘1’. Initially state will be q0, head will point to s:

Using $\delta(q0, 1) = (q1, 1, R)$, it will move to state q1 and head will move to right as:

Using $\delta(q1, B) = (q0, B, L)$, it will move to state q0 and head will move to left as:

It will run in the same way again and again and not halt.

Option C says M halts on all string ending with 0, but it is not halting for 0. So, option C is incorrect.

Option B says that TM does not halt for any string $(00 + 1)^*$. But NULL string is a part of $(00 + 1)^*$ and TM will halt for NULL string. For NULL string, tape will be,

Using $\delta(q0, B)$ = halt, TM will halt. As TM is halting for NULL, this option is also incorrect.
So, option (A) is correct.

S58: D

Explanation:

The fibonacci sequence generation not possible with automata

S59: C

Explanation:

All the languages that are accepted by finite automata are known as regular language. Grammar that generates regular languages is regular grammar. So, a push down automata accepts both regular and context free languages

S60: A

Explanation:

For $P1$, we just need to give a run on the machine. Finite state machines always halts unlike TM.

For $P2$, check if the CFG generates any string of length between n and $2n-1$, where n is the pumping lemma constant. If So, $L(CFG)$ is infinite, else finite. Finding the pumping lemma constant is not trivial - but there are other procedures which can do this -

Hence both are decidable

S61: B

Explanation:

First 0’s are pushed into stack. Then 1’s are pushed into stack.
Then for every 2 as input a 1 is popped out of stack. If some 2’s are still left and top of stack is a 0 then string is not accepted by the PDA. Thereafter if 2’s are finished and top of stack is a 0 then for every 3 as input equal number of 0’s are popped out of stack. If string is finished and stack is empty then string is accepted by the PDA otherwise not accepted

Hence option B is the answer

S62: C

Explanation:

S=T Both generates all strings over Σ.

S63: A

Explanation:

The number of output here follows the transitions in place of states as in Moore machine.

S64: B

Explanation:

According to the question, minimum of 3 states are required to recognize an octal number divisible by 3.

S65: D

<u>Explanation:</u>

There can be four cases while processing the given input string. Case-1: m=0 – In this cases the input string will be of the form {c(4*n)dn}. In this condition, keep on pushing c's in the stack until we encounter with 'd'. On receiving 'd' check if top of stack is 'c', then pop 'cccc' from the stack. Keep on popping cccc's until all the d's of the string are processed. If we reach to the end of input string and stack becomes empty, then reached to the final state, i.e., accepts the input string else move to dead state. Case-2: n=0 – In this cases the input string will be of the form {a(2*m)bm}. In this condition, keep on pushing a's in the stack until we encounter with 'b'. On receiving b check if top of stack is 'a', then pop 'aa' from the stack. Keep on popping aa's until all the b's of the string are processed. If we reach to the end of input string and stack becomes empty, then reached to the final state i.e., accepts the input string else move to dead state. Case-3: m, n>0 – In this cases the input string will be of the form {(a(2*m)c(4*n)dnbm}. In this condition, keep on pushing a's and c's in the stack until we encounter with 'd'. On receiving d check if top of stack is 'c', then pop 'cccc' from the stack. Keep on popping cccc's until all the d's of the input string are processed. Then On receiving b check if top of stack is 'a', then pop 'aa' from the stack. Keep on popping aa's until all the b's of the input string are processed. If we reach to the end of input string and stack becomes empty, then reach to final state i.e., accept the input string else move to dead state. Case-4: m, n=0 – In this case the input string will be empty. Therefore directly jump to final state.