Q.1) Max Marks: 1

Consider the following statements is/are True

I. If L_1 and L_2 are context free languages, then the language L_1-L_2 is context free

II. If L_1 is context free and L_2 is regular, then the language L_1-L_2 is context free

A **I Only**

B **II Only** Correct Option

Solution: (B)

Answer: B

Explanation:

I. L_1-L_2 , Consider the case $L_1 = \Sigma^*$ then the difference is like Complementation and CFLs are not closed under complementation.

II. L_1-$L_2 \Rightarrow L_1 \cap L_2^c$,
Since L_2^c is regular and CFLs are closed under intersection operation with regular languages. CFL

C **Both I and II**

D **Neither I nor II**

Q.2) Max Marks: 1

The equivalent chomsky normal form for the given CFG is

$S \rightarrow AbA$

$A \rightarrow Aa \mid \varepsilon$

A
$S \rightarrow T A \mid BA \mid b$
$A \rightarrow AC \mid a$
$T \rightarrow AB$
$B \rightarrow b$
$C \rightarrow a$

B
$S \rightarrow T A$
$A \rightarrow AC \mid \varepsilon$
$T \rightarrow AB$
$B \rightarrow b$
$C \rightarrow a$

C Correct Option
$S \rightarrow T A \mid BA \mid AB \mid b$
$A \rightarrow AC \mid a$
$T \rightarrow AB$
$B \rightarrow b$
$C \rightarrow a$

Solution: (C)

Answer: C

Explanation:

Eliminate the ε production from the grammar

$S \rightarrow AbA \mid bA \mid Ab \mid b$

$A \rightarrow Aa \mid a$

Convert it to the CNF

$S \rightarrow T A \mid bA \mid Ab \mid b$

$A \rightarrow Aa \mid a$

$T \rightarrow Ab$

And finally,

$S \rightarrow TA \mid BA \mid AB \mid b$

$A \rightarrow AC \mid a$

$T \rightarrow AB$

$B \rightarrow b$

$C \rightarrow a$

D
$S \rightarrow T A \mid BA \mid AB \mid b$
$A \rightarrow AC \mid \varepsilon$
$T \rightarrow AB$

$B \rightarrow b$
$C \rightarrow a$

Q.3)

Max Marks: 1

Match the following

a. Regular	X. intersection of Recursive and Recursively enumerable language
b. Recursive	Y. $a^i b^j c^k d^l$ | i=l and j=k
c. Context free language	Z. Complement of a CFL
d. Recursive enumerable language	W. $a^i b^j c^k$ | i*j*k mod 5

A a-w, b-z, c-y, d-x — Correct Option

Solution: (A)

Answer:A

Explanation:

$a^i b^j c^k$ | i*j*k mod 5 is Regular

$a^i b^j c^k d^l$ | i=l and j=k is CFL

Complement of a CFL is a Recursive language

intersection of Recursive and Recursively enumerable language is REL

B a-z b-w, c-y, d-x

C a-w, b-z, c-x, d-y

D a-z, b-x, c-y, d-w

Q.4)

Max Marks: 1

Consider the following code segment.

x = u - t;
y = x * v;
z = y + w;
w = t - z;
u = x * y;

The minimum number of total variables required to convert the above code segment to static single assignment form is__________

Correct Answer

Solution: (7)

Answer:7

Explanation: Static Single Assignment is used for intermediate code in compiler design. In Static Single Assignment form(SSA) each assignment to a variable should be specified with distinct names. We use subscripts to distinguish each definition of variables.

In the given code segment each definition is distinct

So, the total number of variables is (x,u,t,y,v,z,w).

Q.5)

Max Marks: 1

Which of the following is/are False

A DCFLs are closed under Complement operation

B CSLs are closed under Kleene star operation

C RELs are closed under intersection operation

D Recursive languages are closed under Homomorphism. — Correct Option

Solution: (D)

Answer: D

Explanation:

DCFLs are closed under Complement operation : True

CSLs are closed under Kleene star operation: True

RELs are closed under intersection operation: True

Recursive languages are closed under Homomorphism. : False

Q.6)

Max Marks: 1

Consider the following DFA

‖ 0 | 1

→ *q0	q3	q1
q1	q4	q2
q2	q5	q0
q3	q0	q4
q4	q1	q5
q5	q2	q3

The language accepted by the given DFA is

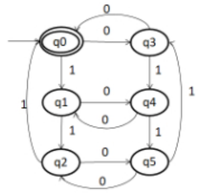
A **The set of strings with an even number of 0s and a "triple" number of 1s.** Correct Option

Solution: (A)

Answer: A

Explanation:

Given DFA is



The DFA accepts all the strings of even length and number of 1's are multiples of 3.

B **The set of strings with an even number of 0s and odd number of 1s**

C **The set of strings with an even number of 0s followed by triple number of 1s.**

D **The set of strings with triple number of 1s followed by an even number of 0s.**

Q.7) Max Marks: 1

The context free grammar that generates the following language
L = { w ∊ {0, 1}* | the length of w is odd and the middle symbol is 0 }

A **S → 0S0 | 1S1 | 0**

B **S → 0S1 | 1S0 | 0**

C **S → 0S0 | 0S1 | 1S0 | 1S1 | 0** Correct Option

Solution: (C)
Answer: C
Explanation:
Possible strings in the language are {0,000,101,11011,10010,.......}
The grammar that generates the given language is
S → 0S0 | 0S1 | 1S0 | 1S1 | 0

D **S → 0S0 | 0S1 | 1S0 | 1S1 | 0 | 1**

Q.8) Max Marks: 1

The left factored and non-left recursive grammar for the given grammar is

```
Numeral ::=  Digits | Digits . Digits
           | Digits e Sign Digits
           | Digits . Digits e Sign Digits
Digits  ::=  Digit | Digits Digit
Digit   ::=  0 | 1 | 2 | 3
```

A
Numeral ::= Digits N1 N2
N1 ::= e Sign Digits | ε
N2 ::= Digits | ε
Digits ::= Digit Digits | Digit
Digit ::= 0 | 1 | 2 | 3

B
Numeral ::= Digits N1 N2
N1 ::= e Sign Digits | ε
N2 ::= . Digits | ε
Digits ::= Digit Digits | Digit
Digit ::= 0 | 1 | 2 | 3

C
Numeral ::= Digits N
N ::= e Sign Digits | . Digits | ε
Digits ::= Digit Digits | Digit
Digit ::= 0 | 1 | 2 | 3

D

Numeral ::= Digits N1
N1 ::= e Sign Digits | ε | .Digits N2
N2 ::= e Sign Digits | ε
Digits ::= Digit D
D ::= Digits | ε
Digit ::= 0 | 1 | 2 | 3

Correct Option

Solution: (D)

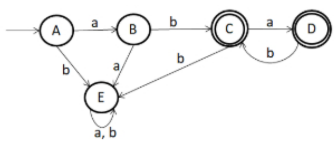
Answer: D
Explanation:
The given grammar is
Numeral ::= Digits | Digits. Digits | Digits e Sign Digits | Digits . Digits e Sign Digits
Digits ::= Digit Digits | Digit
Digit ::= 0 | 1 | 2 | 3

Apply Left Factoring
Numeral ::= Digits N1 N2
N1 ::= e Sign Digits | ε
N2 ::= . Digits | ε
Digits ::= Digit D
D ::= Digits | ε
Digit ::= 0 | 1 | 2 | 3

Q.9)

Max Marks: 1

Consider the following DFA



The language accepted by the given DFA is

A **Set of all strings starts and ends with ab**

B **Set of all strings starts with ab and ends with ba**

C **Set of all strings starts with ab or ends with ba**

D **None of these**

Correct Option

Solution: (D)

Answer: D
Explanation:

A. Set of all strings starts and ends with ab False
Let's consider the string **abbab** starts and ends with ab but not accepted by the DFA
B. Set of all strings starts with ab and end ends with ba
Let's consider the string **abba** starts and ends with ab but not accepted by the DFA
C. Set of all strings starts with ab or ends with ba
Let's consider the string **abb** starts with ab but not accepted by the DFA

Q.10)

Max Marks: 1

The regular expressions for the following language L over the alphabet Σ = {a, b}:
L = All strings that do not end with aa.

A **(a+b)*(ab+ba+bb)**

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

C ε+a+b+(a+b)*(ab+ba+bb)

Correct Option

Solution: (C)

Answer:C
Explanation:
ε+a+b+(a+b)*(ab+ba+bb)
={ε, a, b, ab, ba, bb, aab,bab,.............}
Generates all the strings which does not end with aa

D None of these

Q.11) Max Marks: 2

Which of the following is/are True

I. If L is a recursive language and F is finite language then L-F is recursive

II. If L is a non-recursive and F is a finite language then L ∪ F is Non-recursive

A I only

B II only

C **Both I and II** Correct Option

Solution: (C)

Answer:C

Explanation:

I. L-F is L intersection F^c, Since all finite languages are regular and all regular languages are recursive. F is also recursive. Recursive languages are closed under complement and intersection operations. L-F is Recursive

II. Let $F_1 = F \cap L^c$ set of strings in F that do not belong to L. Clearly F_1 is finite as it is a subset of a finite set.
Assume $L_1 = L \cup F$ is recursive, then $L = L_1 - F_1$ is also recursive, which is a contradiction of L is non-recursive.
L U F is also non-recursive.

D Neither I nor II

Q.12) Max Marks: 2

Which of the following is True

A **Every subset of a regular language is regular.**

B Let L' = L1 ∩ L2. If L' is regular and L2 is regular, L1 must be regular.

C If L is regular, then so is L' = {xy : x ∈ L and y ∉ L} is also regular Correct Option

Solution: (C)

Answer: C

Explanation:

I. False
Every subset of a regular language is regular.
Let's consider an example $a^n b^n$ is subset of a*b* which is not regular

II. False
Let L' = L1 ∩ L2. If L' is regular and L2 is regular, L1 must be regular.
Let L' = ∅. Let L2 = ∅. So L' and L2 are regular. Now let L1 = {ai : i is prime}. L1 is not regular. Yet L' = L1 ∩ L2. Notice that we could have made L2 anything at all and its intersection with ∅ would have been ∅. When you are looking for counterexamples, it usually works to look for very simple ones such as ∅ or Σ*, so it's a good idea to start there first. ∅ works well in this case because we're doing intersection. Σ* is often useful when we're doing union.

III. If L is regular, then so is L' = {xy : x ∈ L and y ∉ L}. True
Proof: Saying that y ∉ L is equivalent to saying that y ∈ L. Since the regular languages are closed under complement, we know that L is also regular. L' is thus the concatenation of two regular languages. The regular languages are closed under concatenation. Thus L' must be regular.

IV. $\{w : w=w^R\}$ is regular. Non-regular False

D $\{w : w=w^R\}$ is regular.

Q.13) Max Marks: 2

Consider the following Context free grammar

S → XY | W

X → aXb | ε

Y → cY | ε

W → aW c | Z

Z → bZ | ε

The language generated by the given context free grammar is

A $L = \{ a^i b^j c^k \mid i, j, k \geq 0, \text{ and } i = j \text{ or } j = k \}$

B $L = \{ a^i b^j c^k \mid i, j, k \geq 0, \text{ and } i = j \text{ or } i=k\}$

Correct Option

Solution: (B)

Answer: B

Explanation:

Given grammar

S → XY

→ aXbY

→ aaXbbY

→ aaεbbY

→ aabbc (i=j)

S→ W

→ aWc

→ aaWcc

→ aaεcc

→ aacc

The grammar generates set of all strings such number of a's are equal to number of b's or number of a's are equal to number of c's.

C $L = \{ a^i b^j c^k \mid i, j, k \geq 0, \text{ and } i = k \}$

D **None of these**

Q.14)

Max Marks: 2

Which of the following languages are decidable

I. L1 = {<M> | M is a Turing machine that rejects all inputs of even length}.
II. L2 = {<M> | M is a Turing machine that halts on an empty input}.
III. L3 = {<M> | there is some input x ∈ {0, 1}* such that M accepts x in less than 100 steps}

A **All are undecidable**

B **II and III only**

C **I and II only**

D **III Only**

Correct Option

Solution: (D)

Answer: D

Explanation:

I. L1 is undecidable. To see this, assume on the contrary that there exists some TM R1 that decides L1, and we use R1 to construct a TM S1 that decides A_{TM}:
S1 = "On input <M, w>:
1. Construct TM M1 that on input x, accept if |x| is odd. If |x| is even, it simulates M on input w. If M accepts w, M1 enters the reject state. If M rejects w, M1 enters the accept state. If M loops, M1 also loops.
2. Run R1 on input <M1>.
3. Accept if R1 accepts, and reject if R1 rejects." Observe that if M accepts w, then M1 is a Turing machine that rejects all inputs of even length. If M rejects or loops on input w, then M1 is a Turing machine that for each input of even length, either loops or accepts.

II. L2 is undecidable. To see this, assume on the contrary that there exists some TM R2 that decides L2, and we use R2 to construct a TM S2 that decides ATM: S2 = "On input <M, w>:
1. Construct TM M2 that ignores its input and simulates M on input w and accept (and halt) if M does. If M rejects w, M2 keeps moving right upon reading any input (thereby looping).
2. Run R2 on input <M2>.
3. Accept if R2 accepts, and reject if R2 rejects." Observe that if M accepts w, then M2 is a Turing machine that halts on an empty input. If M rejects or loops on input w, then M2 is a Turing machine that loops on an empty input.

III. L3 is decidable. First, observe that if <M> ∈ L3, then there exists some string x of length at most 100 such that M accepts x in less than 100 steps. This is because M cannot read beyond the 100th position of its input in less than 100 steps. Therefore, to check whether an input <M> is in L3, it suffices to simulate M over all strings of length at most 100 for at most 99 steps, and accept if M accepts one of these strings, and reject otherwise.

Q.15)

Max Marks: 2

Consider the following grammar
SL→ SL;S
SL→ ε
S→ stmt
The given grammar is

A LR(0) and SLR(1) Correct Option

Solution: (A)

Answer: A
Explanation:
LR(0) Items: No conflicts in LR(0), SLR(1)

```
s0:
   [S' -> .SL]
   [SL -> .SL; S]
   [SL -> .]               goto(s0, SL) = s1;
s1:
   [S' -> SL.]
   [SL -> SL.;S]           goto(s1, ;) = s2;
s2:
   [SL -> SL;.S]
   [S -> .stmt]            goto(s2, S) = s3;
s3:
   [SL-> SL; S.]           goto(s2, stmt) = s4
s4:
   [S -> stmt.]
```

The given grammar is LR(0) and SLR(1)
SLR(1) parse table is

	;	stmt	$	SL	S
s0	r2		r2	1	
s1	s2		Accept		
s2		s4			3
s3	r1		r1		
s4	r3		r3		

B Not LR(0) but SLR(1)

C Neither LR(0) nor SLR(1)

D LL(1) but not LR(0)

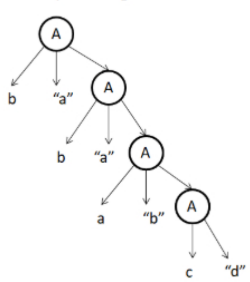
Q.16) Max Marks: 2

. Consider the following SDT
A→ b {print(“a”)}A
A→ a {print(“b”)}A
A→ c{print(“d”)}
The output produced by the SDT for the input string bbac

A aabd Correct Option

Solution: (A)

Answer:A
Explanation:
Given input string is bbac and the output produced by the SDT is aabd



B abda

C dbaa

D None of these

Q.17)

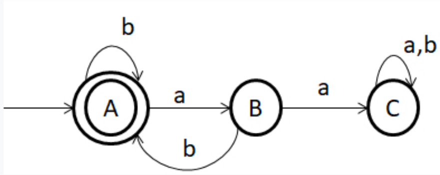
Max Marks: 2

Minimum number of states in a Deterministic finite automaton that accepts the given language is______

L = { w ∈ {a, b}* : every a is followed by at least one b }

Correct Answer

Solution: (3)



Q.18)

Max Marks: 2

Consider the following context free grammar

S→ ABBA

A→ a| ε

B→ b| ε

The entries in the following LL(1) parse table M is

	a	b	$
S	S→ ABBA	S→ ABBA	S→ ABBA
A			
B		B→ b	

The entries for the M[A,a], M[A,b], M[A,$] is

A

M[A,a] = A→ a,
M[A,b] = A→ ε,
M[A,$] = A→ ε

B

M[A,a] = {A→ a, A→ ε}
M[A,b] = A→ ε,
M[A,$] = A→ ε

Correct Option

Solution: (B)

Answer:B

Explanation:

First (S) = {a,b, ε}
First(A) = {a, ε}
First(B) = {b, ε}
Follow(S) = {$}
Follow(A) = First(B) = {b} u First(B) = {b} u First(A)= {b,a} U follow(S) = {a,b,$}
Follow(B) = Frist(B) ={b,a,$}
LL(1) parse table is

	a	b	$
S	S→ ABBA	S→ ABBA	S→ ABBA
A	A→ a,A→ ε	A→ ε	A→ ε
B	B→ ε	B→ b, B→ ε	B→ ε

C

M[A,a] = A→ a,
M[A,b] = A→ ε,
M[A,$] = Nil

D

M[A,a] = {A→ a, A→ ε}
M[A,b] = A→ ε,
M[A,$] = Nil

Q.19)

Max Marks: 2

Which of the following languages are regular

A. {w ∈ Σ* | |w| is a power of 2}, Σ = {a}.
B. {w | w is a string of balanced parenthesis}, Σ = {(,)}.
C. $\{0^n10^m \mid n,m \geq 0\}$, Σ = {0,1}

A

I, II,III

B

I and II only

C I and III only

D **III Only** — Correct Option

Solution: (D)

Answer: D

Explanation:

$\{w \in \Sigma^* \mid |w| \text{ is a power of } 2\}$, $\Sigma = \{a\}$. Not-regular

Possible strings in the language = {a, aa, aaaa,aaaaaaaa,.............}

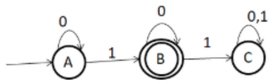
Series is not in the arithmetic progression.

{w | w is a string of balanced parenthesis}, $\Sigma = \{(,)\}$. Non-regular.

Non-regular, we need memory element to check the equality between balanced paranthesis.

$\{0^n10^m \mid n,m \geq 0\}$, $\Sigma = \{0,1\}$ regular.

FA for the language is



Q.20) — Max Marks: 2

Which of the following languages are CFLs

I. $L_1=\{a^ib^jc^kd^l \mid i+k=j+l, \text{ and } i,j,k,l \geq 0\}$

II. $L_2=\{a^ib^jc^kd^l \mid i=j \text{ and } k=l, \text{ and } i,j,k,l \geq 0\}$

A **I Only**

B **II Only**

C **I and II Only** — Correct Option

Solution: (C)

Answer: C

Explanation:

L_1 is CFL

Consider for the case i>=j

Push A for each a

Pop A for each b

Push C for each c

Pop C first and then A for each d.

Accept if stack is empty when input finishes.

For the case

i<j

Push A for each a

Pop A for each b until stack becomes empty

Push B for each b after that

Pop B for each c until stack becomes empty

Push C for each c after that

Pop C first and then B for each d.

Accept if stack is empty when input finishes.

So, clearly L is CFL.

L_2 is also CFL

Push All a's on to the stack. POP one 'a' from the stack for every b

Push All c's on to the stack. POP one 'c' from the stack for every d.

D **Neither I nor II**

close