JNTU ONLINE EXAMINATIONS [ACD-MID-I]

1 . Let language L1 denotes 3 strings &L2 denotes 4 strings. L1 L2=ϕ(no common strings) Then
no of strings for language L1 L2
(a) 3
(b) 7
(c) 4
(d) 12

2 . Regular expression for a language which contains number of a' s are more than number of b's
(a) a+b*
(b) (a+b)* a
(c) a(a+b*)
(d) (aba+a+aab+baa)+

3 . In the figure 3 represent NFA.The total number of states when converted into DFA Figure3
(a) 4
(b) 2
(c) 5
(d) 3

4 . In figure 4, find its , equvalent DFA
(a) Figure 4 a Figure 4 a
(b) Figure 4 b Figure 4 b
(c) Figure 4 c Figure 4c
(d) Figure 4 d Figure 4 d

5 . Let N1 is the finite state M/c for Regular expression R1 and N2 is the finite state machine for Regular expression R2 . With respect to Union operation .
Statement1: Introduce new initial state and provide NULL transitions for initial states of Machine
N1,Machine N2.
Statement2:The final states of Machine N2 becomes final states for N1 U N2 .
(a) Only statement 2 is TRUE
(b) Only statement 1 is TRUE
(c) Both statements are TRUE
(d) Neither statement is correct

6 . Which of the following statement is TRUE with respect to Input buffering
(a) The single buffer input scheme gives move performance
(b) The single buffer input scheme with sentinels gives more performance
(c) The double buffer input scheme with sentinels gives more performance
(d) The double buffer input scheme gives more performance

7 . Consider the grammer Be xpr b exprorbterm | b term b term b term and b factor | bfactor b factor not b factor | (b expr) | true | fabe The Non-terminals of the grammer
(a) { o r , a n d , n ot }
(b) { b e x p r , o r , b t e r m , (,) }
(c) { b e x p r , b t e r m , b f a c t or }
(d) { t r u e , f a b e , o r , a n d , n o t }

8 . S→0SI | IS0 | Λ denote a language L. Identify most correct answer
(a) equal number of o's and equal number of 1's with all combinations
(b) on 1n,n≥0
(c) 0 n , 1 m 0 n , m , n ≥ 0
(

9 . FIRST&FOLLOW functions are used to construct parsing Table. The production A→ α is to be

placed in all First(α) entries. If α is NULL then **A→∈ is to be placed in entries of passing Table** .
(a) FIRST (A) ∩ FOLLOW (A)
(b) FOLLOW(A)
(c) FIRST(A)

(d) FIRST (A), FOLLOW (A)
10. A→Bα if α is NULL the FOLLOW **(B) is**
(a) FOLLOW (A)
(b) FIRST (B)
(c) FIRST (A)
(d) FOLLOW (B)
11. Identify the false statement
(a) LL(K) grammer must be un-ambiguous
(b) LL(K) grammer must be free from left recussion
(c) All LL(K) grammers are context free Grammers
(d) Some LL(K) grammers does not be longs to CFG
12. S→ BabcDe D →e/f Trailing symbols of S,D
(a) {e},{e}
(b) {c,e},{e,f}
(c) {e},{c,f}
(d) {e},{e,f}
13. Predictive Parser must free from
(a) No restriction on the production
(b) common prefixes & left factoring
(c) Left recursion & left factoring
(d) common prefixes and left recursion
14. Consider the following LL(1) passing Table x y $ S S→ xEy E E→ ∈ E→ YE FOLLOW(E), FIRST(S) are
(a) {x}, {y}
(b) {in},{x}
(c) {x}, {in}
(d) {x},{x}
15. Which of the following passer is most power full
(a) SLR
(b) Operator precedence
(c) CALR
(d) LALR
16. A→Bac
B→cd
C→e
Which of the following items does not belongs LR(o) I o group
(a) B → .Cd
(b) C→ e
(c) C → .e
(d) A → .Bac
17. Which of the following production does not belongs to operator precedence grammer
(a) E → EAE
(b) E→ E*E
(c) E→ E+E
(d) E→ (E)
18. Synthesized attribute can be easily simulated by .

(a) LR grammer
(b) Ambiguous grammer
(c) LL grammers
(d) Regular grammer
19. Let the folowing Anotated passe Tree 3+5 Shown in Figure 19 F i g u re 19
(a) 3
(b) 8
(c) 15
(d) 5
20. The translation rule part of YACC specification is enclosed between
(a) % — — - %

(b) % % — — - %
(c) % % — — - % %
(d) % — — - % %
21.Let the language
L1={hope, fear}
L2 ={less, more}
Then L1 L2 denotes
(a) { hope less, hope more, fear less, fear more}
(b) {less hope , less fear, more hope, more fear}
(c) {hope more, fear less }
(d) {hope less, fear more}
22 . Which of the following language is not regular over an alphabet = {a,b}
(a) Number of a's are two
(b) string contain sub string aba
(c) Equal number of a's and equal number of b's
(d) strings start with a end with b
23 . Which of the following regular expression represent identifiers if D represent regular expression for digits and L denote regular expression for alpha numeric characters
(a) D(L+D)*
(b) L*D*
(c) DL*
(d) L(L+D)*
24 . In the figure 4 m/c represent NFA. The total number of states in its equivalent DFA denote NFA, what will the number of states when converted to DFA F i g u re 4
(a) 4
(b) 3
(c) 5
(d) 2
25 . Let N1 denotes finite state Machine for Regular expression R1 while N2 denote finite state machine for regular expression R2 then to get machine for R1 R2 (concatenation of R1 with R2)
Statement1 : The final states of N1 are converted into N on final states and the Null transition is
introduced between final state of N1 and initial state of N2
Statement2: The final states of N2 becomes final state for R1 R2
(a) Both statements are TRUE
(b) o n l y s t at e m e nt 1 i s T RU E
(c) N e i t h e r t h e s t at e m e nt s ar e T RU E
(d) o n l y s t at e m e nt 2 i s T RU E
26 . Consider a regular expression *.0 used in Unix It extracts
(a) Al l fi l e n a m e s s ta r ti n g w i th *
(b) All file names which has extension 0

(c) a file whose name is *.0
(d) All files with extension 0

27. With respect to the following grammer E → E+E | E* E | id identify correct statement

(a) It generates strings whose length is atmost three
(b) It generates strings whose length is atleast three
(c) It generates strings of length three
(d) It is ambiguous grammer

28. Which of the following grammer production is left recursive

(a) A → aA
(b) A → Ab
(c) A → aB
(d) A → Ba

29. Non-recersive descent passing is free from

(a) stack over flow
(b) Back Tracking
(c) errors
(d) passing Table memory

30. The rule used in Non-recursive Predictive Passing if input symbol and Top symbol of stack are matched

(a) Pop the top symbol of stack without advancing input pointer
(b) Advance input symbol pointer
(c) Pop the top symbol of stack
(d) Pop the top symbol of stack and advance input symbol pointer

31. S → iEtSS1 /a
S1 → eS/ ∈
E → b

(a) left recursive grammer
(b) LL(1)
(c) Ambigous
(d) Un-ambiguous grammer

32. A → aBcdefG
B → b
Leading symbols of A,B are

(a) {a, c}, {b}
(b) {a, f}, {b}
(c) {f}, {b}
(d) {a}, {b}

33. Which of the following Top down parsers are Table driven

(a) Recursive descent, Recursive Predictive parser
(b) Recursive descent, Brute-Fource parser
(c) Non-recursive Predictive parser
(d) Recursive descent, Non-recursive predictive parser

34. If the production A → α, α # NULL and a is a terminal. The entry of M[A, **a] is A →** α iff

(a) a ∈ FIRST(A)
(b) a∈ FIRST (α)
(c) a ∈ FOLLOW (α)
(d) a ∈ FOLLOW(A)

35. LALR passing Table is constructed from

(a) LL(0)
(b) LL(1)

(c) L R(0)
(d) LR(1)
36. The dis-advantage of Panic-Mode of error recovery
(a) It takes more implementation complexity
(b) I t n e ve r ge t s i n t o i n f i n i t e l o o p s
(c) S u i t ab l e f o r a l l s i t u at i o n s
(d) I t i s m o re e ffi c i e nt
37. The function of following TRANSLATIONS CHEME for input a+b+c
E → TR
R → + T {Print ("+")} R, R →∈ T → id {print (id.name)}
(a) a+b c +
(b) + + ab c
(c) ab+c+
(d) a b c + +
38. The structure of YACC program contains
(a) D e c l a r at i o n Par t , s u p p o r t i n g C - Ro u t i n g Pa r t
(b) D e c l a r at i o n Par t , s u p p o r t i n g C - Ro u t i n g Pa r t
(c) Declaration Part, Translation rules Part, supporting C- Routing Part
(d) Tra n s l a t i on Par t , s u p p o r t i n g C - R ou t i n g Pa rt
39. The number of states in DFA for strings over an alphabet = {a, b} such that the last but one symbol of the string is a
(a) 2

(b) 5
(c) 4
(d) 3
40. A grammer is said be in minimal form if it is free from use less productions and use less symbols. The minimized grammer for the frammer S → 0 | A
S → AB
B → 1
(a) S → 0 /A , A → A B
(b) S → 0 , B → 1
(c) S → 0
(d) S → 0 | A
41. The resulting grammer when common pre tixes are removed using left factoring from the following grammer S → ab | ad | cs
(a) S → c s | a B | a d B → d
(b) S → c s | B B → a b | a d
(c) S → aB1 | cs ,B1→ b | d
(d) S → B | c s | D B → a b D → a d
42. The rule used in non-recursive predictive parsers if input symbol is terminal and top of the stack is Non-terminal
(a) Push the associated production on to stack without advancing the input pointer
(b) N o - o p e r at i o n o n e i t h e r s ta ck a n d n o a d va n c e m e n t o f i n p u t p oi nt e r
(c) P u s h t h e a s s o c i at e d p r o d u c t i on on to s t a ck a n d a d va nc e i n p u t p oi nt e r
(d) Po p t h e p r o d u c t i on f r om s t ack a n d a d va n c e i n p u t p o i nt e r
43. A → aBcdefG
B → b Leading symbols of A,B are
(a) { a }, { b }
(b) { a , c } , { b }
(c) { a , f } , { b }
(d) { f } , { b }

44. If any entry of LL(1) passing Table contains more than one Production then
(a) G r a m m e r i s LL (1)
(b) Grammer is ambiguous
(c) G r a m m e r i s LL (2)
(d) G r a m m e r i s u n amb i g uo u s
45. In SLR passing Table if A → α . (Reduce) belongs to I, then A → α is written in all places of
M[I,a] where a
(a) a ∈ FIRST (α)
(b) a ∈ F I R S T (A)
(c) a ∈ FOLLOW (A)
(d) a ∈ FOL L OW (α)
46. LR(K) denote
(a) R i ght t o l e f t s c an n i n g t h e i n p u t, Re ve r e Ri g ht m os t d e ri vat i o n , K - s y mb ol s l o ok a
h
e
ad
(b) R i ght t o l e f t s c an n i n g t h e i n p u t, Re ve r s e L e f t m o s t d e ri vat i o n , K - s y mb ol s
L o o ka h e a d
(c) Left to right scanning of the input, Right most derivation in reverse, K symbols Look ahead
(d) L e f t t o r i g ht s c an n i n g o f th e i n p u t , L e f t m o s t d e ri vat i o n i n r e ve r s e , K - s ymb
o l s Lo o ka h e a d
47. In a syntax Directed Translation scheme if the value of an attribute of a node is a function of
the values of attributes of its parent and children then it is called as
(a) S y nth e s i z e d at t r i b u t e

b) P r o s p e c t i ve a t t r i b u t e
(c) Inheritted attribute
(d) C a n n on i c al a t t r i b u t e
48. The TRANSLATION scheme given below
E → TR
R → + T { Print (”+”)} R1 R →∈ T → id {print (id .name)} functions
(a) infix to post fix
(b) p o s t fi x t o p o s t fi x
(c) i n fi x t o i n fi x
(d) p o s t fi x t o p r e fi x
49. A finite automation M denoted by. 5 terple {Q, , δ , qo, qF} the δ maps (Let p ≤**Q)**
(a) Q → X 2 Q
(b) Q X → 2 Q
(c) Q → X p
(d) Q X → p
50. Which of the following string formed by alphabet = { a,b} does not belongs to NFA show n in figure 4 F i g u re 4
(a) ab b
(b) b b
(c) a aa b b
(d) abab

51. The transition function of NFA δ : Q × → **?**
(a) Q Q
(b) ZQ
(c) Q
(d) Q 2
52. Which of the following is not the compiler analysis phase
(a) S e m a nti c A n al y s i s
(b) L e x i c a l A n a l ys i s
(c) S y nta x A n a l ys i s
(d) Object code Analysis
53. Which of the following string is not derivable from the Grammer
S → (L) | a
L → L, S | S
(a) (a, (a , a)), (a , (a , a)))
(b) (a, a)
(c) (a, (a , a))
(d) (a,)
54. A grammer is said to be in Greibach normal form if it is free from NULL (unless S → ∈) and
every production starts with terminal. Which of the following context free Grammer is in GNF form
(a) A → bABb
(b) A → B b B b
(c) A → B a b
(d) A → A B C
55. The rule used in Non-recursive Predictive Passing if input symbol and Top symbol of stack are matched
(a) Po p t h e t o p s y mb o l o f s t a ck
(b) Pop the top symbol of stack and advance input symbol pointer
(c) A d va n c e i n p u t s y mb ol p o i nt e r
(d) Po p t h e t o p s y mb o l o f s t a ck w i t h ou t ad van c i n g i n p u t s ymb o l p oi nt e r
56. LL(1) stands
(a) S c a n n i n g t h e i n p u t s t r i n g f r o m l e f t t o r i g h t a n d a p p l y i n g r e ve r s e l e f t m os
t d e r i va t i on w i t h on e i n p u t s y mb ol l o ok a h e ad

(b) s c a n n i n g t h e i n p u t s tr i n g f r o m r i g ht to l e f t a n d ap p l y i n g l e f t m os t d e r i va ti
o n w i th on e i n p u t s y mb o l l o ok a h e a d
(c) scanning the input string from left to right and applying left most derivation with one input symbol look ahead
(d) s c a n n i n g t h e i n p u t s tr i n g f ro m l e f t to ri g ht a n d ap p l y i n g r i g ht m os t d e r i va t i
on w i t h on e i n p u t s y mb ol l o ok h e a d
57. Action of LL(1) passer if Top of the stack and current input symbols are same term in a l which is end- marker($)
(a) p a rs e r d e c l a r e s e rr o r
(b) p a rs e r h a l ts d u e t o t h e e r r o r
(c) Successful completion parser Halts
(d) Po p t h e s ymb o l f r o m s t a ck
58. The production A → ∈ is to be placed in M[A, a] entries of parsing Table i ff
(a) a ∈ FOL L OW (∈)
(b) a∈ FOLLOW(A)

(c) a ∈ F I R S T (∈)
(d) a ∈ F I R S T (A)
59. Relation between SLR,LALR, LR Parsers
(a) S L R ≤ L R ≤ L A L R
(b) L A L R ≤ L R ≤ S L R
(c) L R ≤ L A L R ≤ S L R
(d) SLR ≤ LALR≤ LR
60. Let the grammer
S1 → S #
S → AaAb
S → BbBa
A → ∈
B → ∈
Identify the correct st atement as FOLLOW(A)= FOLLOW(B)
(a) It result reduce, reduce conflict
(b) I t d e n ot e S LR (1)
(c) I t r e s u l t s h i f t , s h i f t c o n f l i c t
(d) I t r e s u l t s h i f t , r e d u c e c o n f l i c t
61. S → aABe
A → Abc | b
B → d
The sentence 'abbcde' is to be reduced to s the starting reduction
(a) B → d
(b) A → b & B → d
(c) A → A b c
(d) A → b
62. Synthesized attribute can be easily simulated by .
(a) R e gu l a r g ra m m e r
(b) L L g ra m m e r s
(c) A mb i gu o u s gr a mm e r
(d) LR grammer
63. Which of the semantic rule voilates L- attributed syntax Directed Definition Production Semantic Rule A → QRR. i= r(A. i)Q. i= q(R. S)A. s=f(Q.S)
(a) Q.i=q(R.S)
(b) R . i = r (A . i)
(c) R . i = f (Q. S)
(d) A . S = f (Q . S)
64. The expression a+b* when represented in 3- address code becomes (* has higher precedence than +)

(a) T1= b*c T2=T1+a
(b) T 1 = a T 2 = T 1 + b *c
(c) T 1 = a +b T 2 = T 1 * c
(d) T 1 = a T 1 = b T 3 = T 1 + T 2 * c
65. Pumping lemma is to be prove certaing languages are not regular. Let string W= xyz | w | = p, which belongs to language L. If string xyi z, i≥0 must be long to the language if it is regular. The conditions needed for dividing string W into 3 parts x,y z
(a) | y | > 1 , | x y | ≥ p

(b) | y | < 1 , | x y | ≤ p
(c) | y | > 0 , | x y | ≥ p
(d) |y | > 0, |xy|≤ p

66. The language L={W∈(0+1)*/W ends with 00 } The number of states in an associated N FA
(a) 2
(b) 4
(c) 5
(d) 3

67. The number of Tokens in the following linguistic statement (Ignore blanks) int a=b*c+d ;
(a) 9
(b) 8
(c) 1 2
(d) 7

68. Which of the following production belongs to CFG (Context Free Grammer)
(a) a b → A B
(b) A → aBb
(c) a A → b B
(d) A a → B b

69. The recussive descent passer is implemented with stack, input buffer and parsing Table. If there are Two entries for any place of parsing table then the grammer is
(a) ambiguous not LL(1)
(b) U n - A mb i g u ou s & n ot LL (1)
(c) A mb i gu o u s & LL (1)
(d) U n - A mb i g u ou s & L L (1)

70. S1 → S$
S → aABe
A → Abc | b
B → d | ∈
FOLLOW(S), FOLLOW(A), FOLLOW(B)
(a) { $ } , { b } , { e }
(b) { $ } , { d , b , ∈ } , { e }
(c) { $ } , { d , b } , { e }
(d) { $ } , {d, b, e}, { e}

71. F1 → F$
F → T1E1
E1 → +T1E1 | ∈
T1 → FT1
T1 →* FT1| ∈
F → (E1)|id
T1 → ∈ **is be placed in LL(1) parsing Table entries**
(a) M [T 1 , +] , M [T 1 , i d] , M [T 1 , $]
(b) M [T 1 , +] , M [T 1 , *] , M [T 1 , 0]
(c) M[T1, +], M[T1, $]
(d) M [T 1 , +] , M [T 1 , *] , M [T 1 ,)]

72. The possible actions of shift reduce - parser
i) Shift (shifts the next input symbol on stack)
ii) Reduce(handle is reduced by associated Non- terminal
iii) Accept(It the end marker is reached)
iv) Error(Activate error recovery Procedure on error)
(a) i , i i , i v
(b) i , ii , iii , iv

(c) i i , i i i , i v
(d) i , i i i , i v
73. E → (E)| E+T/T
T → T*F/F
F → (E)/ id Which of the following belongs to LR(o) of I o state
(a) F → .id
(b) F → (. E)
(c) E → E . + T
(d) E → T .
74. For which of the following situation inheritted attribute is a natural choice
(a) I m p l a c e m e nt i f c o n s t ru c t
(b) e val u a t i on o f ar t h e m a t i c e x p r e s s i o n s
(c) C h e ck i n g f o r c o r r e c t u s e of L - val u e s
(d) Keeping Track of variable declaration
75. Which of the following language is infinite over an alphabet = {0,1}
(a) s t r i n gs w i t h l e n g t h 2 o r 3
(b) s t r i n gs w i t h l e n g t h e q u a l t o 2
(c) strings with length atleast2
(d) s t r i n gs w i t h l e n g t h a t m o s t 2
76. Regular expression for a language which contain single 1 over an alphabet = { 0 , 1 }
(a) 0*10 *
(b) (1 0 *)
(c) 1
(d) (0 *1)
77. Minimum number of states in DFA to represent the regular expression ab(a+b)*
(a) i n fi n i t e
(b) 3
(c) 4
(d) 2
78. The number of states for NFA which represent regular expression 0*1+0*0
(a) 1
(b) 2
(c) 4
(d) 3
79. The seperation of analysis phase of the compiling into lexical analysis and parsing provides
(a) all the above
(b) I m p r ov i n g t h e e ffi c i e n c y o f c o m p i l e r
(c) s u p p o r t s p o r t ab i l i t y
(d) S i m p l e r D e s i g n
80. With respect to the following grammer E → E+E | E * E | id identify correct statement
(a) I t g e n e r a t e s s t r i n g s w h o s e l e n g t h i s a t m o s t t h r e e
(b) It is ambiguous grammer
(c) I t g e n e r a t e s s t r i n g s o f l e n g t h t h r e e
(d) I t g e n e r a t e s s t r i n g s w h o s e l e n g t h i s a t l e a s t t h r e e
81. Which of the following grammer generates Palindromes of even length over = {a, b}
(a) S → a s b S → a b
(b) S → b s a s → a b
(c) S → a s a | a
(d) S → asa S → bsb S → Λ
82. A → aBb\BC
B → dd

C → e|f FOLLOW(B)
(a) { d }
(b) { e , f }
(c) { b, e, f }
(d) { b , d , e , f }
83. S1 → S$
S → ABa | b
A → b| d
B → d The production S → ABa is to be placed in the following entries of parsing Table
(a) M[s, b], M[s, d]
(b) M [s , a] , M [s , d]
(c) M [S , $] , M [S , a]
(d) M [s , #] , M [s , d]
84. Which of the following statement is not correct with respect to A → w productions A → k
(a) FIRST (W) can be NULL
(b) F I R S T (A) c an b e N U L L
(c) F I R S T (A) c ont ai n s F I R S T (W)
(d) F I R S T (W) c ont ai n s F I R S T (A)
85. Which of the following passer requires back-Tracking
(a) Recursive Descent Passer
(b) R e c u r s i ve p r e d i c t i ve Pa s s e r
(c) L L (1) Pa s s e r
(d) N o n - r e c u r s i ve P re di c ti ve p as s e r
86. Action of LL(1) passer if Top of the stack is a non-terminal X and current input symbol is terminal, a
(a) G e t th e c o nte nt of p as s i n g Ta b l e e nt r y M [x, a] p u s h i t o n t o t h e s t a ck a n d a d va
nc e i n p u t s ymb o l p o i nt e r
(b) a d va n c e i n p u t s ymb o l p oi nt e r
(c) Po p - t h e N o n - t e r m i n a l f r o m s t a ck
(d) Get the content of passing Table ex try M [x, a] and push it on to the stack without advancing input symbol pointer
87. The 'Handle' of wrtto shift- reduce passers
(a) I t i s t h e e n d m a rke r
(b) It is the input string matches with right hand side of the production
(c) I t i s t h e i n p u t s t r i n g m a t ch i n g w i t h s t r i n g N on - te rm i n a l
(d) I t i s t h e i n p u t s t r i n g m a t ch e s w i th l e f t h a n d s i d e o f t h e p r o d u c t i on
88. The graph dependencies of the attributes of different nodes of a pars e Tree is called as
(a) K a r n au g h G r a p h
(b) S t e fi G ra p h
(c) F l ow G r a p h
(d) Dependency Graph
89. Consider the following syntax Directed Definition Production Semantic Rule
D→ TLL. in = T. type
T→ int/real T. type=integer, T. type= real
L→ L, id L1. in= Lin, add type (id.entry, L.in)
L → id add type (id. entry, L. in) The nature of attributes of T&L
(a) I n h e r i t t e d , I n h e r i t t e d
(b) synthesized, Inheritted
(c) s y nt h e s i z e d , s y nth e r i z e d
(d) I n h e r i tt e d , s y nth e r i z e d

90. Let the Translation Rule for YACC program statement is given below
% % E1: E1 + T{$$=$1+$3;} The $1,$3, denote
(a) va l u e s of E 1 , T
(b) values of E, T
(c) va l u e s +, T
(d) va l u e s of + , E 1