GATE SOLVED PAPER - CS

COMPILER DESIGN

YEAR 2001

Q. 1 Which of the following statements is false ?

(A) An unambiguous grammar has same left most and right most derivation

(B) An $LL(1)$ parser is a top-down parser

(C) $LALR$ is more powerful than SLR

(D) An ambiguous grammar can never be $LR(K)$ for any k

YEAR 2002

Q. 2 Dynamic linking can cause security concerns because

(A) Security is dynamic

(B) The path for searching dynamic libraries is not known till run time.

(C) Linking is insecure

(D) Cryptographic procedures are not available for dynamic linking

YEAR 2003 ONE MARK

Q. 3 Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?

(A) Removing left recursion alone

(B) Factoring the grammar alone

(C) Removing left recursion and factoring the grammar

(D) None of this

Q. 4 Assume that the SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between n_1 and n_2 is

(A) n_1 is necessarily less than n_2 (B) n_1 is necessarily equal to n_2

(C) n_1 is necessarily greater than n_2 (D) None of the above

Q. 5 In a bottom-up evaluation of a syntax directed definition, inherited attributes can

(A) always be evaluated

(B) be evaluated if the definition is L-attributed

(C) be evaluated only if the definition has synthesized attributes

(D) never be evaluated

Q. 6 Which of the following statements is FALSE?

(A) In statically typed language, each variable in a program has a fixed type

(B) In up-typed languages, values do not have any types

(C) In dynamically typed languages, variables have no types

(D) In all statically typed languages, each variable in a program is associated with values of only a single type during the execution of the program

YEAR 2003 TWO MARKS

Q. 7 Consider the grammar shown below

$$S \rightarrow | EtSS' | \alpha$$
$$S' \rightarrow eS | \in$$
$$E \rightarrow b$$

In the predictive parse table. M, of this grammar, the entries $M[S', \varepsilon]$ and $M[S', \$]$ respectively are

(A) $\{s' \rightarrow eS\}$ and $\{S' \rightarrow \varepsilon\}$

(B) $\{s' \rightarrow eS\}$ and $\{\}$

(C) $\{s' \rightarrow \varepsilon\}$ and $\{S' \rightarrow \varepsilon\}$

(D) $\{s' \rightarrow eS, S' \rightarrow \varepsilon\}$ and $\{S' \rightarrow \varepsilon\}$

Q. 8 Consider the grammar shown below.

$$S \rightarrow C\ C$$
$$C \rightarrow eC | d$$

The grammar is

(A) LL (1)

(B) SLR (1) but not LL (1)

(C) LALR (1) but not SLR (1)

(D) LR (1) but not LALR (1)

Q. 9 Consider the translation scheme shown below

$S \rightarrow TR$

$R \rightarrow + T$ {print ('+');} $R | \varepsilon$

$T \rightarrow$ num {print (num.val);}

Here num is a token that represents an integer and num. val represents the corresponding integer value. For an input string '9 + 5+ 2', this translation scheme will print

(A) $9+5+2$ (B) $95+2+$

(C) $952++$ (D) $++952$

Q. 10 Consider the syntax directed definition shown below

```
S → id: = E          {gen (id.place = E.place;);}
E → E1 + e2          {t = newtemp ();
                      gen(t=E1. place +E2. place;);
                      E.place = t}
E → id               {E.place = id.place;}
```

Here, gen is a function that generates the output code, and newtemp is a function that returns the name of a new temporary variable on every call. Assume that t_1 's are the temporary variable names generated by newtemp.

For the statement '$X := Y + Z$', the 3-address code sequence generated by this definition is

(A) $X = Y + Z$

(B) $t_1 = Y + Z; X\ t_1$

(C) $t_1 = Y; t_2 = t_1 + Z; X = t_2$

(D) $t_1 = Y; t_2 = Z; t_3 + t_2; X = t_3$

Common Data For Q. 11 & 12

Solve the problems and choose the correct answers.
The following program fragment is written in a programming language that allows variables and does not allow nested declarations of functions.

```
global int i=100, j=5;
void P(x){
          int i=10
          print (x+10);
          i+200;
          j=20;
          print (x);
}
main () {P(i+j);}
```

Q. 11 If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are
(A) 115, 220
(B) 25, 220
(C) 25, 15
(D) 115, 105

Q. 12 If the programming language uses dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are
(A) 115, 220
(B) 25, 220
(C) 25, 15
(D) 115, 105

Q. 13 Consider the following class definitions in a hypothetical object oriented language that supports inheritance and uses dynamic binding. The language should not be assumed to be either Java or C++, thought the syntax is similar

```
Class     P {       Class Q subclass of P {
     void f(int i) {     void f (int i) {
          print (i);          print (2*i)I;
     }                   }
}                   }
```

Now consider the following program fragment:
$P\ x = \text{new } Q();$
$Q\ y = \text{new } Q();$
$P\ z = \text{new } Q();$
$x.f(1);((P)y).f(1);z.f(1);$

Here $((P)y)$ denotes a typecast of y to P. The output produced by executing the above program fragment will be
(A) 1 2 1 (B) 2 1 1
(C) 2 1 2 (D) 2 2 2

Q. 14 Which of the following is NOT an advantage of using shared, dynamically linked libraries as opposed to using statically linked libraries?

(A) Smaller sizes of executable

(B) Lesser overall page fault rate in the system

(C) Faster program startup

(D) Existing programs need not be re-linked to take advantage of newer versions of libraries

YEAR 2004 ONE MARK

Q. 15 Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are non-terminals, and r, s, t are terminals .

(i) $P \rightarrow QR$ (ii) $P \rightarrow Q\ s\ R$

(iii) $P \rightarrow \varepsilon$ (iv) $P \rightarrow Q\ t\ R\ r$

(A) (i) only (B) (i) and (iii) only

(C) (ii) and (iii) only (D) (iii) and (iv) only

Q. 16 Consider a program P that consists of two source modules M_1 and M_2 contained in two different files. If M_1 contains a reference to a function defined in M_2, the reference will be resolved at

(A) Edit-time (B) Compile-time

(C) Link-time (D) Load-time

Q. 17 Consider the grammar rule $E \rightarrow E_1 - E_2$ for arithmetic expressions. The code generated is targeted to a CPU having a single user register. The subtraction operation requires the first operand to be in the register. If E_1 and E_2 do not have any common sub expression, in order to get the shortest possible code

(A) E_1 should be evaluated first

(B) E_2 should be evaluated first

(C) Evaluation of E_1 and E_2 should necessarily be interleaved

(D) Order of evaluation of E_1 and E_2 is of no consequence

YEAR 2004 TWO MARKS

Q. 18 Consider the grammar with the following translation rules and E as the start symbol.

```
E → E1#T        {E.value = E1.value * T.value}
|T              {E.value = T.value}
T → T1&F        {T.value = T1.value + F.value}
|F              {T.value = F.value}
F → num         {F.value =num.value}
```

Compute E. value for the root of the parse tree for the expression: 2 # 3 # & 5 # 6 & 4.

(A) 200 (B) 180

(C) 160 (D) 40

YEAR 2005 ONE MARK

Q. 19 The grammar $A \rightarrow AA \mid (A) \mid \varepsilon$ is not suitable for predictive-parsing

because the grammar is

(A) ambiguous
(B) Left-recursive
(C) right-recurisve
(D) an operator-grammar

YEAR 2005 TWO MARKS

Q. 20 Consider the grammar $E \rightarrow E + n \mid E \times n \mid n$
For a sentence $n + n$, the handles in the right-sentential form of the reduction are

(A) $n, E + n$ and E + n × n
(B) $n, E + n$ and E + E × n
(C) $n, n + n$ and n + n × n
(D) $n, E + n$ and E × n

Q. 21 Consider the grammar
$S \rightarrow (S) \mid a$
Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar $n_1 n_2$ and n_3 respectively. The following relationship holds good

(A) $n_1 < n_2 < n_3$
(B) $n_1 = n_3 < n_2$
(C) $n_1 = n_2 = n_3$
(D) $n_1 \geq n_3 \geq n_2$

Q. 22 Consider line number 3 of the following C-program.

```
int main (){                    | * Line 1 * |
int 1, N;                       | * line  2 * |
fro (i=0, 1 < N, 1++);          | * Line 3 * |
}
```

Identify the compiler's response about this line while creating the object-module

(A) No compilation error
(B) Only a lexical error
(C) Only syntactic errors
(D) Both lexical and syntactic errors

Common Data For Q. 23 & 24

Solve the problems and choose the correct answers.
Consider the following expression grammar. The semantic rules for expression calculation are stared next to each grammar production.

```
E →  number     Eval       = number.val
     |E'+'B     E(1).val   =E(2).VAL +E(3).val
     |E'×'E     E(1).val   =E(2).VAL +E(3).val    ;
```

Q. 23 The above grammar and the semantic rules are fed to a yacc tool (which is an LALR(1) parser generator) for parsing and evaluating arithmetic expressions. Which one of the following is true about the action of yacc for the given grammar?

(A) It detects recursion and eliminates recursion
(B) It detects reduce-reduce conflict, and resolves
(C) It detects shift-reduce conflict, and resolves the conflict in favor of a shift over a reduce action
(D) It detects shift-reduce conflict, and resolves the conflict in favor of a reduce over a shift action

Q. 24 Assume the conflicts part (*a*) of this question are resolved and an LALR(1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression $3 \times 2 + 1$. What precedence and associativity properties does the generated parser realize?

(A) Equal precedence and left associativity; expression is evaluated to 7

(B) Equal precedence and right associativity, expression is evaluated to 9

(C) Precedence of '*x*' is higher than that of '+', and both operators are left associative; expression is evaluated to 7

(D) Precedence of '$\times$' is higher than that of '$\times$', and both operators are left associative; expression is evaluated to 9

YEAR 2006 ONE MARK

Q. 25 Consider the following grammar.

$$S \rightarrow S * E$$
$$S \rightarrow E$$
$$E \rightarrow F + E$$
$$E \rightarrow F$$
$$F \rightarrow id$$

Consider the following $LR(0)$ items corresponding to the grammar above.

(i) $S \rightarrow S * .E$

(ii) $E \rightarrow F. + E$

(iii) $E \rightarrow F + .E$

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

(A) (i) and (ii)

(B) (ii) and (iii)

(C) (i) and (iii)

(D) None of these

YEAR 2006 TWO MARKS

Q. 26 Consider the following grammar

$$S \rightarrow FR$$
$$R \rightarrow * S \mid \varepsilon$$
$$F \rightarrow id$$

In the predictive parser table, M, of the grammar the entries $M[S, id]$ and $M[R,\$]$ respectively

(A) $\{S \rightarrow FR\}$ and $\{R \rightarrow \varepsilon\}$

(B) $\{S \rightarrow FR\}$ and $\{\}$

(C) $\{S \rightarrow FR\}$ and $\{R \rightarrow * S\}$

(D) $\{F \rightarrow id\}$ and $\{R \rightarrow \varepsilon\}$

Q. 27 Consider the following translation scheme.

```
S → ER
R → *E{print{'*');R | ε
E → F+E{print('+'); | F
F → (S) | id{print(id.value);}
```

Here *id* is a taken that represents an integer and *id*. value represents the corresponding integer value. For an input '2 * 3 + 4', this translation scheme prints

(A) 2 * 3 + 4

(B) 2 * + 3 4

(C) 2 3 * 4 +

(D) 2 3 4 + *

Q. 28 Consider the following C code segment.

```
for (i=0;i<N;i++)
     for (J=0;j<N;J++)
          if (i%2)
               (x+=(4*j+s×i);
                    y+=(7+4*j);
                    }
          }
}
```

Which one to the following false?

(A) The code contains loop-in variant computation

(B) There is scope of common sub-expression elimination in this code

(C) There is scope strength reduction in this code

(D) There is scope of dead code elimination in this code

Q. 29 Which one of the following grammars generates the language $L = (a^i b^i \mid i \neq j\}$?

(A) $S \to AC \mid CB$
$C \to aCb \mid a \mid b$
$A \to aA \mid \varepsilon$
$B \to Bb \mid \varepsilon$

(B) $S \to aS \mid Sb \mid a \mid b$

(C) $S \to ACCB$
$C \to aCb \mid \in$
$A \to aA \mid \in$
$B \to Bb \mid \in$

(D) $S \to AC \mid CB$
$C \to aCb \mid \in$
$A \to aA \mid a$
$B \to bB \mid b$

Q. 30 In the correct grammar above, what is the length of the derivation (number of steps starting from S to generate the string $a^l b^m$ with $l \neq m$?

(A) $\max(l, m) + 2$

(B) l+m+2

(C) $l + m + 3$

(D) $\max(l, m) + 3$

YEAR 2007 ONE MARK

Q. 31 Which one of the following is a top-down parser?
(A) Recursive descent parser
(B) Operator precedence parser
(C) An LR(k) parser
(D) An LALR(k) parser

YEAR 2007 TWO MARKS

Q. 32 Consider the grammar with non-terminals $N = \{S, C, S\}$, terminals $T = \{a, b, i, t, e\}$, with S as the start symbol, and the following of rules
$S \rightarrow iCtSS_1 \mid a$
$S_1 \rightarrow eS \mid \varepsilon$

$C \rightarrow b$
The grammar is NOTLL(1) because:
(A) It is left recursive
(B) It is right recursive
(C) It is ambiguous
(D) It is not context-free

Q. 33 Consider the following two statements:
P: Every regular grammar is LL(1)
Q: Every regular set has LR(1) grammar
Which of the following is TRUE?
(A) Both P and Q are true
(B) P is true and Q is false
(C) P is false and Q is true
(D) Both P and Q are false

Q. 34 In a simplified computer the instructions are:
$OP\ R_j, R_i$ – Performs R_j OP R_i and stores the result in register R_i
$OP\ m, R_i$ – Performs val OP R_i abd stores the result in R_i. value denotes the content of memory location m.
$MCVm, R_i$ –Moves the content off memory loction m to register R_i.
$MCVm, Ri, m$ –Moves the content of register R_i to memory location m.
The computer has only two registers, and OP is either ADD or SUB. Consider the following basic block:

$$t_1 = a + b$$
$$t_2 = c + d$$
$$t_3 = e - t_2$$
$$t_4 = t_1 - t_2$$

Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?
(A) 2 (B) 3
(C) 5 (D) 6

Common Data For Q. 35 & 36

Solve the problems and choose the correct answers.

Consider the CFG with $\{S, A, B\}$ as the non-terminal alphabet, $\{a, b\}$ as the terminal alphabet, S as the start symbol and the following set of production rules

$s \rightarrow bA$ $A \rightarrow a$ $A \rightarrow aS$
$A \rightarrow bAA$ $S \rightarrow aB$ $B \rightarrow b$
$B \rightarrow bS$ $B \rightarrow aBB$

Q. 35 Which of the following strings is generated by the grammar?

(A) *aaaabb* (B) *aabbbb*

(C) *aabbab* (D) *abbbba*

Q. 36 For the correct answer string to Q. 9 how many derivation trees are there?

(A) 1 (B) 2

(C) 3 (D) 4

YEAR 2008 ONE MARK

Q. 37 Which of the following describes a handle (as applicable to LR-parsing) appropriately?

(A) It is the position in a sentential form where the next shift or reduce operation will occur

(B) It is a non-terminal whose production will be used for reduction in the next step

(C) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.

(D) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found

Q. 38 Some code optimizations are carried out on the intermediate code because

(A) They enhance the portability of the complier to other target processors

(B) Program analysis is name accurate on intermediate code than on machine code

(C) The information from data flow analysis cannot otherwise be used for optimization

(D) The information from the front end cannot otherwise be used for optimization

YEAR 2008 TWO MARKS

Q. 39 Which of the following are true?

(i) A programming language option does not permit global variables of any king and has no nesting of procedures/functions, but permits recursion can be implemented with static storage allocation

(ii) Multi-level access link (or display) arrangement is needed to arrange activation records-only if the programming language being implemented has nesting of procedures/function

(iii) Recursion in programming languages cannot be implemented with dynamic storage allocation

(iv) Nesting of procedures/functions and recursion require a dynamic heap allocation scheme and cannot be implemented with a stack-based allocation scheme for activation records

(v) Programming languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records

(A) (ii) and (v) only
(B) (i), (iii) and (iv) only
(C) (i), (ii) and (v)
(D) (ii), (iii) and (v) only

Q. 40 An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if

(A) The SLR(1) parser for G has S-R conflicts
(B) The LR(1) parser for G has S-R conflicts
(C) The LR(0) parser for G has S-R conflicts
(D) The LALR(1) parser for G has reduce-reduce conflicts

YEAR 2009 ONE MARK

Q. 41 Which of the following statements are TRUE ?

I. There exist parsing algorithms for some programming languages whose complex are less than $\theta(n^3)$

II A programming language which allows recursion can be implemented with static storage allocation

III No L-attributed definition can be evaluated in the framework of bottom-up parsing

IV Code improving transformations can be performed at both source language and intermediate code level

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

YEAR 2010 ONE MARK

Q. 42 What data structure in a complier is used for managing information about variables and their attributes?

(A) Abstract syntax tree
(B) Symbol table
(C) Semantic stack
(D) Parse table

Q. 43 Which languages necessarily need heap allocation in the runtime environment ?

(A) Those that support recursion
(B) Those that use dynamic scoping
(C) Those that allow dynamic data structure
(D) Those that use global variables

YEAR 2010 TWO MARKS

Q. 44 The grammar $S \rightarrow aSA|bS|c$ is

(A) LL (1) but not LR (1)

(B) LR (1) but not LL(1)

(C) Both LL (1) and LR (1)

(D) Neither LL (1) nor LR (1)

YEAR 2011 ONE MARK

Q. 45 In a compiler, keywords of a language are recognized during

(A) parsing of the program

(B) the code generation

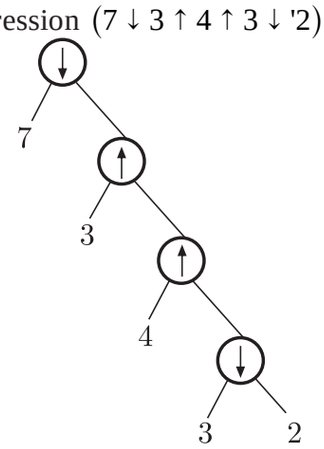
(C) the lexical analysis of the program

(D) dataflow analysis

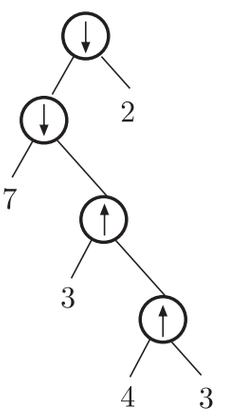
YEAR 2011 TWO MARKS

Q. 46 Consider two binary operators '↑' and '↓' with the precedence of operator ↓ being lower than that of the operator ↑. Operator ↑ is right associative while operator ↓ is left associative. Which one of the following represents the parse tree for expression $(7 \downarrow 3 \uparrow 4 \uparrow 3 \downarrow '2)$

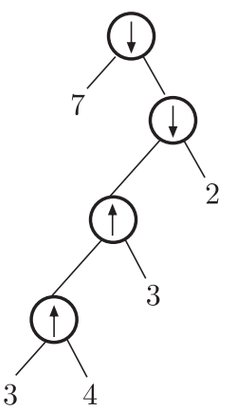
(A)



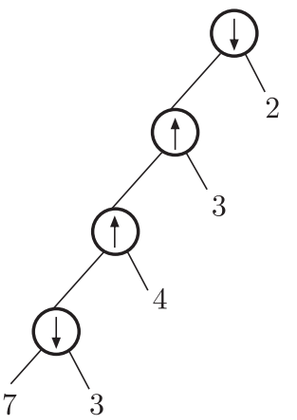
(B)



(C)



(D)



The designer of the system also has an alternate approach of using the DMA controller to implement the same transfer. The DMA controller requires 20 clock cycles for initialization and other overhead. Each DMA transfer cycle takes two clock cycles to transfer one byte of data from interrupt driven program based input-output?

(A) 3.4

(B) 4.4

(C) 5.1

(D) 6.7

Statement Data For Q. 47 and 48

For the grammar below, a partial $LL(1)$ parsing table is also presented along with the grammar. Entries that need to be filled are indicated as $E1$, $E2$, and $E3$. ε is the empty string, $\$$ indicates end of input, and, separates alternate right hand sides of productions.

$$S \rightarrow aAbB, bAaB, \varepsilon$$
$$A \rightarrow S$$
$$B \rightarrow S$$

	a	b	c
S	$E1$	$E2$	$S \rightarrow \varepsilon$
A	$A \rightarrow S$	$A \rightarrow S$	error
B	$B \rightarrow S$	$B \rightarrow S$	$E3$

Q. 47 The FIRST and FOLLOW sets for the non-terminals A and B are

(A) FIRST (A) $= \{a, b, \varepsilon\}$ = FIRST (B),
FOLLOW (A) $= \{a, b\}$

(B) FIRST (A) $= \{a, b, \$\}$
FIRST (B) $= \{a, b, \varepsilon\}$
FOLLOW (A) $= \{a, b\}$
FOLLOW (B) $= \{\$\}$

(C) FIRST (A) $= \{a, b, \varepsilon\}$ =FIRST (B),
FOLLOW (A) $= \{a, b\}$
FOLLOW (B) $= \emptyset$

(D) FIRST (A) $= \{a, b\}$ = FIRST (B)
FOLLOW (A) $= \{a, b\}$
FOLLOW (B) $= \{a, b\}$

Q. 48 The appropriate entries for $E1$, $E2$ and $E3$ are

(A) $E1: S \rightarrow aAbB, A \rightarrow S$
$E2: S \rightarrow bAaB, B \rightarrow S$
$E3: B \rightarrow S$

(B) $E1: S \rightarrow aAbB, S \rightarrow \varepsilon$
$E2: S \rightarrow bAaB, B \rightarrow \varepsilon$
$E3: S \rightarrow \varepsilon$

(C) $E1: S \rightarrow aAbB, S \rightarrow \varepsilon$
$E2: S \rightarrow bAaB, S \rightarrow \varepsilon$
$E3: B \rightarrow S$

(D) $E1: A \rightarrow S, S \rightarrow \varepsilon$
$E2: B \rightarrow S, S \rightarrow \varepsilon$
$E3: B \rightarrow S$

****

ANSWER KEY

Compiler Design									
1	2	3	4	5	6	7	8	9	10
(A)	(D)	(C)	(B)	(C)	(C)	(D)	(D)	(B)	(D)
11	12	13	14	15	16	17	18	19	20
(D)	(A)	(D)	(C)	(B)	(C)	(B)	(C)	(A)	(D)
21	22	23	24	25	26	27	28	29	30
(B)	(C)	(C)	(C)	(C)	(A)	(D)	(D)	(D)	(A)
31	32	33	34	35	36	37	38	39	40
(A)	(A)	(A)	(C)	(C)	(B)	(D)	(B)	(B)	(B)
41	42	43	44	45	46	47	48		
(B)	(B)	(C)	(C)	(C)	(B)	(A)	(C)		