Edition 2021-22

Compiler Design

PEN-Drive / G-Drive Course / VOD & Tablet Users

Workbook

Computer Science Engineering

GATE / ESE / PSUs

Ajay Das

GATE Since 2004

GATE ACADEMY®

steps to success...

Compiler Design

PEN-Drive / G-Drive Course / VOD & Tablet Users

Workbook

Computer Science

GATE ACADEMY ®

Edition : 2021-22

GATE ACADEMY ®

A/114-115, Smriti Nagar, Bhilai - 490 020 (C.G.)

Phone : 0788 - 4034176, 0788 - 3224176

Help Desk No. - +91-97131-13156

For Feedback & Suggestions...

info@gateacademy.co.in

GATE Syllabus

Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow analyses: constant propagation, liveness analysis, common subexpression elimination.

Table of Contents

Video Lecture Information

Sr.	Lecture Name	Duration
Chapter 01 Introduction of Compiler Design, Lexical Analysis & Syntax Analysis		
Lecture 01	Introduction of Compiler (Part 1)	**0:32:43**
Lecture 02	Introduction of Compiler (Part 2)	**0:19:14**
Lecture 03	Introduction of Compiler (Part 3)	**0:39:18**
Lecture 04	Introduction of Grammar & Languages (Part 1)	**1:03:52**
Lecture 05	Introduction of Grammar & Languages (Part 2)	**0:26:17**
Lecture 06	Introduction of Grammar & Languages (Part 3)	**0:20:00**
Lecture 07	Introduction of Grammar & Languages (Part 4)	**0:37:06**
Lecture 08	Workbook Questions (15,16,17,18)	**0:18:59**
Lecture 09	Workbook Questions (19,20,21,22)	**0:26:23**
Lecture 10	Lexical Analysis	**0:41:40**
Lecture 11	Workbook Questions (1,2,3,4,5,6)	**0:16:44**
Lecture 12	Workbook Questions (7,8,9,10,11)	**0:18:29**
Lecture 13	Workbook Questions (12,13)	**0:10:48**
Lecture 14	Calculation of FIRST & FOLLOW Set (Part 1)	**0:31:42**
Lecture 15	Calculation of FIRST & FOLLOW Set (Part 2)	**0:36:22**
Lecture 16	Workbook Questions (23,24,25,26)	**0:20:31**
Lecture 17	Introduction of Top-Down Parsing	**0:53:32**
Lecture 18	LL(1) Parsing (Part 1)	**0:40:48**
Lecture 19	LL(1) Parsing (Part 2)	**0:41:34**
Lecture 20	LL(1) Parsing (Part 3)	**0:28:05**
Lecture 21	Workbook Questions (14,30,32,33,37,38,40)	**0:25:53**
Lecture 22	Introduction of Bottom Up Parsing	**0:43:31**
Lecture 23	LR(0) Parsing	**0:56:24**
Lecture 24	SLR Parsing (Part 1)	**0:48:25**
Lecture 25	SLR Parsing (Part 2)	**0:11:51**
Lecture 26	Introduction of LR(1) Parsing	**0:26:16**
Lecture 27	Construction of CLR Parse Table	**0:30:34**
Lecture 28	Check CLR (Part 1)	**0:27:59**
Lecture 29	Check CLR (Part 2)	**0:32:20**
Lecture 30	LALR Parser (Part 1)	**0:22:14**
Lecture 31	LALR Parser (Part 2)	**0:21:01**
Lecture 32	Operator Precedence Parser	**0:35:00**
Lecture 33	Operator Precedence Graph	**0:36:37**
Lecture 34	Workbook Questions (27,28,29,31,34,35)	**0:33:55**
Lecture 35	Workbook Questions (36,39,41,42,45,46)	**0:29:22**
Lecture 36	Workbook Questions (43,44,49,50,51)	**0:25:51**
Lecture 37	Workbook Questions (47,48,52,53)	**0:08:30**
Lecture 38	FTR (Lexical Analysis)	**0:31:10**

Lecture 39	FTR (Grammars, FIRSt & FOLLOW Set)	**0:28:22**
Lecture 40	FTR (Top Down Parsing)	**0:39:33**
Lecture 41	FTR (Bottom Up Parsing)	**0:53:53**
Chapter 02 Semantic Analysis		
Lecture 01	Introduction of Semantic Analysis	**0:37:59**
Lecture 02	SDT (Part 1)	**0:43:06**
Lecture 03	SDT (Part 2)	**0:44:30**
Lecture 04	Type of SDT	**0:32:39**
Lecture 05	Workbook Questions (1,2,3,4,5)	**0:14:40**
Lecture 06	Workbook Questions (6,7)	**0:15:09**
Lecture 07	Workbook Questions (8,9,12,14,15)	**0:17:29**
Lecture 08	Workbook Questions (10,11,13,16,17,18,19)	**0:34:20**
Lecture 09	FTR (Semantic Analysis)	**0:38:53**
Chapter 03 Intermediate Code Generation		
Lecture 01	Introduction of Intermediate Code Generation	**0:53:30**
Lecture 02	Three Address Code	**0:20:33**
Lecture 03	Directed Acyclic Graph (DAG)	**0:42:29**
Lecture 04	Code Optimization	**0:23:47**
Lecture 05	Data Flow Analysis	**0:33:30**
Lecture 06	RunTime Environment	**0:12:05**
Lecture 07	Workbook Questions (4,6,7,8,9)	**0:31:39**
Lecture 08	Workbook Questions (10,11,12,13,15)	**0:12:41**
Lecture 09	Workbook Questions (16,18,19,20)	**0:29:34**
Lecture 10	Workbook Questions (17,21,22)	**0:24:50**
Lecture 11	Workbook Questions (1,2,3,5,14,23)	**0:28:13**
Lecture 12	Fast Track Revision (FTR)	**0:41:33**
Chapter 04 Complete Revision of Compiler Design by Best Selected Questions		
Lecture 01	Introduction of Compiler Design (1,2,3,4,5,6)	**0:26:51**
Lecture 02	Lexical Analysis (1,2,3,4,5,6,7)	**0:23:29**
Lecture 03	Syntax Analysis Set 1 (1,2,3,4,5)	**0:25:08**
Lecture 04	Syntax Analysis Set 2 (6,7,8,9,10)	**0:34:06**
Lecture 05	Syntax Analysis Set 3 (11,12,13,14,15,16)	**0:25:52**
Lecture 06	Semantic Analysis (1,2,3,4,5,6,7)	**0:26:53**
Lecture 07	Code Generation & Optimization Set 1 (1,2,3,4,5,6,7)	**0:30:55**
Lecture 08	Code Generation & Optimization Set 2 (8,9,10,11,12,13)	**0:35:29**

1 Lexical & Syntax Analysis

Classroom Practice Questions

Q.1 In a compiler the module that checks every character of the source text is called:

(A) The code generator

(B) The code optimizer

(C) The lexical analyzer

(D) The syntax analyzer

[GATE 1989 : IIT Kanpur]

Q.2 Match the following:

List - I

(a) Lexical Analysis

(b) Code Optimization

(c) Code Generation

(d) Abelian Group

List - II

(p) DAG's

(q) Syntax trees

(r) Push Down automata

(s) Finite automata

(A) a-s, b-p, c-q, d-r (B) a-r, b-p, c-q, d-s

(C) a-s, b-q, c-p, d-r (D) a-s, b-p, c-r, d-q

[GATE 1991 : IIT Madras]

Q.3 In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denotes the set of letters and digits respectively, which of the following expressions defines an identifier?

(A) $(L \cup D)^+$ (B) $L(L \cup D)^*$

(C) $(L.D)^*$ (D) $L(L.D)^*$

[GATE 1995 : IIT Kanpur]

Q.4 Type checking is normally done during

(A) Lexical analysis

(B) Syntax analysis

(C) Syntax directed translation

(D) Code optimization

[GATE 1998 : IIT Delhi]

Q.5 The number of tokens in the FORTRAN statement **DO 10 I = 1.25** is

(A) 3

(B) 4

(C) 5

(D) None of the above

[GATE 1999 : IIT Bombay]

Q.6 The number of tokens in the following C statement is

printf ("$i = \%d, \& i = \%x$", $i, \& i$);

(A) 33 (B) 26

(C) 10 (D) 21

[GATE 2000 : IIT Kharagpur]

Q.7 Consider line number 3 of the following C - program

```
int main ( ) {          /*Line 1 * /
int I, N;               /*Line 2 * /
for (I = 0; I < N; I++); /*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.

[GATE 2005 : IIT Bombay]

Q.8 Match all items in Group 1 with correct option from those given in Group 2

Group 1

P. Regular expression

Q. Pushdown automata

R. Dataflow analysis

S. Register allocation

Group 2

1. Syntax analysis
2. Code generation
3. Lexical analysis
4. Code optimization

Codes :

(A) $P-4, Q-1, R-2, S-3$

(B) $P-3, Q-1, R-4, S-2$

(C) $P-3, Q-4, R-1, S-2$

(D) $P-2, Q-1, R-4, S-3$

[GATE 2009 : IIT Roorkee]

Q.9 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

[GATE 2011 : IIT Madras]

Q.10 Match the following :

List-I

(P) Lexical analysis

(Q) Top down parsing

(R) Semantic analysis

(S) Runtime environment

List-II

(i) Leftmost derivation

(ii) Type checking

(iii) Regular expressions

(iv) Activation records

(A) $P \leftrightarrow i, Q \leftrightarrow ii, R \leftrightarrow iv, S \leftrightarrow iii$

(B) $P \leftrightarrow iii, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iv$

(C) $P \leftrightarrow ii, Q \leftrightarrow iii, R \leftrightarrow i, S \leftrightarrow iv$

(D) $P \leftrightarrow iv, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iii$

[GATE 2016 : IISc Bangalore]

Q.11 Match the following according to input (from the left column) to the compiler phase (in the right column) that process is :

List-I

(P) Syntax tree

(Q) Character stream

(R) Intermediate representation

(S) Token stream

List-II

(i) Code generator

(ii) Syntax analyzer

(iii) Semantic analyzer

(iv) Lexical analyzer

(A) $P \rightarrow (ii), Q \rightarrow (iii), R \rightarrow (iv), S \rightarrow (i)$

(B) $P \rightarrow (ii), Q \rightarrow (i), R \rightarrow (iii), S \rightarrow (iv)$

(C) $P \rightarrow (iii), Q \rightarrow (iv), R \rightarrow (i), S \rightarrow (ii)$

(D) $P \rightarrow (i), Q \rightarrow (iv), R \rightarrow (ii), S \rightarrow (iii)$

[GATE 2017 : IIT Roorkee]

Q.12 A lexical analyzer uses the following patterns to recognize three tokens T1, T2, and T3 over the alphabet {a,b,c}.

T1: a?(b|c)*a

T2: b?(a|c)*b

T3: c?(b|a)*c

Note that 'x?' means 0 or 1 occurrence of the symbol x. Note also that the analyzer outputs the token that matches the longest possible prefix.

If the string bbaacabc is processes by the analyzer, which one of the following is the sequence of tokens it outputs?

(A) T1T2T3 (B) T1T1T3

(C) T2T1T3 (D) T3T3

[GATE 2018 : IIT Guwahati]

Q.13 Which one of the following statements is FALSE?

(A) Context-free grammar can be used to specify both lexical and syntax rules.

(B) Type checking is done before parsing.

(C) High-level language programs can be translated to different Intermediate Representations.

(D) Arguments to a function can be passed using the program stack.

[GATE 2018 : IIT Guwahati]

Q.14 The grammar $A \rightarrow AA|(A)|\varepsilon$ is not suitable for predictive parsing because the grammar is

(A) Ambiguous
(B) Left recursive
(C) Right recursive
(D) An operator grammar

Common Data for Questions 15 & 16

Consider the context-free grammar

$E \rightarrow E+E$

$E \rightarrow (E*E)$

$E \rightarrow id$

where E is the starting symbol, the set of terminals is {id, (,+,),*}, and the set of non-terminals is {E}.

Q.15 Which of the following terminal strings has more than one parse tree when parsed according to the above grammar?

(A) id + id + id + id
(B) id + (id* (id * id))
(C) (id* (id * id)) + id
(D) ((id * id + id) * id)

[GATE 2005 : IIT Bombay]

Q.16 For the terminal string with more than one parse tree obtained as solution to in above question, how many parse trees are possible?

(A) 5 (B) 4
(C) 3 (D) 2

[GATE 2005 : IIT Bombay]

Q.17 Which one of the following grammars generates the languages $L = (a^i b^j | i \neq j)$?

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

(B) $S \rightarrow aS|Sb|a|b$

(C) $S \rightarrow AC|CB$
$C \rightarrow aCb|\varepsilon$
$A \rightarrow aA|\varepsilon$
$B \rightarrow Bb|\varepsilon$

(D) $S \rightarrow AC|CB$
$C \rightarrow aCb|\varepsilon$
$A \rightarrow aA|a$
$B \rightarrow Bb|b$

[GATE 2006 : IIT Kharagpur]

Q.18 In the correct grammar above, what is the length of the derivation (number of steps starting from S) to generates the strings $a^l b^m$ with $l \neq m$

(A) $\max(l,m)+2$ (B) $l+m+2$
(C) $l+m+3$ (D) $\max((l,m)+3$

[GATE 2006 : IIT Kharagpur]

Common Data for Questions 19 & 20

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.

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

Q.19 Which of the following strings is generated by the grammar?

(A) aaaabb (B) aabbbb
(C) aabbab (D) abbbba

[GATE 2007 : IIT Kanpur]

Q.20 How many derivation trees are there?

(A) 1 (B) 2
(C) 3 (D) 4

[GATE 2007 : IIT Kanpur]

Q.21 What is the maximum number of reduce moves that can be taken by a bottom up parser for a grammar with no epsilon and unit production (i.e. of type $A \rightarrow \varepsilon$ and $A \rightarrow a$) to parse a string with n tokens?

(A) $n/2$ (B) $n-1$
(C) $2n-1$ (D) 2^n

[GATE 2013 : IIT Bombay]

Q.22 Consider the grammar defined by the following production rules, with two operators* and +

$S \to T * P$

$T \to U \mid T * U$

$P \to Q + P \mid Q$

$Q \to id$

$U \to id$

Which one of the following is TRUE?

(A) + is left associative, while * is right associative

(B) + is right associative, while * is left associative

(C) Both + and * are right associative

(D) Both + and * are left associative

[GATE 2014 : IIT Kharagpur]

Common Data for Questions 23 & 24

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 if input, and, | separates alternate right hand sides of productions.

$S \to a\,Ab\,B \mid b\,Aa\,B \mid \varepsilon$

$A \to S$

$B \to S$

	a	b	$
S	E1	E2	$S \to \varepsilon$
A	$A \to S$	$A \to S$	Error
B	$B \to S$	$B \to S$	E3

Q.23 The FIRST and FOLLOW sets for the non-terminals A and B are

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

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

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

(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) $= \phi$

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

FOLLOW(A) = {a, b}

FOLLOW(B) = {a, b}

[GATE 2012 : IIT Delhi]

Q.24 The appropriate entries for E1, E2 and E3 are

(A) $E1 : S \to aAbB, A \to S$

$E2 : S \to bAaB, B \to S$

$E3 : B \to S$

(B) $E1 : S \to aAbB, S \to \varepsilon$

$E2 : S \to bAaB, S \to \varepsilon$

$E3 : S \to \varepsilon$

(C) $E1 : S \to aAbB, S \to \varepsilon$

$E2 : S \to bAaB, S \to \varepsilon$

$E3 : B \to S$

(D) $E1 : A \to S, S \to \varepsilon$

$E2 : B \to S, S \to \varepsilon$

[GATE 2012 : IIT Delhi]

Q.25 Consider the following grammar :

$P \to xQRS$

$Q \to yz \mid z$

$R \to w \mid \varepsilon$

$S \to y$

What is FOLLOW(Q)?

(A) {R} (B) {w}

(C) {w, y} (D) {w, ∉}

[GATE 2017 : IIT Roorkee]

Q.26 Consider the grammar given below:

S → Aa

A → BD

B → b|ε

D → d|ε

Let a,b,d, and $ be indexed as follows:

a	b	d	$
3	2	1	0

Compute the FOLLOW set of the non-terminal B and write the index values for the symbols in the FOLLOW set in the descending order. (For example, if the FOLLOW set is {a, b, d,$}, then the answer should be 3210)

[GATE 2019 : IIT Madras]

Q.27 Consider the SLR (1) and LALR (1) parsing tables for a context free grammar. Which of the following statements is/ are true?

(A) The goto part of both tables may be different

(B) The shift entries are identical in both the tables

(C) The reduce entries in the tables may be different

(D) The error entries in the table may be different.

[GATE 1992 : IIT Delhi]

Q.28 Which of the following statements is true?

(A) SLR parser is more powerful than LALR.

(B) LALR parser is more powerful than Canonical LR parser.

(C) Canonical LR parser is more powerful than LALR parser.

(D) The parsers SLR, Canonical LR, and LALR have the same power.

[GATE 1998 : IIT Delhi]

Q.29 Which of the following statements is false?

(A) An unambiguous grammar has the same leftmost and rightmost derivation

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

(C) LALR is more powerful than SLR (Simple LR)

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

[GATE 2001 : IIT Kanpur]

Q.30 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 the above

[GATE 2003 : IIT Madras]

Q.31 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

[GATE 2003 : IIT Madras]

Q.32 Consider the grammar shown below

$S \rightarrow i\,E\,t\,S\,S' \mid a$

$S' \rightarrow e\,S \mid \varepsilon$

$E \rightarrow b$

In the predictive parse table M, of this grammar, the entries $M[S', e]$ 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 $\{S' \rightarrow \varepsilon\}$

[GATE 2003 : IIT Madras]

Q.33 Consider the grammar shown below :

$S \rightarrow CC$

$C \rightarrow cC \mid d$

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)

[GATE 2003 : IIT Madras]

Q.34 Which of the following grammar rules violate the requirement of an operator grammar? P, Q, R are non-terminals and r, s, t are terminals.

(i) $P \rightarrow QR$ (ii) $P \rightarrow QsR$

(iii) $P \rightarrow \varepsilon$ (iv) $P \rightarrow QtRr$

(A) (i) only

(B) (i) and (iii) only

(C) (ii) and (iii) only

(D) (iii) and (iv) only

[GATE 2004 : IIT Delhi]

Q.35 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 be 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) $n_1 \geq n_3 \geq n_2$

[GATE 2005 : IIT Bombay]

Q.36 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) $S \rightarrow S * \cdot E$

(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

[GATE 2006 : IIT Kharagpur]

Q.37 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* $\{R \rightarrow \varepsilon\}$

[GATE 2006 : IIT Kharagpur]

Q.38 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

[GATE 2007 : IIT Kanpur]

Q.39 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

[GATE 2007 : IIT Kanpur]

Q.40 Consider the grammar with non-terminals $N = \{S, C, S_1\}$, terminals $T = \{a, b, i, t, e\}$, with S as the start symbol, and the following set of rules

$S \rightarrow iCtS\ S_1 \mid a$

$S_1 \rightarrow e\,S \mid \varepsilon$

$C \rightarrow b$

The grammar is not LL(1) because :

(A) It is left recursive

(B) It is right recursive

(C) It is ambiguous

(D) It is not context free

[GATE 2007 : IIT Kanpur]

Q.41 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 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 sentential form where the right hand side of the production may be found.

[GATE 2008 : IISc Bangalore]

Q.42 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

[GATE 2008 : IISc Bangalore]

Q.43 The grammar $S \rightarrow a\,S\,a|b\,S|c$ 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)

[GATE 2010 : IIT Guwahati]

Q.44 Consider the following two sets of LR(1) items of an LR (1) grammar

$X \rightarrow c.X, c/d$ $\quad X \rightarrow c.X, \$$

$X \rightarrow .cX, c/d$ $\quad X \rightarrow .cX, \$$

$X \rightarrow .d, c/d$ $\quad X \rightarrow .d, \$$

Which of the following statements related to merging of the two sets in the corresponding LALR parser is/are FALSE?

1. Cannot be merged since look aheads are different.
2. Can be merged but will result in S-R conflict.
3. Can be merged but will result in R-R conflict.
4. Cannot be merged since goto on c will lead to two different sets.

(A) 1 only (B) 2 only

(C) 1 and 4 only (D) 1, 2, 3 and 4

[GATE 2013 : IIT Bombay]

Q.45 A canonical set of items is given below

$S \rightarrow L. > R$

$Q \rightarrow R.$

On input symbol <the set has

(A) A shift reduce conflict and a reduce-reduce conflict.

(B) A shift-reduce conflict but not a reduce-reduce conflict.

(C) A reduce –reduce conflict but not a shift-reduce conflict.

(D) Neither a shift-reduce nor a reduce conflict.

[GATE 2014 : IIT Kharagpur]

Q.46 Which one of the following is TRUE at any valid state in shift-reduce parsing?

(A) Viable prefixes appear only at the bottom of the stack and not inside

(B) Viable prefixes appear only at the top of the stack and not inside

(C) The stack contains only a set of viable prefixes

(D) The stack never contains viable prefixes

[GATE 2015 : IIT Kanpur]

Q.47 Match the following :

List-I

P. Lexical analysis

Q. Parsing

R. Register allocation

S. Expression evaluation

List-II

1. Graph coloring
2. DFA minimization
3. Post-order traversal
4. Production tree

Codes :

(A) $P-2, Q-3, R-1, S-4$

(B) $P-2, Q-1, R-4, S-3$

(C) $P-2, Q-4, R-1, S-3$

(D) $P-2, Q-3, R-4, S-1$

[GATE 2015 : IIT Kanpur]

Q.48 Among simple LR (SLR), canonical LR, and look – ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order

(A) SLR, LALR

(B) Canonical LR, LALR

(C) SLR, canonical LR

(D) LALR, canonical LR

[GATE 2015 : IIT Kanpur]

Q.49 Consider the following grammar G

$S \rightarrow F|H$

$F \rightarrow p|c$

$H \rightarrow d|c$

Where S, F, and H are non-terminal symbol, p, d, and c are terminal symbol. Which of the following statements (s) is/are correct?

S1 : LL(1) can parse all string that are generated using grammar G

S2 : LR(1) can parse all strings that are generated using grammar G

(A) Only S1

(B) Only S2

(C) Both S1 and S2

(D) Neither S1 nor S2

[GATE 2015 : IIT Kanpur]

Q.50 Which of the following statements about parser is/are CORRECT?

(i) Canonical LR is more powerful than SLR.

(ii) SLR is more powerful than LALR.

(iii) SLR is more powerful than Canonical LR.

(A) (i) only

(B) (ii) only

(C) (iii) only

(D) (ii) and (iii) only

[GATE 2017 : IIT Roorkee]

Q.51 Consider the augmented grammar given below:

S′ → S

S → ⟨L⟩ | id

L → L,S | S

Let I0 = CLOSURE ({[S′ → S]}). The number of items in the set GOTO (I0, <) is_____

[GATE 2017 : IIT Roorkee]

Q.52 Which one of following kinds of derivation is used by LR parsers?

(A) Leftmost

(B) Leftmost in reverse

(C) Rightmost

(D) Rightmost in reverse

[GATE 2019 : IIT Madras]

Q.53 Consider the following grammar.

S → aSB | d

B → b

The number of reduction steps taken by a bottom-up parser while accepting the string aaadbbb is _____

[GATE 2020 : IIT Delhi]

Self - Practice Questions

Q.1 The number of tokens in the following C code is:

Print f (“i= %d, j= % d, & i= %x, i, j & i);

(A) 3 (B) 28

(C) 12 (D) 23

Q.2 A non left recursive and left factored grammar in which all non- empty rules defining the same non terminal have disjoint first sets, such grammar is called is___

(A) LL (1) (B) LR (0)

(C) SLR (1) (D) None of these

Q.3 Consider the grammar

$Z \rightarrow Z \uparrow Y$

$Y \rightarrow T \hat{} Y / id$

Which of the following is false

(A) $\uparrow$ is left associative

(B) $\hat{}$ is right associative

(C) $\hat{}$ has higher precedence than $\uparrow$

(D) Both $\uparrow$ and $\hat{}$ are left associative

Q.4 The # define….. Direction in C is handled on a C compiler

(A) By the lexical analysis

(B) By the syntax analysis

(C) By the semantic analysis

(D) By the code optimizer

Q.5 The grammar $S \rightarrow T \mid \varepsilon \mid a,\ T \rightarrow S \mid a$

(A) is ambiguous and hence not SLR (1)

(B) is unambiguous & LR (0)

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

(D) is Unambiguous and hence not SLR (0)

Q.6 Which one of the following is TRUE at any valid state in shift reduce parsing?

(A) Viable prefixes appear only at the bottom of the stack and not inside

(B) Viable prefixes appear only at the top of the stack and not inside

(C) The stack contains only a set of viable prefixes.

(D) The stack never contains viable prefixes.

Q.7 Compute the FOLLOW set of S for the following CFG:

$S \rightarrow SPQR$

$P \rightarrow pPt \mid \varepsilon$

$Q \rightarrow qQ \mid \varepsilon$

$R \rightarrow Rr \mid Qm \mid \varepsilon$

(A) $\{p,q,r,m,\$\}$ (B) $\{p,q,r,t,\$\}$

(C) $\{q,r,m,\$\}$ (D) $\{p,q,r,m,t,\$\}$

Q.8 Consider the following CFG:

$E \rightarrow A$

$A \rightarrow BC \mid DBC$

$B \rightarrow Bb \mid \in$

$C \rightarrow c \mid \in$

$D \rightarrow a \mid d$

What is the FIRST and FOLLOW set of nonterminal B?

(A) FIRST $= \{b, \in\}$, FOLLOW $= \{b, c\}$

(B) FIRST $= \{b\}$, FOLLOW $= \{b, c, \$\}$

(C) FIRST $= \{b\}$, FOLLOW $= \{b, \$\}$

(D) FIRST $= \{b, \in\}$, FOLLOW $= \{b, c, \$\}$

Q.9 Consider the following grammar:

$E \rightarrow E * F \mid F$

$F \rightarrow E + G \mid G$

$G \rightarrow id \mid \in$

In the above grammar:

(A) * is left associative, + is right associative

(B) + is left associative, + is right associative

(C) Both are left associative

(D) Both are right associative

Q.10 Consider the following statements about parsers:

S1: Top- down parser cant parse left recursive grammar.

S2: Keywords of a language are recognized during parsing of the program.

Choose the suitable option about above statements.

(A) S_1 is false, S_2 is true

(B) S_1 is true, S_2 is false

(C) Both statements are true

(D) Both statement are false

Q.11 Choose the False statement.

(A) No left recursive / ambiguous grammar can be LL (1)

(B) The class of grammars that can be parsed using LR methods is proper subset of the class of grammar that can be parsed by LL method

(C) LR parsing is non- backtracking method

(D) LR parsing can describe more languages than LL parsing

Q.12 Which of the following is True?

(A) Handle of a string is a sub string that matches left hand side of production

(B) RR conflicts never occur in LALR (1)

(C) SR conflicts occur in LALR (1)

(D) None of these

Q.13 The grammar

$S \to FA$

$S \to \varepsilon \mid \phi TA$

$F \to i$

Reflects that

(A) ϕ is left associative.

(B) ϕ is right associative

(C) Cannot deduce associative from the grammar

(D) None of the above.

Q.14 In the grammar $E \to E = E \mid E \mid E \mid i$ when we construct a SLR (1) machine

(A) No inadequate states parse

(B) Inadequate states can be resolved using the associativity and precedence of $= \& +$.

(C) Inadequate states cannot be resolved

(D) None of the above

Q.15 Consider the following statements:

(A) LL (k) grammars have one to one correspondence with DCFLs.

(B) LE (k) grammars have one to one correspondence with CFLs.

(A) A is true but B is False.

(B) A is false but B is True.

(C) Both are False

(D) Both are True.

Q.16 Consider grammar with start symbol 'E"

$E \to TE'$

$E' \to +TE' \mid \in$

$T \to FT'$

$T' \to *FT' \mid \in$

$F \to (E) \mid id$

Which of the following statement is correct?

(i) '+'has more precedence than '*'

(ii) '*' has more precedence than '+'

(iii)Both have same precedence

(iv)' id' has less precedence than '('

(A) i (B) ii

(C) iii (D) iv

Q.17 Construct the LALR (1) set of items for the grammar:

$S' \to S$

$S \to +SS \mid a$

Then, identify, in the list below, one of the LALR (1) sets of items for

(A) $[S \to a., \$]$

(B) $[S \to a., +a]$

(C) $[S \to +SS., \$ + a]$

(D) $[S \to +SS., \$]$

Q.18 Consider the following CFG, with S as start symbol:

$S \to aA \mid CB$

$A \to BaA \mid \in$

$B \to bB \mid Abc \mid \in$

$C \to B$

Which of the following is correct/

First $(S) = \{a, b, \in\}$ $(S) = \{a, \in\}$

(A) First $(B) = \{b, \in\}$

Follow $(C) = \{b, a, \$\}$

First $(S) = \{a, b, \$\}$

(B) First $(B) = \{b, \in\}$

Follow $(C) = \{a, b, \in\}$

(C) First $(B) = \{b, \in\}$

Follow $(C) = \{b, a, \$\}$

(D) First $(B) = \{a, \in\}$

Follow $(C) = \{b, a, \$\}$

Q.19 Consider an SLR (1) and LALR (1) tables, which of the following is true?

(A) Shift entries are different

(B) Reduce entries are same

(C) Goto entries are different

(D) Error entries are different

Q.20 Among simple LR (SLR), canonical LR, and Look- ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?

(A) SLR, LALR
(B) Canonical LR, LALR
(C) SLR, canonical LR
(D) LALR, canonical LR

Q.21 $S \to aSAb \mid bSBc \Rightarrow \text{First}(S) = \{a, b\}$

$A \to +AB \mid \varepsilon \Rightarrow \text{First}(A) = \{+, \varepsilon\}$

$B \to *BC \mid \varepsilon \Rightarrow \text{First}(B) = \{*, \varepsilon\}$

$C \to aC + d \Rightarrow \text{First}(C) = \{a, d\}$

What is in the follow (S)?

(A) $\{a, b, c, +, \$\}$ (B) $\{a, c, +, *, \$\}$
(C) $\{b, c, +, *, \$\}$ (D) $\{a, b, d, *, \$\}$

Q.22 Consider the following two grammars.

$G_1 : A \to Al \mid 0A \mid 01$

$G_2 : A \to 0A \mid 1$

Which of the following is True regarding above grammars?

(A) G_1 is LR (k)
(B) G_2 is LR (k)
(C) Both G_1 and G_2 is LR (k)
(D) None is LR (k)

Q.23 Consider the following grammar.

$S \to aB \mid aAb$

$A \to aAb \mid a$

$B \to aB \mid \varepsilon$

How many back tracks are required to generate the string aab from the above grammar?

(A) 1 (B) 2
(C) 3 (D) 4

Q.24 $S \to aA * S$

$A \to +S \mid (S \mid \varepsilon$

Set {+,(} will be in the

(A) First (A) (B) First (E)
(C) Follow (E) (D) Follow (A)

Q.25 The grammar $S \to aSa \mid bS \mid \in$ 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)

Q.26 Consider the SDTS below

$E \to E + T \mid T$

$T \to id$

Choose the correct statement?

(A) + is left associative
(B) + is right associative
(C) The grammar is ambiguous
(D) Associativity cannot be associated with +.

Q.27 $S \to Sa \mid b$

Which of the following is True?

(A) There will be SR conflict during parsing
(B) There will be RR conflict during parsing
(C) There will be both conflict
(D) There will be no conflict

Q.28 $P \to P\alpha Q \mid Q$

$Q \to Q\beta R \mid R$

$R \to \text{num}$

If $2\alpha 3\alpha 4\beta 1\alpha 2\beta 1$ is evaluated to 18, then which of the following is the correct value for α and β

(A) +, * (B) +, –
(C) *, – (D) –,+

Q.29 Which of the following is operator grammar?

(A) $S \to AA$
$A \to a \mid \in$

(B) $S \to SAS$
$A \to a$

(C) $S \to AB$
$A \to aA \mid + \mid a$
$B \to aB \mid + \mid b$

(D) $S \to A + B$
$A \to aA \mid a$
$B \to bB \mid b$

Q.30 Consider the following statements:

S_1 : A regular grammar is always linear but not all linear grammar is regular.

S_2 : In LL grammar, the usage of production rule can be predicted exactly, by looking at a limited part of input.

Which of the above statements are true?

(A) S_1 is true and S_2 is false

(B) S_2 is true and S_1 is false

(C) Both are true

(D) Both are false

Q.31 The grammar which is equivalent to

$S \to SAa \mid Sa \mid a$

$A \to Ab \mid b$

After eliminating of left recursion is

(A) $S \to aS'$

$S' \to AaS' \mid aS'$

$A \to bA'$

$A' \to bA' \mid \in$

(B) $S \to AaS' \mid aS'$

$S' \to aS' \mid \in$

$A \to bA'$

$A' \to bA' \mid \in$

(C) $S \to aS'$

$S' \to AaS' \mid aS'$

$A \to bA'$

$A' \to bA' \mid \in$

(D) $S \to aS'$

$S' \to AaS' \mid aS'$

$A \to bA'$

$A' \to \in$

Q.32 Consider the following grammar:

$E \to TE'$

$E' \to +TE' \mid \in$

$T \to FT'$

$T' \to *FT' \mid \in$

$E \to (E) \mid id$

Which of the following is correct regarding the entry in the predictive parsing table (LL (1) parsing table) M for above grammar?

(A) $M[T,(] = T \to FT', M[T',+] = T' \to \in$

(B) $M[E,(] = E' \to \in, M[E',*] - E' \to \in$

(C) $M[E,(] = E \to TE', M[E,*] \to E \to \in$

(D) $M[T,id] = T \to FT', M[T,)] = T \to FT'$

Q.33 Consider the following augmented grammar with labels a and b

$S' \to S$

$a : S \to (S)S$

$b : S \to \varepsilon$

LR (0) sets are given as following for the above grammar.

(1)	(2)	(3)	(4)	(5)	(6)
$S' \to S$	$S' \to S$	$S \to (S.)S$	$S \to (S.)S$	$S \to (S.)S$	$S \to (S)S.$
$S \to S(S)$		$S \to .(S)S$		$S \to .(S)S$	
$S \to$		$S \to .$		$S \to .$	

If the following SLR (1) table is constructed as below then find the missing entries at E_1, E_2 and E_3 respectively

	(	)	$	S
1	S_3	r_b	r_b	2
2	.		Accept	
3	S_3	E_1	E_2	4
4		S_5		
5	S_3	r_b	r_b	E_3
6		r_a	r_a	

(A) $r_a, r_b, 5$ (B) $r_a, r_a, 6$

(C) $r_b, r_a, 5$ (D) $r_b, r_b, 6$

Q.34 Assume $x, -, +$ and / are operators. Precedence and associativity given for those operators as following:

1. $\times$ has highest precedence among all operators and it is left associative
2. $-, +$ and / are having equal precedence and they are right associative.

Using $\times$ as Multiplication, $-$ as subtraction, $+$ as Addition and / is Division.

The output of the following expression: $10 \times 10 - 5 + 15 - 5 \times 10 / 5$ is _____,

Q.35 Consider the following grammar

$S \to ABA$

$A \to Bc \mid dA \mid \in$

$B \to eA$

How many entries have multiple productions in LL (1) table?

Q.36 Find the number of tokens in the following C code using lexical analyzer of compiler

```
Main ( )
{
Int * a, b;
B = 10;
A = &b;
Print (" % d % d", b, *a);
B = */ * pointer */ b;
}
```

Q.37 Consider the grammar given below

$S \to E\#$

$E \to .T \mid E;T$

$T \to Ta \mid \varepsilon$

Number of inadequate states in DFA with LR (0) items is

Q.38 Consider the following augmented grammar.

$S \to aAb \mid eb$

$A \to e \mid f$

The number of states in LR (0) construction are_____.

Q.39 Consider the following grammar:

$S \to Aa \mid B$

$B \to a \mid bC$

$C \to a \mid \in$

The number of productions in simplified CFG is_____.

Q.40 Consider the following grammar which is not LL (1) because LL (1) table contain multiple entry for same production.

$S \to .aAbB \mid bAaB \mid \in$

$A \to S$

$B \to S$

The number of entries have multiple production in LL (1) table are_____.

Q.41 Let G to a grammar with the following productions:

$E \to E + T \mid T$

$T \to T * F \mid F$

$T \to (E) + T - F$

$F \to id$

If LR (1) parser is used to construct the DFA using the above production, then how many look a-heads are present for an item $T \to .T * F$ in the initial state_____.

Q.42 Find the number of tokens in the following C code using lexical analyzer of compiler.

```
Main ( )
{
/* int a = 10; */
Int * u, * v, s;
u = & s;
v = & s;
printf (" %d %d", s, *u);
//code ended
}
```

Q.43 Consider the following program:

```
Main ( )
{ int x = 10 ;
It (x < 20;
Else
y = 20 ;
}
```

When lexical analyzer scanning the above program, how many lexical errors can be produced?

Q.44 How many DFA states are constructed for the following augmented grammar using LR (0) parser?

$S' \to \cdot S\$$

$S' \to x \mid (A)$

$A \to A, S \mid S$

Where S is the start symbol and $S' \to S\$$ is augmented production.

Q.45 Let G be the following grammar:

$S \to aABbCD$

$A \to ASd \mid \in$

$B \to SAc \mid hC \mid \in$

$C \to Sf \mid Cg$

$C \to \cdot BD \mid \in$

How many number of productions will be in G after elimination of all null productions only?

{Consider $S \to a \mid b$ as 2 productions (i) $S \to a$, (ii) $S \to b$}.

Answer Keys

Classroom Practice Questions									
1	C	2	A	3	B	4	C	5	C
6	C	7	A	8	B	9	C	10	B
11	C	12	D	13	B	14	A	15	A
16	A	17	D	18	A	19	C	20	B
21	B	22	B	23	A	24	C	25	C
26	31	27	B,C,D	28	C	29	A	30	D
31	B	32	D	33	A	34	B	35	B
36	D	37	A	38	A	39	C	40	C
41	D	42	B	43	C	44	D	45	D
46	C	47	C	48	C	49	D	50	A
51	5	52	D	53	7				
Self - Practice Questions									
1	C	2	A	3	D	4	A	5	A
6	C	7	A	8	D	9	C	10	B
11	B	12	D	13	B	14	B	15	C
16	B	17	C	18	A	19	D	20	C
21	C	22	B	23	B	24	A	25	D
26	A	27	D	28	C	29	D	30	C
31	C	32	A	33	D	34	90	35	2
36	35	37	2	38	9	39	8	40	2
41	4	42	34	43	0	44	10	45	18

2 Semantic Analyzer

Classroom Practice Questions

Q.1 Which data structure in a compiler is used for maintaining information about variables and their attributes?

(A) Abstract syntax tree

(B) Symbol table

(C) Semantic stack

(D) Parse table

[GATE 2010 : IIT Guwahati]

Q.2 A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar.

$S \rightarrow xxW$ {print "1"}

$S \rightarrow y$ {print "2"}

$W \rightarrow Sz$ {print "3"}

What is the translation of "xxxxyzz" using the syntax directed translation scheme described by the above rules?

(A) 23131 (B)11233

(C) 11231 (D)33211

[GATE 1995 : IIT Kanpur]

Q.3 Consider the productions A → PQ and A → XY. Each of the five non-terminals A, P,Q,X, and Y has two attributes: s is a synthesized attribute, and i is an inherited attribute. Consider the following rules.

Rule 1: P.i=A.i+2, Q.i=P.i+A.i, and A.s=P.s+Q.s

Rule 2: X.i=A.i+Y.s and Y.i=X.s+A.i

Which one of the following is TRUE?

(A) Both Rule 1 and Rule 2 are L-attributed

(B) Only Rule 1 is L-attributed

(C) only Rule 2 is L-attributed

(D) Neither Rule 1 nor Rule 2 is L-attributed

[GATE 2020 : IIT Delhi]

Q.4 In the following grammar

X :: = X ⊕ Y / Y

Y :: = Z * Y / Z

Z :: = id

Which of the following is true? a. '⊕' is left associative while '*' is right associative b. Both '⊕' and '*' are left associative c. '⊕' is right associative while '*' is left associative d. None of the above

(A) a (B) b

(C) c (D) d

[GATE 1997 : IIT Madras]

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

(A) Always be evaluated.

(B) Be evaluated only if the definition is L-attributed.

(C) Be evaluated only if the definition has synthesized attributes.

(D) Never be evaluated.

[GATE 2003 : IIT Madras]

Q.6 Consider the translation scheme shown below

S → T R

R → + T {print ('+');} R | ε

T → 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) 9 5 + 2 +
(C) 9 5 2 + + (D) + + 9 5 2

[GATE 2003 : IIT Madras]

Q.7 Consider the syntax directed definition shown below.

S → id : = E {gen (id.place = E.place;);}
E → E1 + E2 {t = newtemp ( ); gen (t = El.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 ti'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) t1 = Y + Z; X = t1
(C) t1 =Y; t2 = t1 + Z; X = t2
(D) t1 = Y; t2 = Z; t3 = t1 + t2; X = t3

[GATE 2003 : IIT Madras]

Q.8 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

[GATE 2004 : IIT Delhi]

Q.9 Consider the grammar $E \rightarrow E+n|E\times n|n$ for a sentence $n+n\times n$, then handles in the right sentential form of the reduction are

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

[GATE 2005 : IIT Bombay]

Q.10 Consider the following expression grammar. The semantic rules for expression calculation are stated next to each grammar production.

E → number E.val = number. val
| E '+' E E(1).val = E(2).val + E(3).val
| E '×' E E(1).val = E(2).val × E(3).val

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

[GATE 2005 : IIT Bombay]

Q.11 Consider the following expression grammar. The semantic rules for expression calculation are stated next to each grammar production.

E → number E.val = number. val
| E '+' E E(1).val = E(2).val + E(3).val
| E '×' E E(1).val = E(2).val × E(3).val

Assume the conflicts in 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 × 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 '×' is higher than that of '+', and both operators are left associative; expression is evaluated to 7

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

[GATE 2005 : IIT Bombay]

Q.12 Consider the following translation scheme.

$S \rightarrow ER$

$R \rightarrow E\{\text{print}('*');\}R \mid \varepsilon$

$E \rightarrow F + E\{\text{print}('+');\} \mid F$

$F \rightarrow (S) \mid id\{\text{print}(id.value);\}$

Here id is a token 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+*

[GATE 2006 : IIT Kharagpur]

Q.13 One of the purpose of using intermediate code in compilers is to

(A) Make parsing and semantic analysis simpler.

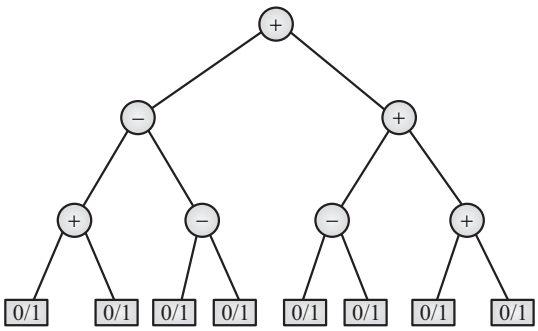
(B) Improve error recovery and error reporting.

(C) Increase the chances of reusing the machine-independent code optimizer in other compilers.

(D) Improve the register allocation.

[GATE 2014 : IIT Kharagpur]

Q.14 Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is ___.



[GATE 2014 : IIT Kharagpur]

Q.15 Consider the following Syntax Directed Translation Scheme (SDTS), with non-terminals {S,A} and terminals {a,b}.

$S \rightarrow aA$ {print 1}

$S \rightarrow a$ {print 2}

$A \rightarrow Sb$ {print 3}

Using the above SDTS, the output printed by a bottom up parser, for the input aab is :

(A) 1 3 2 (B) 2 2 3

(C) 2 3 1 (D) Syntax error

[GATE 2016 : IISc Bangalore]

Q.16 The attributes of three arithmetic operators in some programming language are given below.

Operator	Precedence	Associativity	Arity
+	High	Left	Binary
−	Medium	Right	Binary
*	Low	Left	Binary

The value of the expression 2 – 5 + 1 – 7 * 3 in this language is _____ ?

[GATE 2016 : IISc Bangalore]

Q.17 Which one of the following statements is FALSE?

(A) Context-free grammar can be used to specify both lexical and syntax rules.

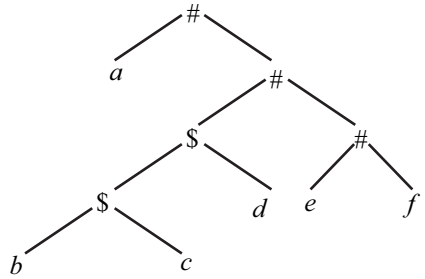
(B) Type checking is done before parsing.

(C) High-level language programs can be translated to different Intermediate Representations.

(D) Arguments to a function can be passed using the program stack.

[GATE 2018 : IIT Guwahati]

Q.18 Consider the following parse tree for the expression a # b $ c $ d # e # f, involving two binary operators $ and #.



Which one of the following is correct for the given parse tree?

(A) $ has higher precedence and is left associative; # is right associative

(B) # has higher precedence and is left associative; $ is right associative

(C) $ has higher precedence and is left associative; # is left associative

(D) # has higher precedence and is right associative; $ is left associative

[GATE 2018 : IIT Guwahati]

Q.19 Consider the following grammar and the semantic actions to support the inherited type declaration attributes. Let $X_1, X_2, X_3, X_4, X_5,$ and X_6 be the placeholders for the non-terminals D, T, L or L_1 in the following table :

Production rule	Semantic action
$D \rightarrow T\ L$	$X_1 \cdot type = X_2 \cdot type$
$T \rightarrow int$	$T \cdot type = int$
$T \rightarrow float$	$T \cdot type = float$
$L \rightarrow L_1, id$	$X_3 \cdot type = X_4 \cdot type$ addType(id.entry, $X_5 \cdot type$)
$L \rightarrow id$	addType(id.entry, $X_6 \cdot type$)

Which one of the following are the appropriate choices for X_1, X_2, X_3 and X_4 ?

(A) $X_1 = L,\ X_2 = T,\ X_3 = L_1,\ X_4 = L$

(B) $X_1 = T,\ X_2 = L,\ X_3 = L_1,\ X_4 = T$

(C) $X_1 = L,\ X_2 = L,\ X_3 = L_1,\ X_4 = T$

(D) $X_1 = T,\ X_2 = L,\ X_3 = T,\ X_4 = L_1$

[GATE 2019 : IIT Madras]

Self - Practice Questions

Q.1 Consider the following SDT.

$A \rightarrow BC\ *$

(I) $B.i = f(A, i)$

(II) $B.i = f(A, S)$

(III) $A.S = f(B.s)$

Which of the above is violating L-attribute definition?

(A) I only (B) II only

(C) I, II (D) I, II, III

Q.2 If attribute can be evaluated in depth-first order then definition is

(A) S-attributed

(B) L-attributed

(C) Both (A) and (B)

(D) None of these

Q.3 Let G be the grammar with following transition

$S \rightarrow a\ \{\text{print "0"}\}\ A$

$A \rightarrow b\ \{\text{print "1"}\}\ B$

$A \rightarrow c\ \{\text{print "2"}\}$

$A \rightarrow \varepsilon\ \{\text{print "–"}\}$

$B \rightarrow d\ \{\text{print "1"}\}\ A$

$B \rightarrow \varepsilon\ \{\text{print "0"}\}$

What is the output produced for the input "*abdbdc*" using the bottom up parsing with above translations?

(A) 211-10 (B) 211110

(C) 111-20 (D) 111012

Q.4 Let "G" be a grammar with the following translations :

$S \rightarrow p\ \{\text{print "}G\text{"}\}\ P$

$P \rightarrow q\ \{\text{print "}A\text{"}\}\ Q$

$P \rightarrow r\ \{\text{print "}T\text{"}\}$

$P \rightarrow \varepsilon\ \{\text{print "}E\text{"}\}$

$Q \rightarrow S\ \{\text{print "}A\text{"}\}\ P$

$Q \rightarrow \varepsilon\ \{\text{print "}O\text{"}\}$

What is the output produced for the input "*pqsqsr*" using the bottom up parsing with above translations?

(A) *TAAAG* (B) *TAAAAG*

(C) *GAAA* (D) *AAAGAT*

Q.5 Consider the syntax directed translation scheme below for the grammar :

$N \rightarrow L$

$L \rightarrow L + B \mid B$

$B \rightarrow 0 \mid 1$

The Schema is

$N \rightarrow L$

$\{N.\text{val} = L.\text{val}\}$

$N \rightarrow L + B$

$\{L.\text{val} = L.\text{val} * 2 + B.\text{val}\}$

$L \rightarrow B$

$\{L.\text{val} = B.\text{val}\}$

$B \to 0$ $\{B.\text{val} = 0\}$

$B \to 1$ $\{B.\text{val} = 1\}$

The value of N.val is

(A) The number of bits in the input string

(B) The value of the non decimal input in binary form.

(C) The value of the binary input in decimal form

(D) None of the above.

Q.6 $X \to YZ$

$Y \to Y + Z \{\text{print}('+');\}$

$T \{Y.\text{val} = T.\text{val}\}$

$Z \to *Y \{\text{print}('*');\} Z$

$T \{Z.\text{val} = T.\text{val}\}$ ε

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

For $2+3*2$, the above translation scheme prints

(A) $2+3*2$ (B) $23+2*$

(C) $232*2+$ (D) $23*2+$

Q.7 A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar.

$S \to xx$ W {print "1"}

$S \to y$ W {print "2"}

$W \to Sz$ {print "3"}

What is the translation of *xxxxyzz* using the syntax-directed translation scheme describe by the above rules?

(A) 23131 (B) 11233

(C) 11231 (D) 33211

Q.8 Match the following errors corresponding to their phase :

Group A

1. Unbalanced parenthesis
2. Appearance of illegal characters
3. Unbalanced parenthesis

Group B

A. Syntactic error

B. Semantic error

C. Lexical error

(A) 1 → A, 2 → C, 3 → B

(B) 1 → B, 2 → C, 3 → A

(C) 1 → A, 2 → B, 3 → C

(D) 1 → B, 2 → C, 3 → A

Q.9 In a bottom-up evaluation of a syntax directed definition, inherited attribute can

(A) Always be evaluated

(B) Be evaluated only if definition in L-attribute

(C) Never be evaluated

(D) Be evaluated only if the definition has synthesized attributes

Q.10 Type checking is normally done using

(A) Lexical analysis

(B) Syntax analysis

(C) Syntax directed translation

(D) Code optimization

Q.11 Consider the translation scheme given below :

$S \to T - R \{\text{print}('-')\} \mid R$

$R \to +T \{\text{print}('+') \mid R \mid F$

$F \to \text{id} \{\text{print}(\text{id.value})\} \mid \in$

For an input scheme $10-5+4$, this scheme will print

(A) $10\ 5+4-$ (B) $10\ 54-+$

(C) $105-4+$ (D) $1054+-$

Q.12 Consider following SDT

$S \to A$ sign // $S.\text{val} = A.\text{val};$

Sign $\to +$ // sign.sign $= 1$

Sign $\to -$ // sign.sign $= 0$

$A \to n$ // $A.\text{val} = \text{value}(n)$

$A \to A1, n$ // $A1.\text{sign} = A.\text{sign};$

if $(A.\text{sign} = 1)$ then

$A.\text{val} = \min(A1.\text{val}, \text{value}(n));$

else

$A.\text{val} = \max(A1.\text{val}, \text{value}(n));$

A partial modified grammar and actions exclusively using synthesized attributes is given below :

$S \to B+$ // $S.\text{val} = B.\text{val}$; print $(B.\text{val})$

$S \to C-$ // $S.\text{val} = C.\text{val}$; print $(C.\text{val})$

$B \to n$ // $B.\text{val} = \text{value}(n)$

$B \to B1.n$ // $B.\text{val} = B1$

$C \to n$ // $C.\text{val} = \text{value}(n)$

$C \to C1.n$ // $C.\text{val} = B2$

What is the value of $B1$ and $B2$ to complete the grammar respectively?

(A) Max ($B1$.val, value (n)), Min ($C1$.val, value (n))

(B) Min ($B1$.val, value (n)), Max ($C1$.val, value (n))

(C) Max ($B1$.val, value (n)), Max ($C1$.val, value (n))

(D) Min ($B1$.val, value (n)), Min ($C1$.val, value (n))

Q.13 Consider the following SDT :

$E \to E + E \quad \{E.\text{val} = E_1.\text{val} + E_2.\text{val}\}$

$E \to E \times E \quad \{E.\text{val} = E_1.\text{val} - E_2.\text{val}\}$

$E \to (E) \quad \{E.\text{val} = E_1.\text{val}\}$

$E \to \text{id} \quad \{E.\text{val} = \text{id.lex}\}$

The value of the attribute computed at root when the expression $[(3\times3)+(3\times5)-6]+7$ is evaluated using the above SDT are _____.

Q.14 A shift reduce parser carries out the action specified within braces immediately after reducing with the corresponding rule of grammar :

$S \to xxW \quad$ {print "1"}

$S \to y \quad$ {print "2"}

$W \to Sz \quad$ {print "3"}

What is the translation of *xxxxyzz* using the syntax directed translation scheme described by the above rules?

Q.15 Consider the following SDT

$S \to S * S_1 \mid S_2 \quad (S.\text{val} = S_1.\text{val} + S.\text{val})$

$S \to S_2 \quad (S_1.\text{val} = S_2.\text{val})$

$S_2 \to S_2 \# S_3 \quad (S_2.\text{val} = S_2.\text{val} - S_3.\text{val})$

$S_2 \to S_3 \quad (S_2.\text{val} = S_3.\text{val})$

$S_3 \to \text{id} \quad (S_3.\text{val} = \text{id})$

Evaluate the expression

15 # 12 * 5 # 25 # 30 * 60

Answer Keys

Classroom Practice Questions									
1.	B	2.	A	3.	B	4	A	5	B
6	B	7	B	8	C	9	D	10	C
11	B	12	D	13	C	14	6	15	C
16	9	17	B	18	A	19	A		
Self - Practice Questions									
1.	B	2.	B	3.	B	4.	B	5.	C
6.	B	7.	A	8.	A	9.	C	10.	B
11.	C	12.	B	13.	-5	14.	23121	15.	12

3 Intermediate Code Generation & Optimization

Classroom Practice Questions

Q.1 The pass number for each of the following activities

(i) Object code generation

(ii) Literals added to literal table

(iii) Listing printed

(iv) Address resolution of local symbols that occur in a two assembler respectively are

(A) 1, 2, 1, 2 (B) 2, 1, 2, 1

(C) 2, 1, 1, 2 (D) 1, 2, 2, 2

[GATE 1996 : IISc Bangalore]

Q.2 The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called

(A) Assembly

(B) Parsing

(C) Relocation

(D) Symbol resolution

[GATE 2001 : IIT Kanpur]

Q.3 Match the following :

Group – I

(A) Pointer data type

(B) Activation record

(C) Repeat-until

(D) Coercion

Group – II

(P) Type conversion

(Q) Dynamic data structure

(R) Recursion

(S) Non-deterministic loop

(A) a - p, b - r, c - s, d - q

(B) a - q, b - r, c - s, d - p

(C) a - q, b - s, c - r, d - p

(D) a - r, b - q, c - s, d – p

[GATE 1990 : IISc Bangalore]

Q.4 Generation of intermediate code based on an abstract machine model is useful in compilers because

(A) It makes implementation of lexical analysis and syntax analysis easier.

(B) Syntax directed translations can be written for intermediate code generation.

(C) It enhances the portability of the front end of the compiler.

(D) It is not possible to generate code for real machines directly from high level language programs.

[GATE 1994 : IIT Kharagpur]

Q.5 A linker is given object modules for a set of programs that were compiled separately. What information need not be included in an object module?

(A) Object code.

(B) Relocation bits.

(C) Names and locations of all external symbol defined in the object module.

(D) Absolute addresses of internal symbols.

[GATE 1995 : IIT Kanpur]

Q.6 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 necessary be interleaved

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

[GATE 2004 : IIT Delhi]

Q.7 Consider the following C code segment.

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
                if (i%2)
    {
     X += (4 * j + 5 * i);
                    Y += (7 + 4 * j);
                }
    }
}
```

Which one of the following is false?

(A) The code contains loop-invariant computation

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

(C) There is scope of strength reduction in this code.

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

[GATE 2006 : IIT Kharagpur]

Q.8 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 and stores the result in R_i. Val denotes the content of memory location m.

MOV m, R_i – Moves the content of memory location m to register R_i.

MOV R_i, m – Moves the content of registers, and to memory location m

The computer has only two registers, and OP is either ADD or SUB. Consider the following basic block:

t1 = a + b

t2 = c + d

t3 = e – t2

t4 = t1 – t3

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 instruction in the code generated for this basic block?

(A) 2 (B) 3

(C) 5 (D) 6

[GATE 2007 : IIT Kanpur]

Q.9 Some code optimization are carried out on the intermediate code because

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

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

(C) The information from dataflow analysis cannot otherwise be used for optimization.

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

[GATE 2008 : IISc Bangalore]

Q.10 Which languages necessarily need heap allocation in the runtime environment?

(A) Those that support recursion.

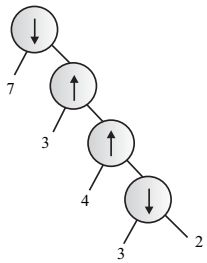
(B) Those that use dynamic scoping

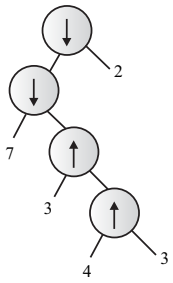
(C) Those that allow dynamic data structures

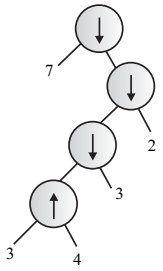
(D) Those that use global variables.

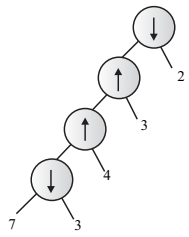
[GATE 2010 : IIT Guwahati]

Q.11 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) 

[GATE 2011 : IIT Madras]

Q.12 Which one of the following is FALSE?

(A) A basic block is a sequence of instructions where control enters the sequence at the beginning and exists at the end.

(B) Available expression analysis can be used for common sub-expression elimination.

(C) Live variable analysis can be used for dead code elimination.

(D) $x = 4*5 \Rightarrow x = 20$ is an example of common sub-expression elimination.

[GATE 2014 : IIT Kharagpur]

Q.13 Which one of the following is NOT performed during compilation?

(A) Dynamic memory allocation

(B) Type checking

(C) Symbol table management

(D) Inline expansion

[GATE 2014 : IIT Kharagpur]

Q.14 For a C program accessing $X[i][j][k]$, the following intermediate code is generated by a complier. Assume that the size of an integer is 32 bits and the size of character is bits.

$t0 = i*1024$

$t1 = j*32$

$t2 = k*4$

$t3 = t1 + t0$

$t4 = t3 + t2$

$t5 = X[t4]$

Which one of the following statements about the source code for the C program is CORRECT?

(A) X is declared as "int *X* [32][32][8]"

(B) X is declared as "int *X* [4][1024][32]"

(C) X is declared as "*char X* [4][32][8]"

(D) X is declared as "*char X* [32][16][2]"

[GATE 2014 : IIT Kharagpur]

Q.15 Which of the following statements are CORRECT?

1. Static allocation of all data areas by a compiler makes it impossible to implement recursion.
2. Automatic garbage collection is essential to implement recursion.
3. Dynamic allocation of activation records is essential to implement recursion.
4. Both heap and stack are essential to implement recursion.

(A) 1 and 2 only (B) 2 and 3 only

(C) 3 and 4 only (D) 1 and 3 only

[GATE 2014 : IIT Kharagpur]

Q.16 Consider the basic block given below.

$a = b + c$

$c = a + d$

$d = b + c$

$e = d - b$

$a = e + b$

The minimum number of nodes and edges present in the DAG representation of the above basic block respectively are

(A) 6 and 6 (B) 8 and 10

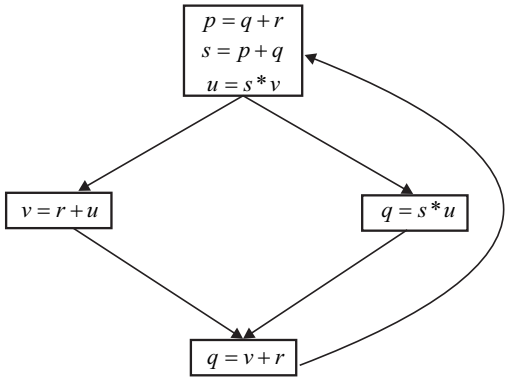
(C) 9 and 12 (D) 4 and 4

[GATE 2014 : IIT Kharagpur]

Q.17 A variable x is said to be live at a statement S_i in a program if the following three conditions hold simultaneously:

1. There exists a statement S_j that uses x
2. There is a path from S_i to S_j in the flow graph corresponding to the program

3. The path has no intervening assignment to x including at S_i and S_j



The variables which are live both at the statement in basic block 2 and at the statement in basic block 3 of the above control flow graph are

(A) p, s, u (B) r, s, u

(C) r, u (D) q, v

[GATE 2015 : IIT Kanpur]

Q.18 The least number of temporary variables required to create a three-address code in static single assignment form for the expression q + r/3 + s – t * 5 + u * v/w is _____.

[GATE 2015 : IIT Kanpur]

Q.19 In the context of abstract-syntax-tree (AST) and control-flow-graph (CFG), which one of the following is TRUE?

(A) In both AST and CFG, let node N_2 be the successor of node N_1. In the input program, the code corresponding to N_2 is present after the code corresponding to N_1

(B) For any input program, neither AST nor CFG will contain a cycle

(C) The maximum number of successors of a node in an AST and a CFG depends on the input program

(D) Each node in AST and CFG corresponds to at most one statement in the input program

[GATE 2015 : IIT Kanpur]

Q.20 Consider the intermediate code given below.

(1) $i = 1$

(2) $j = 1$

(3) $t_1 = 5 * i$

(4) $t_2 = t_1 + j$

(5) $t_3 = 4 * t_2$

(6) $t_4 = t_3$

(7) $a[t_4] = -1$

(8) $j = j + 1$

(9) if $j <= 5$ $goto(3)$

(10) $i = i + 1$

(11) If $i < 5$ $goto(2)$

The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are

(A) 5 and 7 (B) 6 and 7

(C) 5 and 5 (D) 7 and 8

[GATE 2015 : IIT Kanpur]

Q.21 A student wrote two context-free grammars G1 and G2 for generating a single C-Like array declaration. The dimension of the array is at least one. For example,

int a[10][3];

The grammars use D as the start symbol, and use six terminal symbols int; id[ ] num.

Grammar G1	**Grammar G2**
$D \rightarrow \text{int } L;$	$D \rightarrow \text{int } L;$
$L \rightarrow id[E$	$L \rightarrow id\ E$
$E \rightarrow num]$	$E \rightarrow E[num]$
$E \rightarrow num][E$	$E \rightarrow [num]$

Which of the grammar correctly generate the declaration mentioned above?

(A) Both G1 and G2

(B) Only G1

(C) Only G2

(D) Neither G1 nor G2

[GATE 2016 : IISc Bangalore]

Q.22 Consider the following intermediate program in three address code

$p = a - b$

$q = p * c$

$p = u * v$

$q = p + q$

Which one of the following corresponds to a static single assignment form of the above code?

(A) $p_1 = a - b$
$q_1 = p_1 * c$
$p_1 = u * v$
$q_1 = p_1 + q_1$

(B) $p_3 = a - b$
$q_4 = p_3 * c$
$p_4 = u * v$
$q_5 = p_4 + q_4$

(C) $p_1 = a - b$
$q_1 = p_2 * c$
$p_3 = u * v$
$q_2 = p_4 + q_3$

(D) $p_1 = a - b$
$q_1 = p * c$
$p_2 = u * v$
$q_2 = p + q$

[GATE 2017 : IIT Roorkee]

Q.23 Consider the following grammar :

$stmt \rightarrow if$ expr then expr else expr; stmt| 0

expr $\rightarrow$ term relop term |term

term $\rightarrow$ id |number

id $\rightarrow a|b|c$

Number $\rightarrow$ [0-9]

Where relop is relational operator (e.g., <,>,..), 0 refers to the empty statement, and if, then, else are terminals. Consider a program P following the above grammar containing then if terminals. The number of control flow paths in P is _____.

For example the program

If e_1 then e_2 else e_3 has 2 control flow paths, $e_1 \rightarrow e_2$ and $e_1 \rightarrow e_3$

[GATE 2017 : IIT Roorkee]

Self - Practice Questions

Q.1 Consider the following statements regarding run-time environment

Which of the above statement is true?

(A) The storage used for heap section can grow at run time but not stack section.

(B) Only control links and access links are saved or restored when a function call or return happen at runtime.

(C) Control links are used in the activation record to access the non-local data.

(D) Temporary variables are one of the contents of an activation record.

Q.2 Which of the following is the postfix form of the following expression?

$-b + c * d / e$

(A) $-b\ cde*/+$ (B) $b - cd*e/+$

(C) $b\ cd*e/+-$ (D) $b\ cde*/+-$

Q.3 Number of internal nodes in the *DAG* representation of the following expression are :

$\big((a*a)*(a*a)*\big((a*a)*(a*a)\big)\big)$

(A) 2 (B) 3

(C) 4 (D) 5

Q.4 Post fix conversion for the given expression is

$(x+y)*(p-q)/(m+n)$

(A) $xy + pq - *mn + /$

(B) $xypq + - *mn + /$

(C) $xy + pq* - mn + /$

(D) $xy + - *mn/+$

Q.5 Arrays with dynamic bounds are unpopular in HLLs as

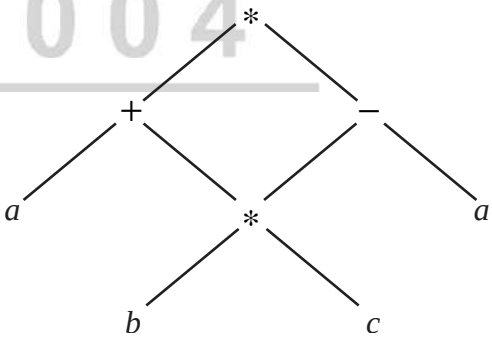
(A) The checking of bounds is prohibitively expensive.

(B) Dynamic storage allocation is more expensive than static storage allocation.

(C) More efficient code can be generated.

(D) All of the above

Q.6 The equivalent expression for the DAG is



(A) $\big((a+b)*c\big)*\big(b*(c-a)\big)$

(B) $a + (b*c - a)$

(C) $(a + (b*c))*((b*c) - a)$

(D) $a*(a + b*c) - a$

Q.7 Consider the C program given below :

```
main ( )
{
    a = a + b;
    c = a × c;
    d = c − d;
    a = c/d;
    printf ("%d ", a);
}
```

What will be the minimum number of nodes and edges present in the DAG representation of the output of above C program?

(A) 6 and 6 (B) 7 and 6

(C) 6 and 7 (D) 8 and 8

Q.8 What will be the output of the following program for call by name and copy restore parameter passing mechanism respectively?

```
void func (int a, int b)
{
    a = a + b;
    b = a + b;
    a = b − a;
    b = a − b
}
main ( )
{
    int j, i = 4;
    int array [10] = {1,2,3,4,5,6,7,8,9,0}
    func (i, array [i]);
    for (j = 0; j < 10; j++)
    printf ("%d", array [j]);
}
```

(A) call by name : −1 2 3 4 5 6 7 8 9 9
copy restore : 1 2 3 4 −9 6 7 8 9 0

(B) call by name : −1 2 3 4 5 6 7 8 9 9
copy restore : 1 2 3 4 5 6 7 8 9 0

(C) call by name : −1 2 3 4 5 6 7 8 9 9
copy restore : 1 2 3 4 −8 6 7 8 9 0

(D) call by name : −1 2 3 4 5 6 7 8 9 9
copy restore : 1 2 3 4 −8 6 7 8 9 0

Q.9 Which of the following parameter is not included in the activation record of recursive function call?

(A) Local variables (B) Return value

(C) Global variable (D) Access links

Q.10 Consider the following three address code table :

Location	Operator	Operand 1	Operand 2
(1)	*	a	b
(2)	Uminus	(1)	
(3)	+	c	d
(4)	+	(2)	(3)
(5)	+	a	b
(6)	+	(5)	(3)
(7)	–	(4)	(6)

Which of the following expression represents the above three address code (triple representation)?

(A) $-(a*b)-(c+d)+(a+b+c+d)$

(B) $+(a*b)-(c+d)-(a+b+c+d)$

(C) $-(a*b)+(c+d)-(a+b+c+d)$

(D) $-(a*b)+(c+d)+(a+b-(c+d))$

Q.11 Consider the following statements :

S_1 : Static allocation can not support recursive function.

S_2 : Stack allocation can support pointers but cannot deallocate storage at run-time.

S_3 : Heap allocation can support pointers and it can allocate or deallocate storage at run-time.

Which of the above statements are true?

(A) S_1 and S_2 (B) S_2 and S_3

(C) S_3 and S_1 (D) S_1, S_2 and S_3

Q.12 What is the minimum number of extra registers required to swap two numbers where two numbers are stored in two different registers?

Q.13 A directed acyclic graph represents one form of intermediate representation. The number of non-terminal nodes in DAG of $a = (b+c)*(b+c)$ expression is

Q.14 Consider the following expression :

The number

$$((x+y)-((x+y)*(x+y)))$$
$$+((x+y)*(x+y))$$
$$+((x+y)\div(x+y))$$

r of nodes to represent the DAG for the above expression is _____

Q.15 Consider the following expression

$$x = a*b-c*d+e$$

For generating target code how many register will be required apart from accumulator A?

Q.16 Consider two binary operators * and ^ with precedence of operator * being lower than of ^. Operator * is left associative while operator ^ is right associative. The value of 3*2^3*4 is _____

Q.17 Given the 3-address code for a basic block

Num	Instruction	Meaning
1.	$Ld\ a, T_1$	$T_1 \leftarrow a$
2.	$Ld\ b, T_2$	$T_2 \leftarrow b$
3.	$L\,dc, T_3$	$T_3 \leftarrow c$
4.	$Ld\ d, T_4$	$T_4 \leftarrow d$
5.	Add T_1, T_2, T_5	$T_5 \leftarrow T_1 + T_2$
6.	Add T_5, T_3, T_6	$T_6 \leftarrow T_3 + T_5$
7.	Add T_6, T_4, T_7	$T_7 \leftarrow T_6 + T_4$
8.	$ST\ T_7, a$	$a \leftarrow T_7$

How many registers are needed to allocate this basic block with no spills?

Q.18 Find the minimum number of temporary variables created in a 3-address code for the following expression

$$a+b*c+d-e-a+b*c$$

Consider following grammar for precedence and associativity.

Q.19 Given the 3-address code for a basic block:

Number	Instruction	Meaning
1	$Ld\ a, T_1$	$T_1 \leftarrow a$
2	$Ld\ b, T_2$	$T_2 \leftarrow b$
3	$Ld\ c, T_3$	$T_3 \leftarrow c$
4	$Ld\ d, T_4$	$T_4 \leftarrow d$
5	Add T_1, T_2, T_5	$T_5 \leftarrow T_1 + T_2$
6	Add T_1, T_2, T_5	$T_5 \leftarrow T_1 + T_2$
7	Add T_1, T_2, T_5	$T_5 \leftarrow T_1 + T_2$
8	$ST\ T_7, a$	$a \leftarrow T_7$

How many registers are needed to allocate this basic block with no spills?

$$S \rightarrow ES$$
$$E \rightarrow E-F \mid F$$
$$F \rightarrow F+G \mid T$$
$$G \rightarrow G*H \mid H$$
$$H \rightarrow id \mid \varepsilon$$

Q.20 Consider the intermediate code given below :

1. $a = 10$
2. $b = 15$
3. $a = a + b$
4. $b = a - b$
5. $a = a - b$
6. if $(a == b)$ go to (3)

The number of nodes and edges in the control-flow graph constructed for the above code, respectively are X and Y. The value of $X + Y$ is _____.

Answer Keys

Classroom Practice Questions									
1.	B	2.	C	3	A	4	C	5	C
6	B	7	D	8	B	9	A,B	10	C
11	B	12	D	13	A	14	A	15	D
16	A	17	C	18	8	19	C	20	B
21	A	22	B	23	1024				
Self - Practice Questions									
1.	D	2.	B	3.	B	4.	A	5.	D
6.	C	7.	D	8.	A	9.	C	10.	C
11.	C	12.	0	13.	3	14.	8	15.	1
16.	12338	17.	4	18.	2	19.	4	20.	10