

Enrollment No./Seat No.:

GUJARAT TECHNOLOGICAL UNIVERSITY

Bachelor of Engineering - SEMESTER - VII EXAMINATION - WINTER 2025

Subject Code: 3174801

Date: 13-11-2025

Subject Name: Compiler Design

Time: 10:30 AM TO 01:00 PM

Total Marks: 70

Instructions

1. Attempt all questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.
4. Simple and non-programmable scientific calculators are allowed.

	Marks
Q.1 (a) Give the difference among compiler, interpreter and assembler.	03
(b) Discuss cross compiler with example.	04
(c) Explain phases of compiler in detail.	07
Q.2 (a) Define token, pattern and lexemes with example.	03
(b) Write a regular expression to recognize IP address of a computer. An IP address template is of the form N.N.N.N, where N can be 0, 1, 2, 10, 25, 249, 255, etc. but cannot be 01, 005, 259, etc. i.e. syntactically valid IP addresses are 192.108.16.32, 140.182.0.1, 192.168.17.198, etc.	04
(c) Construct a DFA without constructing NFA for following regular expression. Find minimized DFA. $a^*b^*a(a \mid b)^*b^*a\#$	07
OR	
(c) Construct an NFA for following regular expression using Thompson's notation and then convert it into DFA. $aa^*(b \mid c)a^*c\#$	07
Q.3 (a) Show syntax directed definition for simple desk calculator.	03
(b) Explain role of parser with suitable diagram.	04
(c) Construct LL(1) parsing table for the following Grammar and parse the string $a^+(a^*a)$: $E \rightarrow E+T \mid T$ $T \rightarrow T * F \mid F$ $F \rightarrow (E) \mid a$	07
OR	
(a) Show annotated parse tree for $6*7+3n$, where n indicates newline.	03
(b) Explain why every S-attributed definition is L-attributed.	04

- (c) Construct CLR parsing table for the following grammar : 07
S → BB
B → xB | y

- Q.4** (a) Compare between ad-hoc methods and systematic approaches for error handling. 03
(b) Translate the expression $-(a+b)*(c+d)+(a+b+c)$ into Quadruples and Triples. 04
(c) Discuss various code optimization techniques. 07

OR

- (a) Explain error recovery strategies in brief. 03
(b) Explain any two types of intermediate code representations for $a = b*-c + b*-c$. 04
(c) Discuss critical design issues in code generation. 07

- Q.5** (a) Define Instruction-Level Parallelism (ILP). Explain its importance in modern processor architectures. 03
(b) Explain code-scheduling constraints. 04
(c) Discuss activation record management in recursive and non-recursive function calls. 07

OR

- (a) Differentiate between statically scheduled and dynamically scheduled instructions with examples. 03
(b) Explain Basic-Block Scheduling method. 04
(c) Discuss efficient heap memory management techniques and the challenges of dynamic memory allocation. 07
