## III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019 COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70 Note: 1. Question Paper consists of two parts (Part-A and Part-B) 2. Answer **ALL** the question in **Part-A** 3. Answer any FOUR Questions from Part-B (14 Marks) PART -A 1. Differentiate between token, lexeme and pattern with examples. [3M] a) b) Name the three techniques for constructing LR parsing table. [2M] What are the actions performed by Shift reduce parser? [2M] c) Define abstract Syntax tree. d) [2M] Write about the sub-division of run-time memory. [3M] e) List the characteristics of peephole optimization. f) [2M] PART -B **(56 Marks)** 2. Explain various data structures used for lexical analysis. [7M] a) Discuss in brief about Bootstrapping process with suitable diagram. [7M] b) Define Context Free Grammar. Explain how it is suitable for parsing? 3. a) [7M] Write an algorithm to find LR(0) items and give an example. b) [7M] 4. Design LALR(1) parser for the following grammar: [7M] a)  $S \rightarrow aAd \mid bBd \mid aBc \mid bAc$  $A \rightarrow e$  $B \rightarrow e$  where a, b, c, d, e are terminals. Give the SDT scheme for desk calculator. b) [7M] Explain the construction of syntax tree for expressions. 5. [7M] a) Discuss the concept of back patching with an example. b) [7M] What is a flow graph? Explain how flow graph can be constructed for a given 6. [7M] a) program. b) Discuss the advantages and disadvantages of heap storage allocation strategy. [7M] 7. Distinguish between machine dependent and machine independent optimization. a) [7M] Explain the algebraic transformations of local machine independent optimization. b) [7M]

## III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019 COMPILER DESIGN

(Computer Science and Engineering)

Note: 1. Question Paper consists of two parts ( <b>Part-A</b> and <b>Part-B</b> ) 2. Answer <b>ALL</b> the question in <b>Part-A</b>	
3. Answer any <b>FOUR</b> Questions from <b>Part-B</b>	
$\underline{\mathbf{PART}} - \underline{\mathbf{A}} \tag{14}$	Marks)
Give the types of a language processing system.	[2M]
What are the problems in top-down parsers?	[2M]
Write the applications of SDTs.	[3M]
	[3M]
<u> </u>	[2M]
Write the criteria for achieving machine independent code optimization.	[2M]
$\underline{PART - B} \tag{56}$	Marks)
Define compiler. Describe the phases of a compiler with a neat sketch.	[7M]
Explain the recognition of keywords and identifiers with a suitable transition diagram.	[7M]
Eliminate ambiguities in the following grammar: S -> iEtS  iEtSeS  a	[7M]
Define LR(k) parser. Draw and explain the model of LR parser.	[7M]
How to detect and reduce handle in LR parsers? Explain	[7M]
Construct LALR parsing table for the following grammar:  S -> CC  C -> cC   d	[7M]
What is the role of type checking in error detection and error recovery? Explain.	[7M]
Give the translation scheme to convert an expression grammar into three address code.	[7M]
Explain the activities of caller and callee in stack allocation strategy with an	[7M]
	[7M]
Trow to decess non rocal data. Explain with example.	[/1/1]
Discuss about the principal sources of optimization with examples.	[7M]
	PART -A (14  Give the types of a language processing system. What are the problems in top-down parsers? Write the applications of SDTs. What is Static Checking? List out some examples of static checks. Mention the issues in design of code generation. Write the criteria for achieving machine independent code optimization.  PART -B (56  Define compiler. Describe the phases of a compiler with a neat sketch. Explain the recognition of keywords and identifiers with a suitable transition diagram.  Eliminate ambiguities in the following grammar: S -> iEtS  iEtSeS  a  E -> b  c  d  Define LR(k) parser. Draw and explain the model of LR parser.  How to detect and reduce handle in LR parsers? Explain. Construct LALR parsing table for the following grammar: S -> CC C -> cC   d  What is the role of type checking in error detection and error recovery? Explain. Give the translation scheme to convert an expression grammar into three address code.  Explain the activities of caller and callee in stack allocation strategy with an example. What is the role of parameter passing in it? How to access non-local data? Explain with example.

Time: 3 hours

Max. Marks: 70

## III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019 COMPILER DESIGN

(Computer Science and Engineering)

PART —A (14 Marks)  1. a) What are the functions of preprocessor? [2M] b) Differentiate between top-down parser and bottom-up parser. [3M] c) Give the usage of look ahead symbol in LALR parsing. [2M] d) How to generate polish notation using translation schemes? [2M] e) What are the advantages and disadvantages of heap storage allocation strategies for records? f) Write the factors that affects the target code generation. [2M]  PART —B (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system. [7M] b) Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M] S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals. b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M] b) Discuss the transformations that are characteristic of peephole optimizations. [7M]	-	Note: 1. Question Paper consists of two parts (Part-A and Part-B)  2. Answer ALL the question in Part-A  3. Answer any FOUR Questions from Part-B		
b) Differentiate between top-down parser and bottom-up parser.  c) Give the usage of look ahead symbol in LALR parsing.  d) How to generate polish notation using translation schemes?  e) What are the advantages and disadvantages of heap storage allocation strategies for records?  f) Write the factors that affects the target code generation.  PART —B  (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system.  Explain how input buffering helps lexical analyzer in compilation process.  [7M]  3. a) Design a non-recursive predictive parser for the following grammar:  S -> AaAb   BbBb  A -> e  B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis?  [7M]  Explain.  4. a) State and explain the rules used to construct the LR(1) items.  b) Discuss the evolution order of SDTs. Also write its applications.  [7M]  5. a) What is a three address code? What are its types? How it is implemented?  b) What are the one-pass code generation methods? Explain any one.  [7M]  6. a) Generate the flow-graphs for the following expressions:  S -> id: = E   S; S   if E then S else S   do S while E.  E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.			$\underline{\mathbf{PART}} - \underline{\mathbf{A}} \tag{1}$	14 Marks)
c) Give the usage of look ahead symbol in LALR parsing. d) How to generate polish notation using translation schemes? [2M] e) What are the advantages and disadvantages of heap storage allocation strategies for records? f) Write the factors that affects the target code generation. [2M]  PART—B (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system. [7M] b) Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M] S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals. b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] S -> id: = E   S; S   if E then S else S   do S while E. E -> id + id   id b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting. [7M]	1.	a)	What are the functions of preprocessor?	[2M]
d) How to generate polish notation using translation schemes?  e) What are the advantages and disadvantages of heap storage allocation strategies for records?  f) Write the factors that affects the target code generation.  PART —B (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system. [7M]  b) Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M]  S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis? [7M]  Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M]  b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M]  b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M]  6. a) Generate the flow-graphs for the following expressions: [7M]  b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting. [7M]		b)	* * *	[3M]
e) What are the advantages and disadvantages of heap storage allocation strategies [3M] for records?  f) Write the factors that affects the target code generation. [2M]  PART -B (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system. [7M] Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M] S -> AaAb   BbBb A -> e  B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M]  b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M]  b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M]  S -> id: = E   S;S   if E then S else S   do S while E.  E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage [7M] collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]				
for records?  Write the factors that affects the target code generation.  PART —B  (56 Marks)  2. a) Describe the functionality of compilers in a typical language processing system. [7M]  Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M]  S -> AaAb   BbBb  A -> e  B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis? [7M]  Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M]  b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M]  b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M]  S -> id: = E   S;S   if E then S else S   do S while E.  E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting. [7M]				
PART -B  2. a) Describe the functionality of compilers in a typical language processing system. [7M]  Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M]  S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis? [7M]  Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M]  b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M]  b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M]  S -> id: = E   S; S   if E then S else S   do S while E. E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]		e)		ies [3M]
<ol> <li>a) Describe the functionality of compilers in a typical language processing system. [7M]         Explain how input buffering helps lexical analyzer in compilation process. [7M]     </li> <li>a) Design a non-recursive predictive parser for the following grammar: [7M]         S -&gt; AaAb   BbBb         A -&gt; e         B -&gt; e         where a, b, e are terminals.         </li> <li>b) What kinds of source program errors would be detected during lexical analysis? [7M]         Explain.     </li> <li>4. a) State and explain the rules used to construct the LR(1) items. [7M]         b) Discuss the evolution order of SDTs. Also write its applications. [7M]     </li> <li>5. a) What is a three address code? What are its types? How it is implemented? [7M]</li> <li>b) What are the one-pass code generation methods? Explain any one. [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: S -&gt; id: = E   S;S   if E then S else S   do S while E. E -&gt; id + id   id         b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.     </li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ol>		f)	Write the factors that affects the target code generation.	[2M]
b) Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M] S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals. b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] S -> id: = E   S;S   if E then S else S   do S while E. E -> id + id   id b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting. [7M]			$\underline{PART - B} $	56 Marks)
b) Explain how input buffering helps lexical analyzer in compilation process. [7M]  3. a) Design a non-recursive predictive parser for the following grammar: [7M] S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals. b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] S -> id: = E   S;S   if E then S else S   do S while E. E -> id + id   id b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting. [7M]	2.	a)	Describe the functionality of compilers in a typical language processing system	. [7M]
S -> AaAb   BbBb A -> e B -> e where a, b, e are terminals.  b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] S -> id: = E   S;S   if E then S else S   do S while E. E -> id + id   id b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]				
b) What kinds of source program errors would be detected during lexical analysis? [7M] Explain.  4. a) State and explain the rules used to construct the LR(1) items. [7M] b) Discuss the evolution order of SDTs. Also write its applications. [7M]  5. a) What is a three address code? What are its types? How it is implemented? [7M] b) What are the one-pass code generation methods? Explain any one. [7M]  6. a) Generate the flow-graphs for the following expressions: [7M] S -> id: = E   S;S   if E then S else S   do S while E. E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]	3.	a)	S -> AaAb   BbBb A -> e	[7M]
<ul> <li>b) Discuss the evolution order of SDTs. Also write its applications. [7M]</li> <li>5. a) What is a three address code? What are its types? How it is implemented? [7M]</li> <li>b) What are the one-pass code generation methods? Explain any one. [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: [7M]</li> <li>b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.</li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ul>		b)	What kinds of source program errors would be detected during lexical analys	is? [7M]
<ul> <li>b) Discuss the evolution order of SDTs. Also write its applications. [7M]</li> <li>5. a) What is a three address code? What are its types? How it is implemented? [7M]</li> <li>b) What are the one-pass code generation methods? Explain any one. [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: [7M]</li> <li>b) Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.</li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ul>	4.	a)	State and explain the rules used to construct the LR(1) items.	[7M]
<ul> <li>b) What are the one-pass code generation methods? Explain any one. [7M]</li> <li>6. a) Generate the flow-graphs for the following expressions: [7M] <ul> <li>S -&gt; id: = E   S;S   if E then S else S   do S while E.</li> <li>E -&gt; id + id   id</li> <li>b) Why garbage collection is important for code optimization? Explain garbage [7M] collection by using reference counting.</li> </ul> </li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ul>			•	
<ul> <li>6. a) Generate the flow-graphs for the following expressions:  S -&gt; id: = E   S;S   if E then S else S   do S while E.  E -&gt; id + id   id  b) Why garbage collection is important for code optimization? Explain garbage [7M] collection by using reference counting.</li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ul>	5.	,	* <del>*</del>	
S -> id: = E   S;S   if E then S else S   do S while E.  E -> id + id   id  b) Why garbage collection is important for code optimization? Explain garbage [7M] collection by using reference counting.  7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]		b)	What are the one-pass code generation methods? Explain any one.	[7M]
<ul> <li>b) Why garbage collection is important for code optimization? Explain garbage [7M] collection by using reference counting.</li> <li>7. a) Prove that simple code generation algorithms allocate the registers efficiently. [7M]</li> </ul>	6.	a)	$S \rightarrow id := E \mid S; S \mid if E \text{ then } S \text{ else } S \mid do S \text{ while } E.$	[7M]
		b)	Why garbage collection is important for code optimization? Explain garba	nge [7M]
b) Discuss the transformations that are characteristic of peephole optimizations. [7M]	7.	a)	Prove that simple code generation algorithms allocate the registers efficiently.	[7M]
		b)	Discuss the transformations that are characteristic of peephole optimizations.	[7M]

## III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019 COMPILER DESIGN

(Computer Science and Engineering)

11	me: 3	Note: 1. Question Paper consists of two parts (Part-A and Part-B) 2. Answer ALL the question in Part-A 3. Answer any FOUR Questions from Part-B	irks: 70
		PART -A (14	Marks)
	a)	Define Boot strapping.	[2M]
	b)	Give the rules to find the first function.	[2M]
	c)	List the properties of LR parser.	[2M]
	ď)	How is object code different from intermediate code generation?	[3M]
	e)	Write the limitations of access links.	[3M]
	f)	What is peephole? What peephole optimizations can be performed on code?	[2M]
		$\underline{PART - B} \tag{56}$	Marks)
	a)	Consider a hypothetical programming language that has only integer and floating point's constants, data declaration and assignment statements. Describe in detail the steps involved in design of lexical analyzer for this language.	
	b)	Describe the need and functionality of linkers, assemblers and loaders.	[7M]
	a)	What is an LL(1) grammar? Can you convert every context free grammar into LL(1).	[7M]
	b)	Consider the following grammar $E \rightarrow T + E T$ $T \rightarrow V*T V$	[7M]
		V -> id Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.	
	a)	Justify how LALR parsing is efficient over SLR parsing.	[7M]
	b)	Write the procedure for eliminating left recursion from SDTs.	[7M]
	a)	What is the role of type system in type checker? Write the syntax directed definition for type checker.	[7M]
	b)	Explain the steps involved in partitioning a sequence of three address statements into basic blocks.	[7M]
	a)	What is runtime stack? Explain the storage allocation strategies used for recursive procedure calls.	[7M]
	b)	Discuss about register allocation and assignment in target code generation.	[7M]
	a)	Explain the structure preserving transformations of local machine independent optimization.	[7M]
	b)	What is Data flow Analysis? Explain its role in code optimization.	[7M]