Registration no: | | | | | | | | | | |

Total Number of Pages: **M.TECH**
CSPE202

2nd Sem MTech Regular/ Back Examination – 2014-15
COMPILER CONSTRUCTION
BRANCH(S): COMPUTER SCIENCE AND ENGINEERING/ INFORMATION TECHNOLOGY
Time: 3 Hours
Max marks: 70
Q.CODE:T283

Answer Question No.1 which is compulsory and any five from the rest.
The figures in the right hand margin indicate marks.

Q1 Answer the following questions: (2 x 10)

a) Define backpatching. Mention the functions that are used in backpatching?
b) Define peephole optimization. List the characteristics of peephole optimization.
c) What is in the stack for an LR(1) parser? How would you use the LR(1) stack to do error recovery?
d) Write quadruples, triples and indirect triples for the following expression
x[i] := y
x := y[i]
e) Define and differentiate between compile time errors and runtime errors.
f) Explain the machine dependent and machine independent code optimization. What are their advantages?
g) Explain why lexical analysis is usually a relatively slow phase of the compilation process
h) Define Stages, Phases and Passes in compiling process.
i) Explain the mechanism of bootstrapping and porting to design new compilers
j) Define Stages, Phases and Passes in compiling process.

Q2 a) Name the five main stages of a compiler and describe the function of each. What is the benefit of running stages in parallel rather than sequentially? (5)

b) Write the algorithm for predictive parser and parse the input expression: (5)

x-2∗y using the below given grammar G:

E→E+T | E-T | T

T→T*F | T/F | F

F→num | id

Q3 a) What are the various data structures used to construct symbol table in main memory? What information is recorded in the Symbol Table of a compiler for a block structured language?. (5)

b) Considering the grammar G: (5)

S → SuT | SvT | T

T → TvU | U

$U \rightarrow a$

Where, v has higher precedence than u.

(i) What difficulties arise in top-down parsing of strings in G?

(ii) Rewrite the productions of G to give a new grammar G_1 that does not have the difficulties of G.

Q4 a) Write the algorithms for creation of FIRST and FOLLOW set for a given grammar? (5)

b) Explain the working principle of operator precedence parsing algorithm. Explain the parsing action for the input string **id_1–id_2/ id_3*id_4↑id_5– id_1** with the reference to the operator precedence relation table given below. (5)

	–	*	/	↑	id	$
–	•>	<•	<•	<•	<•	•>
*	•>	•>	<•	<•	<•	•>
/	•>	•>	•>	<•	<•	•>
↑	•>	•>	•>	<•	<•	•>
id	•>	•>	•>	•>		•>

Q5 a) How is the Symbol table involved in the interactions between the different components of the compiler and in error detection? Give a simple example in each case. (5)

b) What is the objective of intermediate code generation? Generate three address code for the following code segment. (5)

```
sum = 0;
prod = 1;
for ( i =1; i <= 30; i++)
sum = sum + a[i];
prod = prod * a[i];
```

Q6 Define LL(1) parser? Change the following grammar to an LL(1) grammar by using left-recursion elimination and left factoring. Then construct the LL(1) table. (10)

S→ a C | a B
B→ B b | b
C→ b c

Q7 Describe the structure of LR parser. Consider the following grammar (10)

E → (L) | a
L → L , E | E

[a] Construct DFA of LR (0) items for this grammar.

[b] Construct the LR (1)-parsing table.

[c] Show the parsing stack and actions of an LR(1) parser for the input string
((a),a,(a,a))

Q8 (2.5 x 4)

a) Briefly specify the features of a Predictive Parser and its operation.
b) What is in the stack for an LL(1) parser? How would you use the LL(1) stack to do error recovery?
c) What are the conditions for a grammar to be LL(1)? Give an example of a selection structure in a procedural language that may cause problems in generating an LL(1) grammar. Briefly explain the problem and its resolution.
d) What is the relationship between finite state automata and LR transition diagrams?