

**SEVENTH SEMESTER EXAMINATION-2010
COMPILER DESIGN**

Time: 3 Hours

Max. Marks: 70

**Answer Question No.1, which is compulsory, and any five from the rest.
The figure in the right-hand margin indicates marks.**

1. Answer the following questions: **(20)**
- (a) Explain why lexical analysis is usually a relatively slow phase of the compilation process.
 - (b) What is peephole optimizations?
 - (c) What is cross compiler? How is the bootstrapping of compiler done to a second machine?
 - (d) What is the relationship between finite state automata and LR transition diagrams?
 - (e) Define and differentiate between phase and pass of a compiler?
 - (f) Justify the multi-pass structure of compiler construction.
 - (g) Explain the difference between Bottom-up and Top-down parsing?
 - (h) Explain why an ambiguous grammar cannot be LL(1).
 - (i) List all the LR(0) items for the following grammar: $S \rightarrow AS|b, A \rightarrow SA|a$
 - (j) What do you mean by underlying source grammar. Explain with an example.

2. (a) How SLR parser are different form LR parser? Construct the SLR parser for the following grammar: $A \rightarrow (A)|b$ Show the parsing action for the string ((b))? **(8)**

(b) identify lexeme and suggest reasonable attributes value for tokens?

```
int max(i,j)
{
    return i > j ? i : j;
}
```

(2)

3. (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) Discuss different symbol table organizations? Explain how scope rules and the block structure of a programming language influence the symbol table organization?**(5)**

4. (a) Show that the following grammar is LL(1) $S \rightarrow A, A \rightarrow aB, B \rightarrow bBC|f, C \rightarrow g.$ **(5)**

(b) Write quadruples, triples and indirect triples for the following expressing

```
x[i] := y
x := y[i]
```

(5)

5. Discuss the structure of LALR parser? What do you mean by single symbol look ahead? Describe an LALR(1) parser for the following grammar: $S \rightarrow B, B \rightarrow beginDAend, D \rightarrow Dd;|\epsilon, A \rightarrow A;E|E, E \rightarrow B|s$ **(10)**

6. (a) Explain the working principle of operator precedence parsing algorithm? Explain the parsing action for the input string $id_1/id_2*id_3^id_4-id_5$ with the reference to the operator precedence relation table given below. **(5)**

	-	*	/	^	Id	\$
-	.>	<.	<.	<.	<.	.>
*	.>	.>	<.	<.	<.	.>
/	.>	.>	.>	<.	<.	.>
^	.>	.>	.>	<.	<.	.>
id	.>	.>	.>	.>		.>

(b) Define and discuss the objectives of SDTS. What is difference between s-attributed and l-attributed grammar? **(5)**

7. (a) What is the objective of intermediate code generation? What is the different form of intermediate code generated by intermediate code generation phase? Generator three address code for the following code segment:

```
main()
{
    int a = 1, z = 0;
    int b [ 10];
    while (a <= 10)
        b [a] = 2**a;
        z = b [a] +a;
}
```

(5)

(b) What is the need of a symbol table in compilation process? Explain updating the process of the symbol table during Syntax analysis and Intermediate code generation? **(5)**

8. (a) Identify lexeme and suggest reasonable attribute value for tokens?

```
int max(i,j)
{
    return i > j ? i : j;
} (2.5)
```

- (b) Compute FIRST and FOLLOW set for all the non-terminals for the following grammar:

$S \rightarrow P\$$, $P \rightarrow \{D;C\}$, $D \rightarrow d,D|d$, $C \rightarrow c,C|c$
(2.5)

- (c) Explain how scope rules and the block structure of a programming language decides the structure of symbol table? **(2.5)**

- (d) What is cross compiler? How is the bootstrapping of compiler done to a second machine? **(2.5)**