Code No: R1631051 **R16** SET - 1

III B. Tech I Semester Regular Examinations, October/November - 2018

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**

~~~~~~

PART –A

1. a) What is a preprocessor? Mention its objectives. [2M]

b) What is recursive decent parsing? [2M]

c) Define inherited and synthesized attributes. [2M]

d) What is three-address code? Give an example. [3M]

e) Draw the typical structure of an activation record. [3M]

f) What is dead code? [2M]

PART -B

2. a) Write regular expressions for the following languages: [7M]
i) All strings of lowercase letters that contain the five vowels in order.
ii) All strings of lowercase letters in which the letters are in ascending lexicographic order.
iii) All strings of a's and b's with an even number of a's and an odd number of b's.

b) Differentiate between static and dynamic scoping. [7M]

3. a) Present the formal definition and notational conventions of CFG. [7M]

b) Explain the procedure for eliminating ambiguity from a grammar. Give an example. [7M]

4. a) Differentiate between LR(1), Canonical-LR and LALR parsing methods. [7M]

b) Below grammar generates binary numbers with a "decimal" point: [7M]

$S \rightarrow L . L \mid L$
$L \rightarrow LB \mid B$
$B \rightarrow 0 \mid 1$

Design an L-attributed SDD to compute S.val, the decimal-number value of an input string.

5. a) Give Three-Address Code and it's quadruple representation for the assignment: [6M]

$a = b * - c + b * - c ;$

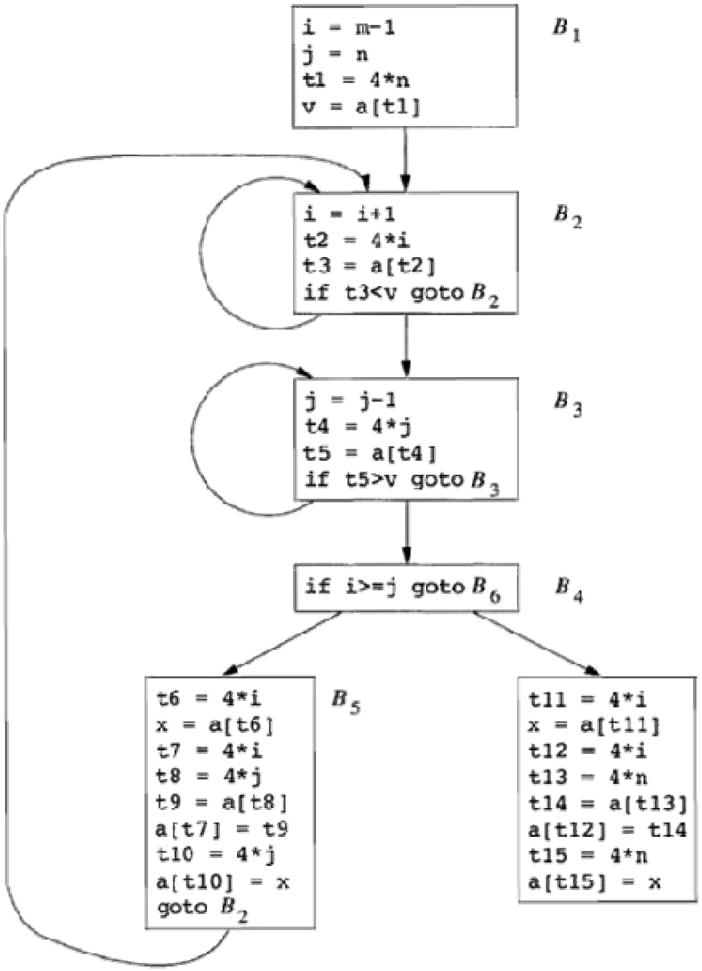
b) Discuss in detail about type synthesis and type inference. [8M]
~~~~~~

6. a) What are the limitations of access links? How displays solve those issues? Explain an example. [7M]

b) Generate code for the following three-address statements assuming a and b are arrays whose elements are 4-byte values: [7M]

x = a [i]
y = b [j]
a [i] = y
b [j] = x

7. a) Identify and eliminate global common subexpressions in the flow graph below: [9M]



```
i = m-1
j = n
t1 = 4*n
v = a[t1]
```

```
i = i+1
t2 = 4*i
t3 = a[t2]
if t3<v goto B2
```

```
j = j-1
t4 = 4*j
t5 = a[t4]
if t5>v goto B3
```

```
if i>=j goto B6
```

```
t6 = 4*i
x = a[t6]
t7 = 4*i
t8 = 4*j
t9 = a[t8]
a[t7] = t9
t10 = 4*j
a[t10] = x
goto B2
```

```
t11 = 4*i
x = a[t11]
t12 = 4*i
t13 = 4*n
t14 = a[t13]
a[t12] = t14
t15 = 4*n
a[t15] = x
```

b) Explain data-flow abstraction with an example. [5M]

*****

Code No: R1631051 **R16** SET - 2

III B. Tech I Semester Regular Examinations, October/November - 2018

COMPILER DESIGN

(Computer Science and Engineering)

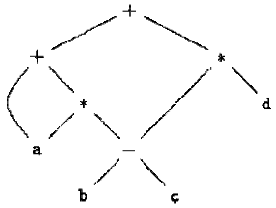
Time: 3 hours Max. Marks: 70

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**

~~~~~~

PART –A

1. a) What happens in Analysis and Synthesis phases of compilation? [2M]
b) Define an ambiguous grammar. [2M]
c) What is lookahead-LR parsing? [2M]
d) Compute three-address code for the DAG below: [3M]



e) What does heap and stack areas of run-time memory store? [3M]
f) Define a global common sub expression. [2M]

PART -B

2. a) How compilers can be used for optimization in parallel systems? [7M]
b) With a suitable transition diagram, explain recognition of keywords and identifiers. [7M]

3. a) Consider the context-free grammar: S -> S S + \ S S * \ a. For the string aa + a* give a leftmost derivation, rightmost derivation and a parse tree. [7M]
b) Construct SLR parsing table for the grammar in above question. [7M]

4. a) Show that the following grammar: [7M]

S → Aa | bAc | Bc | bBa
A → d
B → d

is LR(1) but not LALR(l).
b) Discuss in detail about dependency graphs with suitable examples. [7M]

5. a) Write about type inference for polymorphic functions. [7M]
b) Translate the arithmetic expression a[i] = b*c - b*d into a syntax tree, quadruples and triples. [7M]

6. a) What are the principles associated with designing calling sequences and the layout of activation records? [7M]
~~~~~~

b) Generate code for the following three-address statements assuming stack allocation, where register SP points to the top of the stack. [7M]

```
call p
call q
return
call r
return
return
```

7. a) Discuss about copy propagation and dead code elimination. [7M]

b) With suitable examples, explain about live-variable analysis. [7M]

*****

Code No: R1631051 **R16** SET - 3

III B. Tech I Semester Regular Examinations, October/November - 2018

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**

~~~~~~

PART –A

1. a) List any 4 compilers and 2 interpreters you know. [2M]
b) What is the key difference between lexical analysis and parsing? [2M]
c) What is syntax-directed definition? [2M]
d) Give three-address code for the statement: do i = i + 1 ; while (a [i] < v); [3M]
e) What is an activation link? Give an example. [3M]
f) Define a transfer function. [2M]

PART -B

2. a) What are program translators? Explain. [7M]

b) Describe the languages denoted by the following regular expressions: [7M]
(i) (a|b)*a(a|b)(a|b.
(ii) a*ba*ba*ba*

3. a) Give an algorithm to eliminate productions containing useless symbols from a grammar. [7M]
b) Compute FIRST and FOLLOW for the grammar: S -> S S + \ S S * \ a [7M]

4. a) Present the algorithm for LALR parsing table construction. [7M]
b) For the grammar below: [7M]
$E \rightarrow E + T \mid T$
$T \rightarrow num\ .\ num \mid num$
Give an SDD to determine the type of each term T and expression E.

5. a) Explain the value-number method for constructing the nodes of a DAG. [7M]
b) Generate three-address code for the grammar below: (B is a Boolean expressing and S is a statement) [7M]
$S \rightarrow if\ (\ B\ )\ S_1$
$S \rightarrow if\ (\ B\ )\ S_1\ else\ S_2$
$S \rightarrow while\ (\ S\ )\ S_1$

6. a) List and explain different subdivisions of run-time memory. [4M]
~~~~~~

b) Construct flow graph for the three-address code equivalent of the below code: [10M]

```
for (i=0; i<n; i++)
        for (j=0; j<n; j++)
                c[i] [j] = 0.0;
for (i=0; i<n; i++)
        for (j=0; j<n; j++)
                for (k=0; k<n; k++)
                        c[i][j] = c[i][j] + a[i][k]*b[k][j];
```

7. a) Optimize the code given below, by eliminating common subexpressions, performing reduction in strength on induction variables, and eliminating all the induction variables. [7M]

```
   dp = 0.
   i = 0
L: tl = i*8
   t2 = A[tl]
   t3 = i*8
   t4 = B[t3]
   t5 = t2*t4
   dp = dp+t5
   i = i+1
   if i<n goto L
```

b) Explain the procedures for elimination of unreachable code and algebraic simplifications in Peephole Optimization [7M]

*****

Code No: R1631051 **R16** SET - 4

III B. Tech I Semester Regular Examinations, October/November - 2018

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**

~~~~~~

PART –A

1. a) What is the purpose of Loader/Linker in language processing? [2M]
b) What are left-most and right-most derivations? [2M]
c) What is an annotated parse tree? Give an example. [2M]
d) Give directed acyclic graph for the expression: a + a * (b - c) + (b - c) * d. [3M]
e) What are the basic functions of the memory manager? [3M]
f) Define a semi lattice. [2M]

PART -B

2. a) List and explain in detail about different phases of compilation. [9M]
b) What are the problems that might arise while recognizing tokens? [5M]

3. a) Design grammars for the following languages: [7M]
(i) The set of all strings of 0s and 1s, such that every 0 is immediately followed by at least one 1.
(ii) The set of all strings of 0s and 1s that are palindromes.
b) Explain the structure of LR parsing table, with an example. [7M]

4. a) Discuss about the Dangling-Else ambiguity. [7M]
b) Explain the procedure for eliminating left recursion from SDTs. [7M]

5. a) Explain about one-pass code generation using back patching. [7M]
b) Construct parse trees for the types in t [2] [3] and char [10]. [7M]

6. a) The following C program computes Fibonacci numbers: [7M]

```
int  f (int  n) {
        int  t,s;
        if (n < 2) return 1;
        s = f(n-1);
        t = f(n-2);
        return s+t;
}
```

Suppose that the activation record for f includes the following elements in order: return value, argument n, local s, and local t. Show the complete activation tree for the call f(5).
b) Discuss the design issues of Code Generator. [7M]

7. a) Explain about the method of computing transfer equations for reaching definitions. [7M]
b) Construct an algorithm that will perform redundant-instruction elimination in a sliding peephole on target machine code. [7M]

*****
~~~~~~