Dr. Sanjeev M. Raikar

IT 7 – 2 (RC) 2007-08

B.E. (IT) (Semester – VII) (RC) (2007-08) Examination, Nov./Dec. 2017
PRINCIPLES OF COMPILERS

Duration : 3 Hours

Max. Marks : 100

Instructions : *i) Answer **any 5** questions, atleast **one** from **each** Module.*
*ii) Assume necessary **data**.*

Module – I

1. a) With the help of a neat diagram explain the structure of compiler. **6**
   b) Write note on :
      i) Bootstrapping
      ii) Porting. **8**
   c) Discuss briefly the compiler writing tools. **6**

2. a) For the given LEX program, give the DFA implementation of the lexical analyser for the following regular definitions.

```
% {A, AB, ABB%}
%%
{a} {yylval = 1; return (A);}
{a*b+} {yylval = 2; return (AB);}
{a*b*} {yylval = 3; return (ABB);}
%%
```

   b) Identify lexemes, tokens and pattern in the following. **8**

```
void add (int i, int j)
{
    int t;
    t = i + j;
}
```

   c) Write a note on : **5**
      i) Preprocessor
      ii) Front end and back end of compiler
      iii) Analysis phase and synthesis phase of compi. **7**

P.T.O.

Module – II

3. a) Distinguish between top-down and bottom-up parsing. 4

   b) Eliminate left recursion from the following grammar. 3

   $A \rightarrow Ac \mid Aad \mid bd \mid c$

   c) Construct predictive parsing table for the grammar. 12

   $E \rightarrow E + T \mid T$

   $T \rightarrow TF \mid F$

   $F \rightarrow F * \mid a \mid b$

   d) Define a handle. 1

4. a) Check whether the following grammar is SLR(1) or not. 12

   $S \rightarrow cA \mid ccB$

   $A \rightarrow cA \mid a$

   $B \rightarrow ccB \mid b$

   b) Explain shift Reduce Parser. 8

Module – III

5. a) Write the translation scheme for flow of control statements. 8

   b) Translate (a or b) and (c < d) and (d < e) into three address statements using backpatching. 8

   c) Write a note on : 4

      i) Lexical errors

      ii) Syntax errors.

6. a) What is a symbol table ? Explain the contents of symbol table and data structures used to create symbol table. 10

   b) Explain with the help of diagram, the structure of activation record. 5

   c) Explain static scope and dynamic scope. 5

Module – IV

7. a) Construct a DAG for the following code.

   $x = y + z$

   $m = y - z$

   $l = x + m$

   $p = x - m.$

   b) Describe the local transformations that can be applied to basic blocks.

   c) Explain the labeling algorithm and code generation from a labeled tree.

Dr. Sunil M. Raikar

8. a) Write a three address code for the fragment given below. Convert it to basic blocks and draw the flow graph.

```
int count = 0, x;
for (int i = 0; i < 10; i ++)
    {
    if (i < 5)
    count = count + 1;
    else
    count = count + 2;
    }
    x = count;
```

b) Compute the cost of the following : 10

$MOV\ *R_1, *R_0$

$ADD\ *R_2, *R_0$

c) Explain the following techniques of code optimization. 2

i) Dead code elimination

ii) Constant folding

iii) Common subexpression elimination

iv) Copy propagation.

8

Deepali M. Raikar

IT 7 – 2 (RC) 2007-08

B.E.(IT) (Semester – VII) (RC) (2007-08) Examination,
November/December 2018
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

Instructions : 1) Answer **any 5** questions, atleast **one** from **each** Module.
2) Assume necessary **data**.

MODULE – I

1. a) Explain the compilation phases with a neat diagram. 8

b) Explain the following terms with examples : 6

i) Lexemes

ii) Patterns

iii) Tokens

c) What is the role of finite automata in Lexical analysis ? 6

2. a) Construct the minimized DFA for the following expression : (a/b)*abb. 8

b) Consider the following lex program :

```
% {A, B %}
% %
{ab*}        {yylval = 1; return (A);}

{a+b*}       {yylval = 2; return (A);}

{a*b+}       {yylval = 3; return (B);}
%%
```

Give the implementation of Lexical analyzer using DFA. 8

c) Discuss briefly the compiler writing tools. 4

P.T.O.

MODULE – II

3. a) Explain the features of recursive descent parser.

   b) Construct the Predictive Parsing table for the given grammar and parse the string aabbab. 1

   $S \rightarrow aA \mid aB \mid €$

   $A \rightarrow bS \mid aAA$

   $B \rightarrow aS \mid bBB$

   c) Eliminate the left recursion from the following grammar. 5

   $A \rightarrow Ac \mid Aad \mid bd \mid €$

4. a) Give the following grammar : 8

   $S \rightarrow P$

   $P \rightarrow *R \mid R$

   $R \rightarrow (S) \mid id$

   i) Obtain collection of sets of LR(0) items.

   ii) Draw goto graph.

   b) Explain the importance of handle in the shift reduce parser. 4

   c) Construct LALR parsing table for the following grammar and parse the string cccd. 8

   $S \rightarrow CC$

   $C \rightarrow cC \mid d$

MODULE – III

5. a) Convert the following code to Quadruples, Triples and indirect triples : 6

   $y = a*a + 2*a*b$

   b) Write the translation scheme for Assignment statements.

   c) Describe the contents of Symbol table and explain the data structure used in symbol table management. 8

6. a) Write the translation scheme for flow of control statements. 6

   b) Describe various storage allocation strategies. 8

   c) Provide two representations of syntax tree. 6

MODULE – IV

7. a) Provide an algorithm to partition 3-address code into basic block. Indicate the basic blocks and draw flow graph for the following source code : 12

```
i = 0;
val = 0;
while (i++<10)
{
    i = 2*i;
    val+=i;
}
```

b) Describe various issues in design of code generator.

8. a) Obtain assembly language code generated after compiling the following statements. Show the contents of address descriptor and register descriptor. 5

t = a – b

v = t + u

b) Write a short note on Peephole optimization. 8

c) Explain the following code optimization techniques : 6

a) Copy Propagation 9

b) Common subexpression elimination

c) Dead Code elimination.

Deepali M. Raikar

IT 7 – 2 (RC)

B.E. (IT) (Semester – VII) (RC – 2007-08) Examination, May/June 2018

PRINCIPLES OF COMPILERS

Duration : 3 Hours

Max. Marks : 100

Instructions : *i) Answer **any 5** questions, atleast **one** from **each** Module.*

*ii) **Assume** necessary **data**.*

MODULE – I

1. a) Show the output of each phase of compilation for the following statements :

```
int t = 0;
int i;
for (i = 0; i < 5; i ++)
t = t + 5;
```

10

b) With the help of T diagrams explain the process of boot strapping. **6**

c) Explain the following :

i) Preprocessor **4**

ii) Loader.

2. a) Write a note on LEX tool. Give a LEX program to identify integers from a given string. **8**

b) Explain with the help of pseudocode the use of the sentinel character in the input buffering. **6**

c) Distinguish between

i) Front-end and back-end of compiler.

ii) Compiler and interpreter.

iii) Lexeme and Token. **6**

P.T.O.

MODULE – II

3. a) Eliminate left recursion in the grammar

$A \rightarrow ABd \mid Aa \mid a$

$B \rightarrow Be \mid b$

b) Left factorize the grammar

$S \rightarrow abBde \mid aCde \mid abBd \mid a$

$B \rightarrow bd \mid bc \mid bde \mid \in$

$C \rightarrow c \mid \in$

c) Write a note on YACC.

d) Test whether the following grammar is LL (1) or not. Construct the parsing table and parse the string abdgg.

$S \rightarrow AB \mid gDa$

$A \rightarrow ab \mid c$

$B \rightarrow dc$

$C \rightarrow gc \mid g$

$D \rightarrow fD \mid g$

4. a) Construct operator precedence parser for the following grammar :

$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow T, S \mid S$

Parse the input (a, $\wedge$).

b) Construct the LALR parsing table for the following grammar :

$S \rightarrow CC$

$C \rightarrow aC$

$C \rightarrow d$

MODULE – III

5. a) Explain the translation to produce three address code for boolean expressions.

Deepali M. Raikar

b) Obtain a Quadruple, Triple, Syntax tree and indirect triple for the following expression

(a – b) * (c – d) – (a + b). 8

c) Explain :

i) Syntactic errors.

ii) Semantic errors. 4

6. a) Describe the different data structures used for maintaining symbol table. 8

b) Explain error recovery strategies. — Panic mode recovery, Phase level, Global Correction, Error Production. 8

c) Explain back patching. 4

MODULE – IV

7. a) Explain the following :

i) Register descriptor.

ii) Address descriptor. 6

b) Describe the different issues in the design of code generator. 8

c) With the help of an example, explain the code generation algorithm. 6

8. a) Construct DAG for the following :

(a * b) + (c – d) * (a * b) + b. 4

b) Consider

```
int a [10]
sum = 0;
for (i = 0; i <= 10; i ++)
sum = sum + a [i];
```

Construct basic blocks and flow graphs. 6

c) Explain the following techniques of code optimization with examples. 6

i) Induction variable

ii) Dead code elimination.

d) Write a note on next use information. 4

Deepali M. Raikar

Paper / Subject Code: BE747 / Principles of Compilers

BE747

Total No. of Printed Pages:3

B.E. (Information Technology) Semester- VII (Revised Course 2016-17)
EXAMINATION NOV/DEC 2019
Principles of Compilers

Duration : Three Hours] [Total Marks :100]

Instructions:
1. Attempt two questions from Part-A, two questions from Part-B and one from Part-C.
2. Figures to the right indicate marks.
3. Make suitable assumptions wherever necessary.

Part-A

Answer any two questions from the following:

1. a) Identify lexeme, token and pattern in the following C statements:-
int × 1, y;
×1= y+35; [4]

1. b) Explain the phases of Compiler. [8]

1. c) Define the following terms. [4]
I. Linker
II. Loader
III. Pre-processor
IV. Interpreter

1. d) Construct the DFA for the regular expression RE=(a/b)*abb. [4]

2. a) Eliminate left recursion from the following grammar:-
S -> P
R ->Pe|Pd|dR
R -> f [3]

b) Check whether the following grammar is an LL(1) grammar
S->iEtS|iEtSes|a
E->b

Also define the FIRST and FOLLOW procedure [10]

c) Write a short note on YACC tool. [7]

3. a) Explain the features of recursive decent parser

3. b) Explain the handle in the shift reduce parser

3. c) Construct SLR parsing table for the following grammar:-
S -> E
E-> $T|E+T|E-T$
T-> $i|(E)$

PART-B

Answer any two questions from the following:

4. a) What is a symbol table? Explain the contents of symbol table and data structures used to create symbol table.

b) Explain with the help of a diagram the structure of activation record.

c) Explain static scope and dynamic scope.

5. a) Translate the following statement:

A = B + C * D[I] into

i) Quadruples
ii) Triples
iii) Indirect triples

b) Explain the characteristic of peephole optimization.

6. a) Explain the concept of basic blocks and flow graph with the example

b) Explain the concept of DAG and Construct the DAG for the following example

a=b*c
d=b
e=d*c
b=e
f=b+c
g=d=f

PART-C

Answer **any one questions from the following:**

a) Obtain the assembly code generated after compiling the following statements. Show the content of address and register descriptor.

T1 = a + b
T2 = t1 + c

[10]

b) Explain the following terms:
i) Register descriptor
ii) Address descriptor

[5]

c) Define the input buffering scheme. Explain with the help of pseudopod the use of the sentinel character in the input buffering. [5]

a) Give the following grammar
S->P
P->*R|R
R->(S)|id

Obtain collection of set of LR(O) items. [4]

Draw goto graph [4]

b) Explain the following terms with examples: [6]
i. Lexemes
ii. Patterns
iii. Tokens

c) Explain the process of Bootstrapping with the help of T-diagrams. [6]

Deepali M. Raikar

Paper / Subject Code: BE830 / Principles of Compilers

BE830

Total No. of Printed Pages:2

B.E. (Information Technology) Semester- VII (Revised Course 2007-08)
EXAMINATION Nov/Dec 2019
Principles of Compilers

Duration : Three Hours] **[Total Marks :100]**

Instructions:
1) Answer any five full questions, with at least one from each module.
2) Figures to the right indicate marks
3) Make suitable assumptions wherever necessary

Module – I

Q1 a) Define the following terms:- **4**
i) Linker
ii) Loader
iii) Ore-processor
iv) Interpreter

b) Compile the following set of C statements by showing the input and output of each phase of compilation. **10**

```
float a= 10;
for (int i= 0; i<10 ; i++)
{
a=a*i;
}
```

c) With the help of a neat diagram explain the structure of Complier. **6**

Q2 a) Explain the specification of LEX tool. **6**
b) Differentiate between analysis and synthesis phase of compiler. **4**
c) Describe the role of Lexical analyser. **6**
d) What a short note on bootstrapping process. **4**

MODULE – II

Q3 a) Construct the LALR parsing table for the given grammar. **12**

$$E \rightarrow E + E|EE|E^*|(E)|a\ |b|E$$

b) Discuss the features of YACC tool. **4**
c) Explain the importance of handle in the shift reduce parser. **4**

Q4 a) Explain the features of recursive descent parser. **4**
b) Discuss the difficulties faced by the backtracking parser. How do you overcome those difficulties? **6**
c) Consider the following grammar. **10**

$$S \rightarrow a|\hat{}|(T)$$

Deepali Raikar

$$T \rightarrow T, S \mid S$$

Construct an operator precedence parsing table and check if the string ((a,a),^, (a)) can be parsed or not.

MODULE –III

Q.5

a) Convert the following code to quadruples, triples and indirect triples. 6

$$z = a + b^* - c/d$$

b) Describe the contents of symbol Table 6

c) Explain the syntax directed translation for Booleans. 8

Q.6

a) What are different ways of intermediate Language Graphical Representation? 6

b) Explain the translation scheme for assignment statement. 7

c) Provide the structure of an activation record with diagram. 7

MODULE –IV

Q.7

a) Construct the DAG for the following :- 5

```
t1=a+b
x=t1
t2=a-b
y=t2
z=x+y
```

b) Write a 3 – address code for the fragment given below. Convert it to the basic blocks and draw the flow graph. Further optimize the code 10

```
count=0;
for(int i=0 ; i< 10 ; i++)
{
count = count +1;
m=100;
x[i] = count ;
}
```

c) Write short note on next use information 5

Q.8

a) Explain the issues in the design of code generator. 8

b) Explain the following techniques of code optimization:- 12

i) Dead code Elimination
ii) Common sub expression elimination
iii) Copy propagation
iv) Constant folding

IT 7 – 2 (RC) 2007-08

B.E.(IT) (Semester – VII) (RC) (2007-08) Examination, November/December 2018
PRINCIPLES OF COMPILERS

Duration : 3 Hours

Total Marks : 100

Instructions : 1) Answer **any 5** questions, atleast **one** from **each** Module.
2) Assume necessary **data**.

MODULE – I

1. a) Explain the compilation phases with a neat diagram. **8**

b) Explain the following terms with examples : **6**

i) Lexemes

ii) Patterns

iii) Tokens

c) What is the role of finite automata in Lexical analysis ? **6**

2. a) Construct the minimized DFA for the following expression : (a/b)*abb. **8**

b) Consider the following lex program :

```
% {A, B %}
% %
{ab*}        {yylval = 1; return (A);}

{a+b*}       {yylval = 2; return (A);}

{a*b+}       {yylval = 3; return (B);}
%%
```

Give the implementation of Lexical analyzer using DFA. **8**

c) Discuss briefly the compiler writing tools. **4**

P.T.O.

MODULE – II

3. a) Explain the features of recursive descent parser. 5

b) Construct the Predictive Parsing table for the given grammar and parse the string aabbab. 10

$S \rightarrow aA \mid aB \mid €$

$A \rightarrow bS \mid aAA$

$B \rightarrow aS \mid bBB$

c) Eliminate the left recursion from the following grammar.

$A \rightarrow Ac \mid Aad \mid bd \mid €$ 5

4. a) Give the following grammar : 8

$S \rightarrow P$

$P \rightarrow *R \mid R$

$R \rightarrow (S) \mid id$

i) Obtain collection of sets of LR(0) items.

ii) Draw goto graph.

b) Explain the importance of handle in the shift reduce parser. 4

c) Construct LALR parsing table for the following grammar and parse the string cccd. 8

$S \rightarrow CC$

$C \rightarrow cC \mid d$

MODULE – III

5. a) Convert the following code to Quadruples, Triples and indirect triples : 6

$y = a*a + 2*a*b$

b) Write the translation scheme for Assignment statements. 8

c) Describe the contents of Symbol table and explain the data structure used in symbol table management. 6

6. a) Write the translation scheme for flow of control statements. 8

b) Describe various storage allocation strategies. 6

c) Provide two representations of syntax tree. 6

MODULE – IV

7. a) Provide an algorithm to partition 3-address code into basic block. Indicate the basic blocks and draw flow graph for the following source code : 12

```
i = 0;
val = 0;
while (i++<10)
{
    i = 2*i;
    val+=i;
}
```

b) Describe various issues in design of code generator. 8

8. a) Obtain assembly language code generated after compiling the following statements. Show the contents of address descriptor and register descriptor. 5

```
t = a – b
v = t + u
```

b) Write a short note on Peephole optimization. 6

c) Explain the following code optimization techniques : 9

a) Copy Propagation
b) Common subexpression elimination
c) Dead Code elimination.
