IT 7 – 2 (RC)

B.E. (IT) (Semester – VII) (RC) Examination, Nov./Dec. 2016
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

Instruction : *Answer **any five** questions by selecting at least **one** question from **each** Module.*

MODULE – 1

1. a) Define the following terms : **4**

   1) Linker

   2) Loader

   3) Preprocessor

   4) 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) Write a LEX program to identify the token as an identifier and explain the steps to execute the LEX file. **6**

2. a) If you have a C to Java translator running on C machine and a working C compiler. Using T diagram create a C to Java translator for Java machine. **4**

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

```
% {ABA, AB, BA %}
%%
    {A+B*} {yyval = 1, return (AB)}
    {B*A+} {yyval = 2, return (BA)}
    {A+BA} (yyval = 3, return (ABA)}.
```

   10

   c) Write a short note on the different compiler writing tools. **6**

P.T.O.

MODULE – 2

3. a) What do you mean by an ambiguous grammar ? Explain with the help of an example how to eliminate ambiguity in the grammar. 5

   b) Consider the following grammar :

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

   $T \rightarrow T, S \mid S$

   Construct a operator precedence passing table and check if the string ((a, a), $\wedge$, (a)) can be parsed or not. 8

   c) Define handle in shift reduce parsers. 2

   d) Discuss the difficulties faced by the backtracking parser. How do you overcome those difficulties ? 5

4. a) Consider the following grammar :

   $S \rightarrow aA \mid aB \mid \in$

   $A \rightarrow bS \mid aAA$

   $B \rightarrow aS \mid bBB$

   i) Calculate first and follow for the given grammar. 3

   ii) Construct predictive parsing table. 5

   iii) Parse the string aabbab. 2

   b) Construct a canonical parsing table for the following grammar : 10

   $S \rightarrow E\#$

   $E \rightarrow E + T \mid T$

   $T \rightarrow i \mid (E)$.

MODULE – 3

5. a) Explain the SDT scheme to produce 3 address code for numeric representation of assignment statement. 6

   b) Translate the following statement :

   A [i] = B [i] + C [i, j] * P into

   i) Three address code. 3

   ii) Quadruples. 2

   iii) Triples. 2

   iv) Indirect triples. 1

   c) Explain the different data structures that can be used to create symbol table. 6

6. a) Write a short note on backpatching. 5

   b) What is an activation tree ? Explain with the help of an example how it is constructed. 5

   c) Consider the following expression :

   a * b + d – e * f * g – h/e

   Generate the intermediate code using :

   i) Syntax tree. 2

   ii) Quadruples. 2

   iii) Triples. 2

   d) Write a short note on the errors that can occur in the front end phases of the compiler. 4

MODULE – 4

7. a) Explain with the help of an example the technique of peephole optimisation. 5

b) Construct a directed acyclic graph for the following statement :

A [k] = B [i] * C [i] +.D. 5

c) Write a short note on the issues in the design of the code generator. 4

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

t = a – b

v = t * c

x = v + a. 6

8. a) Explain the code generation algorithm. 6

b) Consider the following statements :

```
sum = 0.0
for (i = 0; i ≤ 15; i++)
    {sum = sum + a [i];}
    sum = sum * sum;
```

i) Construct the basic block for the above code. 4

ii) Optimise each of the block by using different optimisation techniques. 6

c) Explain the following w.r.t. code generation. 4

i) Next use information

ii) Register descriptors.

IT 7 – 2 (RC)

B.E. (I.T.) (Semester – VII) (RC) Examination, May/June 2016
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

Instructions : *1) Answer **any five** questions with atleast **one** from **each** Module.*
*2) **Assume** suitable data **if necessary.***

MODULE – I

1. a) Explain the compilation phases with a neat diagram with an example. **10**

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

      i) Lexems

      ii) Patterns

      iii) Tokens.

   c) Explain the function of preprocessor. **4**

2. a) Show the output of each phase of compilation for the following statements : **8**

      If (a < b)

       a=b + 2;

      else

       b=a + 3;

   b) Explain the specifications of a LEX tool. **6**

   c) Discuss briefly some of the compiler writing tools and mention the name of the phase where it is used. **6**

MODULE – II

3. a) Construct the predictive parsing table for the given grammar and parse the string int id1, id2 : **10**

      $D \rightarrow TV$

      $T \rightarrow$ int | float

      $V \rightarrow$ id, V|id

P.T.O.

b) Construct the Operator Precedence parsing table for the given grammar and parse the string id = **id.

$S \rightarrow L = R \mid R$

$L \rightarrow *R \mid id$

$R \rightarrow L$

4. a) Consider the following grammar :

$E \rightarrow wX|xY$

$X \rightarrow yX|z$

$Y \rightarrow yY|z$

Construct

i) LR(0) items

ii) GOTO graph

iii) SLR Parsing table.

Parse the string wyyyz.

b) Write a note on YACC parser generator.

MODULE – III

5. a) Convert the following statements into triples, quadruples and indirect triples:

a[i] = b[i] * c[i]

b) Obtain the assembly code generated after compiling the following statements. Show the contents of address descriptors and register descriptors.

u = c – d

v = u + a

c) Explain the data structures used to create symbol table.

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

b) What is an activation tree ? Explain how it is constructed with the help of an example.

c) What do you mean by Back patching and how it is been carried out ?

7. a) Write the al
trace it on t

A[k] = B[i]

b) Explain the

i) Next u

ii) Regist

iii) Addre

8. a) Write a th
basic bloc

Count = 0

for (int i=

{

cou

Ma

x[i]

}

b) Explain

MODULE – IV

7. a) Write the algorithm for the construction of the Directed Acyclic Graph and trace it on the given expression :

   A[k] = B[i] * C[j] + D. 8

   b) Explain the following with respect to code generation : 12

      i) Next use information

      ii) Register Descriptor

      iii) Address Descriptor.

8. a) Write a three-address code for the fragment given below. Convert it into the basic blocks and draw the flow graph. Further optimize the code. 12

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

   b) Explain in detail Peephole Optimization. 8

IT 7 – 2 (RC)

B.E. (I.T.) (Semester – VII) (RC) Examination, November/December 2015
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

Instructions : *1) Answer **any five** questions with atleast **one** from **each** Module.*
*2) Assume suitable data **if necessary**.*

MODULE – I

1. a) Explain the different phases of the compiler with the help of an example. 6
   b) Explain the process of bootstrapping and porting with the help of T-diagrams. 8
   c) Write a short note on the following : 6
      i) Single pass compiler and multipass compiler
      ii) Linker and loaders.

2. a) Show the output of each phase of compilation for the following statements. 8

```
If (a < b)
   a = b + 2;
else
   b = a + 3;
```

   b) What do you mean by LEX tool ? Give its specification with an example. 8
   c) Comment on the ambiguity of a grammar. 4

MODULE – II

3. a) Construct the predictive parsing table for the given grammar and parse the string aabbab 10

$S \rightarrow aA|aB|€$
$A \rightarrow bS|aAA$
$B \rightarrow aS|bBB$

   b) Construct the Canonical parsing table for the following grammar. 10

$E \rightarrow E + E \mid E^*E \mid (E) \mid id$

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

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

   b) Write a short note on YACC. 4
   c) What do you mean by handle and handle pruning ? 4

P.T.O.

MODULE – III

5. a) Consider the following statement :
   a*b+d-e*f*g-h|e
   Generate the intermediate code using
   i) Syntax tree
   ii) Quadruples
   iii) Triples
   iv) Indirect triples.

   b) Write a short note on the following :
   i) Lexical errors
   ii) Syntax errors.

   c) Explain the translation of case statements.

6. a) What is an activation tree ? Explain how it is constructed with the help of an example.

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

   c) Explain the term static scope and dynamic scope.

MODULE – IV

7. a) Construct DAG for the following code :
   $t_1 = a + b$
   $x = t_1$
   $t_2 = a - b$
   $y = t_2$
   $z = x + y$

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

   c) Explain the different issues faced in the design of the code generator.

8. a) Explain in detail Peephole Optimization.

   b) Discuss the following techniques of code optimization with examples.
   i) Induction variables ii) Dead code elimination.

   c) Write short note on next-use information.

IT 7 – 2 (RC)

B.E. (I.T.) (Semester – VII) (RC) Examination, November/December 2015
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

Instructions : *1) Answer **any five** questions with atleast **one** from **each** Module.*
*2) Assume suitable data **if necessary**.*

MODULE – I

1. a) Explain the different phases of the compiler with the help of an example. **6**
   b) Explain the process of bootstrapping and porting with the help of T-diagrams. **8**
   c) Write a short note on the following : **6**
      i) Single pass compiler and multipass compiler
      ii) Linker and loaders.

2. a) Show the output of each phase of compilation for the following statements. **8**

```
If (a < b)
   a = b + 2;
else
   b = a + 3;
```

   b) What do you mean by LEX tool ? Give its specification with an example. **8**
   c) Comment on the ambiguity of a grammar. **4**

MODULE – II

3. a) Construct the predictive parsing table for the given grammar and parse the string aabbab **10**

   $S \rightarrow aA|aB|€$
   $A \rightarrow bS|aAA$
   $B \rightarrow aS|bBB$

   b) Construct the Canonical parsing table for the following grammar. **10**

   $E \rightarrow E + E \mid E^*E \mid (E) \mid id$

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

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

   b) Write a short note on YACC. **4**
   c) What do you mean by handle and handle pruning ? **4**

P.T.O.

MODULE – III

5. a) Consider the following statement :
   a*b+d-e*f*g-h|e
   Generate the intermediate code using
   i) Syntax tree
   ii) Quadruples
   iii) Triples
   iv) Indirect triples.

   b) Write a short note on the following :
   i) Lexical errors
   ii) Syntax errors.

   c) Explain the translation of case statements.

6. a) What is an activation tree ? Explain how it is constructed with the help of an example.

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

   c) Explain the term static scope and dynamic scope.

MODULE – IV

7. a) Construct DAG for the following code :
   $t_1 = a + b$
   $x = t_1$
   $t_2 = a - b$
   $y = t_2$
   $z = x + y$

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

   c) Explain the different issues faced in the design of the code generator.

8. a) Explain in detail Peephole Optimization.

   b) Discuss the following techniques of code optimization with examples.
   i) Induction variables ii) Dead code elimination.

   c) Write short note on next-use information.

IT 7 – 2 (RC)

B.E. (IT) Semester – VII (RC) Examination, Nov./Dec. 2014
PRINCIPLES OF COMPILERS

uration : 3 Hours | Total Marks : 100

Instructions : *i) Answer **any 5** questions with **atleast 1** from **each** Module.*
*ii) Make suitable assumptions, **if required**.*

MODULE – I

1. a) Distinguish between analysis and synthesis part of a compiler. 4

   b) Show the output off each phase of compilation for the following C statements : 8

```
int p = 1 ;
if (p < 5)
 {
   p = p * 2.0 ;
 }
```

   c) Write short notes on : 8

   i) Boot strapping

   ii) Porting.

2. a) Describe the role of lexical analyzer. Provide an example of token, pattern and lexeme. 8

   b) List and explain various tools used for compiler construction. 5

   c) Give the specification of LEX tool. Also list the steps in compilation of lex program. 7

MODULE – II

3. a) What is left recursion ? Eliminate left recursion from the following grammar : 4

   i) $S \rightarrow aAb|bB$
   $A \rightarrow Aa|\in$
   $B \rightarrow Bb|\in$

   ii) $S \rightarrow P$
   $P \rightarrow Pe|Pd|dR$
   $R \rightarrow f$

P.T.O.

b) Construct predictive parsing table for the following grammar :

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

$T \rightarrow T, S|S$

Parse the input string (a, ∧).

c) Write a note on YACC parser generator.

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

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

$T \rightarrow T, S|S$

Parse the input string (a, ∧).

b) Consider the following grammar :

$P \rightarrow D ; D$

$D \rightarrow id : T$

$T \rightarrow int|char$

i) Obtain collection of LR(O) itemsets

ii) Construct SLR parsing table.

MODULE – III

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

$s = x[i] + y[i] * s$

b) Write the translation scheme for flow of control statements.

c) Explain with diagram, the structure of activation record.

6. a) Briefly describe various storage allocation strategies.

b) What is symbol table ? Explain the contents of symbol table.

c) Write the translation scheme for assignment statements.

MODULE – IV

7. a) What is DAG ? Construct DAG for the following code : 6

$t_1 = b + c$

$t_2 = d * e$

$t_3 = t_2 * t_1$

$t_3 = t_3 * f$

$x = t_1 - t_3$.

b) Explain code generation algorithm. 8

c) Explain the following with respect to code generator : 6

i) Register descriptor

ii) Address descriptor.

8. a) Write short notes on next-use information. 4

b) Write 3-address code for the fragment given below. Convert it to basic blocks and draw flow graph. 8

```
count  = 0;
result = 0;
while (count + + < 20)
    {
      inc = 2* count ;
      result = result + inc ;
    }
```

c) Explain the following code optimization techniques : 8

i) Dead code elimination

ii) Common sub-expression elimination

iii) Copy propagation

iv) Constant folding.

IT 7 – 2 (RC)

B.E. (IT) (Semester – VII) (RC) Examination, May/June 2014
PRINCIPLES OF COMPILERS

ation: 3 Hours

Total Marks: 100

Instructions : *1) Answer **any five** questions selecting atleast **one** from **each** Module.*
*2) Assume suitable data **wherever** required.*

MODULE – 1

a) Differentiate between analysis and synthesis phase of the compiler. 4

b) Explain the various phases of the compilation for a given 'C' code 8

```
float a;
a = 10.0;
for (int i = 0; i < 10; i ++)
  { a = a + i; }
```

c) Explain the process of bootstrapping and porting with the help of T-diagrams. 8

a) For a given LEX program, give the DFA implementation of lexical analyser for the following regular definitions. 8

```
% {QP, PQ, QPQ %}
%%
{Q+P*} { yyval = 1, return (QP)}
{P*Q+} {yyval = 2, return (PQ)}
{Q+PQ} {yyval = 3, return (QPQ)}
```

b) Write a short note on different compiler writing tools. 5

c) Write a short note on the following : 4

i) Compiler
ii) Assembler
iii) Linker
iv) Loader.

d) Explain the significance of regular expression in the lexical analysis phase. 3

P.T.O.

MODULE – 2

3. a) What do you mean by an ambigious grammar ? Explain with the help of an example, how to eliminate ambiguity in the grammar.

   b) Consider the following grammar

   $S \rightarrow aA / aB / \in$

   $A \rightarrow bS / aAA$

   $B \rightarrow aS / bBB$

   i) Compute the first and follow for the given grammar.

   ii) Construct the predictive parsing table for the given grammar.

   iii) Parse the string aabbab by showing the contents of the stack and input buffer.

   c) Discuss the difficulties faced by the back tracking parsers. How do you overcome those difficulties ?

4. a) Consider the following grammar

   $S \rightarrow E = E$

   $E \rightarrow E + T / T$

   $T \rightarrow T * id / id$

   i) Construct LR(1) items for the above grammar.

   ii) Construct the parsing table and check if the following string id = id + id * id can be parsed or not.

   b) What do you mean by shift reduce parsers ? Define handle pruning.

   c) Write a short note on YACC.

MODULE – 3

5. a) Explain the syntax directed translation scheme for the flow of control of Boolean expressions.

   b) Explain the structure of the activation record with the help of an example.

   c) Explain the different data structures that can be used to create symbol table.

   d) Write a short note on the syntax errors.

6. a) Translate the following statement into

   A[i] = B[i] + C [i, j] * P

   i) Three address code.

   ii) Quadruples.

   iii) Triples.

   iv) Indirect triples.

b) Explain the most closely nested rule for accessing non local names with the help of an example. 5

c) Write a short note on backpatching. 5

MODULE – 4

7. a) Write the algorithm for the construction of the directed acyclic graph and trace it on the given expressions. 6

D = B * C

E = A + B

B = B * D

A = E + B.

b) Consider the following statements given below :

X = C – D

Y = X + B [i, j]

i) Write the assembly level language code for the above statements. 3

ii) Write the contents of address and register descriptors for the above statements. 4

iii) Compute the total cost of executing the assembly level language code. 3

c) Write a short note on loop optimisation. 4

8. a) Consider the following pseudo code

```
    while (1)
{ do i = i + 1; while (a[i] < v)
    if (i ≥ j) break;
}
    X = a [i];
```

i) Convert the code to intermediate code form. 4

ii) Construct the basic blocks. 2

iii) Optimise each of the basic blocks. 4

b) Explain the code generation algorithm with the help of an example. 6

c) Explain the technique of peephole optimisation with the help of an example. 4

IT 7 – 2 (RC)

B.E. (IT) (Semester – VII) (RC) Examination, November/December 2013
PRINCIPLES OF COMPILERS

Duration: 3 Hours

Total Marks: 100

Instruction : *Assume suitable data* ***wherever*** *required.*

MODULE – 1

1. a) Explain the process of boot strapping and porting with the help of T-diagrams. **8**

   b) Explain the different phases of the compiler with the help of an example. **6**

   c) Consider the following regular expression (a + b) ? (ab)*
      i) Construct the E-NFA for the given regular expression. **3**
      ii) Convert the E-NFA obtained above to NFA. **3**

2. a) For a given LEX program, give the DFA implementation of the lexical analyser for the following regular definitions **8**

   % {W, WXX, WX %}

   %%

   W^+ {yyval = 1, return (w) }

   W^+X^+ {yyval = 2, return (WXX) }

   W^*X {yyval = 3, return (WX) }.

   b) Define the input buffering scheme. Explain with the help of pseudo code the use of the sentinel character in the input buffering. **6**

   c) Write a short note on the following :
      i) Front end and back end phases of compiler. **2**
      ii) Linker and loaders. **2**
      iii) Analysis phases and synthesis phases of compiler. **2**

MODULE – 2

3. a) Consider the following grammar

   E → E or T / T

   T → T and F / F

   F → not F / (F) / true / false

   i) Compute leading and trailing for the given grammar. **3**

   ii) Construct operator precedence parsing table and parse the string not true or false. **7**

b) Write a short note on the following :

i) Left recursion.

ii) Left factoring.

c) Write a short note on recursive descent parsers.

d) How will you check if the given grammar is ambigious or not ?

4. a) Consider the following grammar :

$S \rightarrow AB$

$A \rightarrow AB \mid Ba$

$B \rightarrow (A) \mid b$

i) Construct CR(0) items for the above grammar.

ii) Construct the parsing table and parse the string bab by showing the operations on the stack and input buffer.

b) Write a short note on YACC.

c) How do you check if the given grammar is LL(1) or not ? Explain with the help of an example.

d) What do you mean by a handle and handle pruning ?

MODULE – 3

5. a) Explain the syntax directed translation scheme for the assignment statement with mixed data type.

b) What is an activation tree ? Explain how it is constructed with the help of an example.

c) Write a short note on the following :

i) Lexical errors.

ii) Syntax errors.

d) What do you mean by back patching and how it is been carried out.

6. a) Consider the following statement :

$a * b + d - e * f * g - h/e$

Generate the intermediate code using

i) Syntax tree

ii) Quadruples

iii) Triples

iv) Indirect triples.

b) Explain the most closely nested rule for accessing non local names with the help of an example. 6

c) Explain the different data structures that can be used to create symbol table. 6

MODULE – 4

7. a) Write the algorithm for the construction of the directed acyclic graph and trace it on the given expression 6

A[k] = B[i] * C[j] + D.

b) Explain the code generation algorithm with the help of an example. 6

c) Explain the following with respect to code generation : 6

i) Next use information

ii) Register descriptor

iii) Address descriptor.

d) Explain any one loop optimisation technique. 2

8. a) Consider the following pseudocode

```
        int i = 1;
    While (i ≤ 20)
{
  i = i + A [ i ],
}
  X = A [i ] ;
```

i) Construct the basic block for the following code. 4

ii) Optimise each of the blocks by using different optimisation techniques. 6

b) Explain how we compute the next use information during code optimisation phase. 5

c) Write a short note on the different issues faced in the design of the code generator. 5

IT 7 – 2 (RC)

B.E. (I.T.) (Semester – VII) (RC) Examination, May/June 2013
PRINCIPLES OF COMPILERS

Duration : 3 Hours

Total Marks : 100

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

MODULE – I

1. a) Provide the difference between compiler and interpreter. 4

   b) Show the output of each phase of compilation for the following C statements : 10

```
int x = 0 ;
for (i = 0 ; i < 5 ; i ++)
    {
      x = x + 5.0 ;
    }
```

   c) Describe the role of lexical analyzer. 6

2. a) Give the specification of LEX tool. List the steps involved in compilation of LEX program. 7

   b) Write a short note on bootstrapping process. 6

   c) Describe input buffering technique used for lexical analysis. What is the advantage of using sentinel character ? 7

MODULE – II

3. a) What is ambiguous grammar ? What are the ways to deal with ambiguity ? 5

   b) Eliminate left recursion from the following grammar. 6

   i) $S \rightarrow A$
   $A \rightarrow Ad \mid Ae \mid aB \mid aC$
   $B \rightarrow bBC \mid f$
   $C \rightarrow g$

   ii) $S \rightarrow (L) \mid a$
   $L \rightarrow L, S \mid S$

P.T.O.

c) Construct predictive parsing table for the following grammar and parse the string dbbbf.

$S \rightarrow aP \mid PR$
$P \rightarrow dQ$
$Q \rightarrow bQ \mid \in$
$R \rightarrow f$

4. a) Write a note on YACC parser generator.

b) Compute leading and trailing sets for the following grammar :

$D \rightarrow L : T \mid L \mid T$
$L \rightarrow L, id \mid id$
$T \rightarrow num$

c) Check if the following grammar is SLR(1) :

$S \rightarrow Ta \mid bTc \mid dc \mid bda$
$T \rightarrow d$

MODULE – III

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

add = add + a[i] * b[i].

b) Explain the syntax directed translation of mixed mode boolean expression.

c) Explain source language issues.

6. a) Describe the syntax directed translation for flow of control statements.

b) What are errors ? How they are reported ?

c) Explain dynamic scope and its approaches.

MODULE – IV

7. a) Construct the DAG for the following : 4

A = B – C

D = A + E

E = D * D

B = E * A

C = E – D

b) Explain code generation algorithm. 8

c) Explain following techniques of code optimization : 8

i) Dead code elimination

ii) Common sub expression elimination

iii) Copy propagation

iv) Constant folding.

8. a) Write the final code generated for the statement x = ((a + b) – e – (c + d)). Show the contents of address descriptor and register descriptor at each statement. 8

b) Explain the ways in which machine implements conditional jumps. 4

c) Explain the transformations on Basic block. 8

B.E. IT (Semester – VII) (RC) Examination, Nov./Dec. 2012
PRINCIPLES OF COMPILERS

Duration : 3 Hours | Total Marks : 100

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

MODULE – I

1. a) Why do we need compiler ? How does it differ from an assembler ? 5

   b) Write a short note on boot strapping process. 6

   c) Show that output of each phase of compilation for the following C statements : 9

```
int a=0 ;
if (a < 6)
  {
    a = a + 3.0 ;
  }
```

2. a) Distinguish between Front-end and back-end of compiler. 4

   b) Describe the role of lexical analyzer. Give an example of token, pattern and lexeme. 8

   c) Write a note on LEX tool. Give a Lex program to identify if the input is word or number. 8

MODULE – II

3. a) Eliminate left recursion from the following grammar : 4

   i) $S \rightarrow aP \mid PR$
   $P \rightarrow Pb \mid d$
   $R \rightarrow f$

   ii) $S \rightarrow aAb \mid bB$
   $A \rightarrow Aa \mid \in$
   $B \rightarrow Bb \mid \in$

b) Perform left factoring for the following grammar :

i) $S \rightarrow aAA$

$A \rightarrow aS \mid bS \mid a$

ii) $S \rightarrow PQ$

$P \rightarrow int \mid float$

$Q \rightarrow a, Q \mid a$

c) Provide the model of predictive parser. Check if the given grammar is LL(l). 12

$S \rightarrow ict\ SR \mid a$

$R \rightarrow e\ S \mid \in$

$C \rightarrow b$

4. a) Write a note on Yet Another Compiler Compiler. 6

b) Construct SLR parsing table for the following grammar : 14

$S \rightarrow Ta \mid bTc \mid dc \mid bda$

$T \rightarrow d$

Parse the input string bdc.

MODULE – III

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

sum = sum + a[i] + b[i]

b) Explain syntax directed definition / translation to produce three address code for assignment. 8

c) Write a note on activation trees. 6

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

b) Explain the storage allocation strategies. 6

c) State and explain the errors in following categories : 6

i) Syntatic errors

ii) Semantic errors.

MODULE – IV

7. a) Explain in detail Peephole optimization. 8

b) Construct the DAG for the following : 4

$t_1 = b + c$

$t_2 = d * e$

$t_3 = b + c$

$t_4 = t_2 * t_3$

$t_5 = t_4 * f$

$x = t_1 - t_5$

c) State the algorithm to partition a 3-address statement into basic blocks. Provide an example. 8

8. a) Write a three address code for the fragment given below. Convert it to the basic blocks and draw the flow graph. Further optimize the code. 12

```
count = 0 ;
result = 0 ;
while (count ++ <20)
  {
     increment = 2 * count ;
     result + = increment ;
  }
```

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

B.E. (IT) RC Sem. VII Examination, May/June 2012
PRINCIPLES OF COMPILERS

ation : 3 Hours | Total Marks :100

Instructions : *1) Answer **any 5** questions, atleast **1** from **each** Module.*
*2) **Assume** suitable data.*
*3) Draw diagram **neatly** with pencil if any.*

Module – I

a) Write short notes on : 4

a) Linkers

b) Loaders.

b) Describe the role of Lexical analyzer. 6

c) Show the output of each phase of compilation for the following C statement : 10

```
int i = 0,
While (i < 5)
{
    i = i + 3.0 ;
}
```

a) Give the specification of LEX. 5

b) Write short note on compiler construction tools. 6

c) Consider the following LEX program

```
% { ABB, BA, AB %}
% %
{abb}    {yy|va| = 1 ; return (ABB) ; }
{b+ a*}  {yy|va|=2 ; return (BA) ;}
{a+b+}   {yy|va|=3 ; return (AB) ; }
% %
```

Give the implementation of lexical analyzer using DFA. 9

P.T.O.

Module – II

3. a) Eliminate left recursion from the following grammar

$A \rightarrow Ba|Aa|c$

$B \rightarrow Bb|Ab|d$

b) Consider the following grammar :

$B \rightarrow TB'$

$B' \rightarrow or\ TB' | \in$

$T \rightarrow FT'$

$T' \rightarrow and\ FT' | \in$

$F \rightarrow not\ F|(B)|t \mid f$

perform predictive parsing for input string not (t or f).

c) Write a short note on YACC tool.

4. a) Perform operator precedence parsing for sentence (a, a) given grammar :

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid s$.

b) Given the following :

$S \rightarrow P$

$P \rightarrow * R \mid R$

$R \rightarrow (s) \mid id$

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

b) Draw goto Graph.

Module – III

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

$(a + b) * (c + d) - (a + b + c)$.

b) Explain the contents of the symbol table.

c) State and explain.

i) Syntatic phase errors

ii) Lexical phase errors.

6. a) Explain the translation of Assignment statements. 8

b) Explain with a diagram the structure of an activation record. 6

c) Write a note on Activation tree. 6

Module – IV

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

b) Construct the DAG for the following :

$x = a + b$

$y = a - b$

$w = a + c$ 4

$z = x * y$. 8

c) Explain in detail the peehole optimization.

a) State the algorithm to partition a three address statement into basic block and explain it with an example. 12

b) Write a short note on next use information. 4

c) Explain the ways in which machine implements conditional jumps. 4

B - 885

(Pages : 03)

[Total No. of Questions : 8]

IT 7-2 (RC)

B.E. (IT) (Semester - VII) (RC) Examination, Nov./Dec. - 2011

PRINCIPLES OF COMPILERS

Duration : 3 Hours

Total Marks : 100

Instructions : 1) *Answer any five questions, atleast one from each Module.*

2) *Assume suitable data.*

3) *Draw diagram neatly with pencil if any.*

MODULE - I

1) a) With the help of an example, describe the phases of compiler. [8]

b) Discuss briefly the compiler writing tools. [6]

c) Write short notes on:- [6]

i) Assemblers.

ii) Preprocessors.

iii) Loaders.

2) a) Consider the following LEX program:- [8]

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

Give the implementation of lexical analyzer using DFA.

b) Differentiate between [4]

i) Analysis phase and synthesis phase.

ii) Front end and Back end.

c) Write a short note on LEX tool. [8]

P.T.O.

MODULE - II

Q3) a) Eliminate left recursion in the grammer.

$S \rightarrow A\,a \mid b$

$A \rightarrow A\,C \mid S\,d \mid E$

[3]

b) Perform predictive parsing for input string not (t or f).

Given grammer :- $B \rightarrow B$ or $T \mid T$

$T \rightarrow T$ and $F \mid F$

$F \rightarrow$ not $F \mid (B) \mid t \mid f$

[12]

c) Write a short note on YACC parser generator.

[5]

Q4) a) Consider the following grammer,

$S \rightarrow (L)^{*} \mid a$

$L \rightarrow L, S \mid S$

Perform operator precedence parsing for sentence (a, a).

[10]

b) Construct the SLR parsing table for the following grammer:-

$S \rightarrow P$

$P \rightarrow * R \mid R$

$R \rightarrow (s) \mid i\,d$

Parse the string $*$ (i d)

[10]

MODULE - III

Q5) a) Convert the following code to Quadruples, Triples, and Indirect triples.

a [i] = sum + b [i] * c [i]

[6]

b) Explain the translation scheme using a numerical representation for booleans. [6]

c) Explain source language issues.

[8]

Q6) a) Explain error recovery strategies.

[8]

b) Explain the translation to produce three - address code for boolean expressions. [8]

c) Write a short note on Back patching.

[4]

MODULE - IV

a) State and explain simple code generator algorithm. [8]

b) Write three - address code for the following fragment. Convert it to basic blocks and draw the flow graph. Further optimize the code. [12]

```
prod : = 0
i : = 1
while (i <= n)
prod = prod * a [ i ]
i = i + 1
```

a) Write an algorithm to construct DAG representation. [8]

b) Explain the following techniques of code optimization. [8]

i) Dead code elimination.

ii) Common subexpression elimination.

iii) Copy propagation.

iv) Constant folding.

c) Construct a DAG for the following code. [4]

```
a : = b + c
d : = b – c
x : = a + d
y : = a – d
```

❄❄❄

6 - 978 (Pages : 03) IT 7-2(RC)

Total No. of Questions : 8]

B.E. (IT) (Semester - VII) Examination, May 2011

PRINCIPLES OF COMPILERS

uration : 3 Hours Total Marks : 100

structions : 1) *Solve any five questions by selecting atleast 1 from each module.*

2) *Answer to the subquestions of a question should be written in continuation to each other.*

3) *Draw figures and diagrams wherever necessary.*

MODULE - I

) a) Explain the compiler construction tools. [8]

b) Differentiate between the following : [6]

(i) Analysis phase and Synthesis phase.

(ii) Source program and Target program.

c) Explain the following parameter passing techniques. [6]

(i) Call-by-value.

(ii) Call-by-reference.

(iii) Call-by-name.

2) a) Show the output of each phase of compilation for the following C code. [8]

```
float a;
while (a < 10)
{
    a = a + 1;
}
```

b) State the use of sentinel character in input buffering. How does it improve the efficiency of the code? [6]

c) Explain in detail the role of a lexical analyzer. [6]

MODULE - II

3) a) Eliminate left factoring and construct predictive parsing table for the following grammar [12]

$S \rightarrow aAA$

$A \rightarrow aS/bS/a$

Parse the string aabaaa.

P.T.O.

b) Eliminate left recursion from the grammar given below.

(i) S → Ab/a
A → SAa/b

(ii) S → AS/b
A → SA/a

c) Write a short note on YACC tool.

Q4) a) Construct SLR parsing table for the following grammar.

S → AB

A → AB/Ba

B → (A)/b

b) Construct operator precedence parser for the following grammar.

E → E+T/T

T → T * F/F

F → (E)/id

MODULE - III

Q5) a) Convert the following code to quadruples, triples and indirect triples.

a[i] = b[i] * c[i]

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

c) Explain the methods of translating boolean expressions to three-address code.

Q6) a) Explain the contents of the symbol table.

b) State and explain the errors in the following categories

(i) Syntactic errors

(ii) Semantic errors.

c) Explain the storage allocation stratergies.

MODULE - IV

Q7) a) Explain the code generation algorithm.

b) Construct DAG for the following code

D = B * C

E = A + B

B = B * C

A = E – D

c) Explain the following with examples. [8]

(i) Local optimization

(ii) Loop optimization.

98) a) Write three-address code for the following fragment. Convert it to basic blocks and draw the flow graph. Further optimize the code. [12]

```
Sum = 0
for (i = 1; i < = n; i++)
{
    Sum = Sum + a[i];
}
```

b) Explain the following issues in the design of code generator [8]

(i) Instruction Costs.

(ii) Choice of evaluation order.

❑❑❑

IT 7 – 2 (RC)

B.E. (IT) Semester – VII (RC) Examination, Nov./Dec. 2010
PRINCIPLES OF COMPILERS

ration : 3 Hours

Total Marks : 100

Instructions : *1) Answer **any five** questions choosing at least **one** from **each** Module.*

*2) **Assume** suitable data, **if necessary**.*

MODULE – I

a) Write short notes on the following : 8

i) Assemblers

ii) Loader

iii) Linker

iv) Preprocessor.

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

c) Differentiate between front end and back end. 4

a) Show the output of each phase of compilation for the following statements. 8

```
if (a < b)
  a = b + 2;
else
    b = a + 3;
```

b) Explain the following terms with examples. 6

i) Lexems

ii) Patterns

iii) Tokens.

c) Explain the specifications of a LEX tool. 6

MODULE – II

a) Construct predictive parsing table for the following grammar and parse the string int id_1, id_2 12

$D \rightarrow TV$

$T \rightarrow int \mid float$

$V \rightarrow id, V \mid id$

P.T.O.

b) Left factorize the grammar

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

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

$C \rightarrow c \mid \varepsilon$

c) Eliminate left recursion from the following grammar.

$S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \varepsilon$

4. a) Construct canonical LR parsing table for the following grammar

$S \rightarrow E = E$

$E \rightarrow E + T \mid T$

$T \rightarrow T * id \mid id$

b) Construct SLR parsing table for the following grammar

$S \rightarrow E \#$

$E \rightarrow T \mid E + T \mid E - T$

$T \rightarrow i \mid (E)$

MODULE – III

5. a) Convert the following code to Quadruples, Triples and Indirect Triples.

a [i] = sum + b [i] * c[i]

b) Explain the terms static scope and dynamic scope.

c) Explain the translation of case statements.

6. a) List and explain the operations on symbol table.

b) Explain the translation scheme using a numerical representation for booleans.

c) Write a note an activation trees.

MODULE – IV

a) Explain the following terms : 6

i) Register Descriptor

ii) Address Descriptor.

b) Write a note on next-use information. 4

c) Construct DAG for the following code 4

$t_1 = a + b$

$x = t_1$

$t_2 = a - b$

$y = t_2$

$z = x + y$.

d) Discuss the following techniques of code optimization with examples 6

i) Induction variables

ii) Dead code Elimination.

a) Write a three-address code for the fragment given below. Convert it to the basic blocks and draw the flow graph. Further optimize the code. 12

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

b) Explain in detail Peephole optimization. 8

Total No. of Printed Pages:2

T.E. (Information Technology) Semester - VI (Revised Course 2019-20)
EXAMINATION JULY 2022
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

Q.1
a) Define lexeme, token, pattern. Provide an example. **3**
b) Show the output of each phase of compilation for the following C statements: - **10**

```
int x, y;
if(x<y)
        x= y + 5.0;
else
        y = x + 2.0;
```

c) Describe various Compiler writing tools. **7**

Q.2
a) a) What is ambiguous grammar? What are the ways to deal with this grammar? **3**
b) Construct Predictive parsing table the following grammar: **10**

B -> B or T | T
T -> T and F | F
F-> not F | (S) | t| f
Check whether the grammar is LL (1) or not **7**

c) Describe the specification and compilation process of LEX tool.

Q.3
a) Construct SLR parsing table for the following grammar: **10**

S -> AB
A -> AB | Ba
B -> (A) | b

b) Construct operator precedence parsing table for the following grammar and parse the string (a , ^) **10**

S-> a | ^ | (T)
T->T, S | S

Part -B

Q.4
a) Convert the following code to quadruples, triples and indirect triples while(a<c)
{
c= c* a + 1;
}
b) Explain the syntax directed translation for flow of control statements.
c) Illustrate the structure of an activation record with diagram.

Q.5
a) Construct a DAG for the following code:-
a = b * c
a = b
e = d * c
b = e
f = b + c
g = f + d
b) Provide the translation scheme for assignment statement
c) Describe various storage allocation strategies.

Q.6
a) Describe the various issues in the design of code generator.
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.

```
v=0;
i:=1;
do begin
      v;=v+a[i]*b[i];
       i:=i+1
end
while(i<=20)
 end
```

c) Define Address descriptor and Register descriptor.

PART-C

Q.7
a) Explain input buffering technique used for lexical analysis phase of compilation.
b) Differentiate between:-
i) Front-end and Back-end of compiler.
ii) Analysis phase and Synthesis phase.
iii) Linkers and Loaders.
c) Briefly describe the various data structures used for Symbol Table.

Q.8
a) Write an algorithm to construct basic blocks from a sequence of 3-address statements.
b) Explain the following code optimization techniques.
i) Common sub-expression elimination
ii) Constant folding
c) Explain in detail Peephole optimization.

5E568B2043F732C5994F834619E98794

Total No. of Printed Pages:2

T.E. (Information Technology) Semester - VI (Revised Course 2019-20)
EXAMINATION JULY 2022
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

Q.1
a) Define lexeme, token, pattern. Provide an example. **3**
b) Show the output of each phase of compilation for the following C statements: - **10**

```
int x, y;
if(x<y)
        x= y + 5.0;
else
        y = x + 2.0;
```

c) Describe various Compiler writing tools. **7**

Q.2
a) a) What is ambiguous grammar? What are the ways to deal with this grammar? **3**
b) Construct Predictive parsing table the following grammar: **10**

B -> B or T | T
T -> T and F | F
F-> not F | (S) | t| f
Check whether the grammar is LL (1) or not **7**

c) Describe the specification and compilation process of LEX tool. **10**

Q.3
a) Construct SLR parsing table for the following grammar:

S -> AB
A -> AB | Ba
B -> (A) | b **10**

b) Construct operator precedence parsing table for the following grammar and parse the string (a , ^)

S-> a | ^ | (T)
T->T, S | S

Part -B

Q.4

a) Convert the following code to quadruples, triples and indirect triples while(a<c)

```
{
c= c* a + 1;
}
```

b) Explain the syntax directed translation for flow of control statements.

c) Illustrate the structure of an activation record with diagram.

Q.5

a) Construct a DAG for the following code:-

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

b) Provide the translation scheme for assignment statement

c) Describe various storage allocation strategies.

Q.6

a) Describe the various issues in the design of code generator.

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.

```
v=0;
i:=1;
do begin
    v;=v+a[i]*b[i];
    i:=i+1
end
while(i<=20)
end
```

c) Define Address descriptor and Register descriptor.

PART-C

Q.7

a) Explain input buffering technique used for lexical analysis phase of compilation.

b) Differentiate between:-
   i) Front-end and Back-end of compiler.
   ii) Analysis phase and Synthesis phase.
   iii) Linkers and Loaders.

c) Briefly describe the various data structures used for Symbol Table.

Q.8

a) Write an algorithm to construct basic blocks from a sequence of 3-address statements.

b) Explain the following code optimization techniques.
   i) Common sub-expression elimination
   ii) Constant folding

c) Explain in detail Peephole optimization.

5E568B2043F732C5994F834619E98794

TE1273

Total No. of Printed Pages:2

T.E. (Information Technology) Semester - VI (Revised Course 2019-20)
EXAMINATION JULY 2022
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

Q.1
a) Define lexeme, token, pattern. Provide an example. 3
b) Show the output of each phase of compilation for the following C statements: - 10

```
int x, y;
if(x<y)
        x= y + 5.0;
else
        y = x + 2.0;
```

c) Describe various Compiler writing tools. 7

Q.2
a) a) What is ambiguous grammar? What are the ways to deal with this grammar? 3
b) Construct Predictive parsing table the following grammar: 10

B -> B or T | T
T -> T and F | F
F-> not F | (S) | t| f
Check whether the grammar is LL (1) or not 7

c) Describe the specification and compilation process of LEX tool. 10

Q.3
a) Construct SLR parsing table for the following grammar:

S -> AB
A -> AB | Ba
B -> (A) | b 10

b) Construct operator precedence parsing table for the following grammar and parse the string (a , ^)

S-> a | ^ | (T)
T->T, S | S

Part -B

Q.4

a) Convert the following code to quadruples, triples and indirect triples while(a<c)

```
{
c= c* a + 1;
}
```

b) Explain the syntax directed translation for flow of control statements.

c) Illustrate the structure of an activation record with diagram.

Q.5

a) Construct a DAG for the following code:-

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

b) Provide the translation scheme for assignment statement

c) Describe various storage allocation strategies.

Q.6

a) Describe the various issues in the design of code generator.

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.

```
v=0;
i:=1;
do begin
    v;=v+a[i]*b[i];
    i:=i+1
end
while(i<=20)
end
```

c) Define Address descriptor and Register descriptor.

PART-C

Q.7

a) Explain input buffering technique used for lexical analysis phase of compilation.

b) Differentiate between:-
   i) Front-end and Back-end of compiler.
   ii) Analysis phase and Synthesis phase.
   iii) Linkers and Loaders.

c) Briefly describe the various data structures used for Symbol Table.

Q.8

a) Write an algorithm to construct basic blocks from a sequence of 3-address statements.

b) Explain the following code optimization techniques.
   i) Common sub-expression elimination
   ii) Constant folding

c) Explain in detail Peephole optimization.

IT 6-5 (NC)

T.E. (IT) (Semester – VI) (NC) Examination, Nov./Dec. 2013
PRINCIPLES OF COMPILERS

Duration : 3 Hours Total Marks : 100

Instructions : *1) Answer **any five** questions with at least **one** from **each** Module.*

*2) Make **necessary** assumptions.*

MODULE – I

1. a) Explain the process of porting and boot strapping with help of T-diagrams. 8

   b) Compute the following set of C-statement by clearly showing the i/p and o/p of each phase of compilation 12

```
int i = 0;
do
{ a[i] = 0.0;
  i = i + 1;
} while (i < 10);
```

2. a) Differentiate between the following terms : 9

   i) Analysis phase and synthesis phase

   ii) Front end and back end

   iii) Single pass compiler and multipass compiler.

   b) What do you mean by an interpreter ? State some examples. 4

   c) Differentiate between static scoping and dynamic scoping. 4

   d) Explain call-by-reference parameter passing method. 3

P.T.O.

MODULE – II

3. a) Consider the following regular expression

   (a/b)* a(a/b)

   1) Construct a $\in$ – NFA for the above r.e.

   2) Convert $\in$ – NFA to DFA

   3) Minimize the number of states if possible.

   b) Build a DFA which accepts all the words cover {a, b} that have even numbe of a's and b's.

   c) Define i/p buffering. Where it is used in compilation ?

4. a) Define the following terms with an example : token, pattern, lexeme.

   b) Give LEX specification for the table :

token	code	value
begin	1	–
real	2	–
int	3	–
end	4	–
identifier	5	ptr to ST
constant	6	ptr to ST
<	7	1
>	7	2
< >	7	3

   c) What are advantages of using regular expression in lexical analysis ?

MODULE – III

5. a) Construct a predictive passing table for the grammar given below and parse the string int x, y, z.

   $D \rightarrow TV$

   $T \rightarrow$ int/float

   $V \rightarrow$ id, V|id.

b) Construct an operator precedence table for the grammar given below 8

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow - F \mid (F) \mid 0 \mid 1.$

6. a) Construct the SLR passing table for the grammar given below, and parse the string bab using the table 12

$S \rightarrow AB$

$A \rightarrow AB \mid Ba$

$B \rightarrow (A) \mid b.$

b) What are the difficulties faced by an back tracking passer ? 4

c) Define handle. What do you mean by handle primary ? 4

MODULE – IV

7. a) State the code generation algorithm and explain it with an example. 10

b) Discuss the following techniques used in code optimization 6

i) loop optimization

ii) Flow of control optimization.

c) Write short note on use of self organizing bit over linear bit in a symbol table organization. 4

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

B[I] = C[J] * D[K] + sum.

b) State the syntax directed translation scheme for Boolean statements using appropriate grammar. 8

c) Construct DAG for the following code. 4

$t_1 = a + b$

$x = t_1$

$t_2 = a - b$

$y = \sqrt{2}$

$z = x + y$

d) State some syntactic errors. 2