Compiler Design
VTU Question
ISE
Paper Set
2017

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, Dec.2016/Jan.2017
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Differentiate between computer and Interpreter. List and explain the various phases of a compiler and show the output of each phase for the expression a : = b + c * 25. **(10 Marks)**

b. Explain the technique of input buffering used by the lexical analyser. **(06 Marks)**

c. Construct transition diagram for the following :
   i) Unsigned numbers
   ii) Relational operators. **(04 Marks)**

2 a. Write the algorithm used of eliminating left recursion. Use this algorithm and eliminate left recursion on the given grammar
$S \rightarrow C \mid a$
$C \rightarrow Dd \mid c$
$D \rightarrow Cc \mid d$ **(07 Marks)**

b. Write the "dangling else" grammar and show that it is ambiguous. Rewrite the grammar incorporating the rule "Match each else with the closest unmathed then". **(08 Marks)**

c. List and explain the various error recovery strategies. **(05 Marks)**

3 a. Construct the predictive parsing table for the given grammar
$E \rightarrow TE'$
$E' \rightarrow +E \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow T \mid \varepsilon$
$F \rightarrow PF'$
$F' \rightarrow *F' \mid \varepsilon$
$P \rightarrow (E) \mid a \mid b \mid ep$ **(10 Marks)**

b. Discuss the conflicts that can arise during shift reduce parsing giving one example for each type. Do shift reduce parse for the string (a, (a, a)) and indicate the presence of conflicts if any. Use the given grammar
$S \rightarrow (L) \mid a$
$L \rightarrow L, S \mid S$ **(10 Marks)**

4 a. Show that the following grammar is LR(1).
$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$
$A \rightarrow d$
$B \rightarrow d$ **(10 Marks)**

b. Explain the procedure for the construction of an LA LR parser. **(10 Marks)**

PART – B

5 a. Define :
   i) Inherited attribute
   ii) Synthesized attribute. **(02 Marks)**

b. Develop the grammar and SDD for a simple desk calculator and show the annotated parse tree for the expression 1 * 2 * 3* (4+5) n. **(10 Marks)**

c. Rewrite the actions of desk calculator SDD so that they manipulate the parser stack explicitly. Illustrate the parser stack implementation. **(08 Marks)**

6 a. Construct a DAG for the arithmetic expression 2* x+y * (2*x – y). Show the steps for constructing the DAG. **(08 Marks)**

b. Generate intermediate code for the statement "if (x < 100 || x > 200 && xi = y) x = 0", along with the required syntax directed translation scheme. Avoid redundant Gotos. **(12 Marks)**

7 a. Show the structure of activation record. Explain the purpose of each item on the activation record. **(08 Marks)**

b. Explain the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. List and explain the performance metrics to be considered when designing a garbage collector. **(04 Marks)**

8 a. Generate the code for the expression
   X = (a – b) + (a + c) **(04 Marks)**

b. What are basic blocks? Explain the DAG representation of basic blocks. **(08 Marks)**

c. Describe the next use information. Write an algorithm to determine the liveness and next use information for each statement in a basic block. **(08 Marks)**

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, June/July 2016
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain various phases of compiler. **(10 Marks)**

b. Construct a transition diagram for recognizing relational operators. Sketch the program segment to implement it, showing the first state and one in final state. **(10 Marks)**

2 a. Write an algorithm to eliminate left recursion from a grammar. Eliminate left recursion from the grammar: $S \rightarrow Aa \mid b,\ A \rightarrow Ac \mid sd \mid a$. **(08 Marks)**

b. Show that the following grammar is ambiguous:

Stmt → if expr then stmt
| if expr then stmt else stmt
| other

write an unambiguous grammar for the same. **(06 Marks)**

c. Give the rules for constructing FIRST and FOLLOW sets. **(06 Marks)**

3 a. What is meant by handle pruning? How it helps in shift reduce parsing? List the actions of a shift reduce parser. **(10 Marks)**

b. For the grammar: $S \rightarrow SS+ \mid SS* \mid a$. Give a bottom-up parse for the input: aaa * a++. **(06 Marks)**

c. What are two types of conflicts during shift-reduce-parsing? Give examples. **(04 Marks)**

4 a. What is the meaning of 'L' and 'R' in LR grammars? Why LR parsing is attractive? **(04 Marks)**

b. Construct canonical LR(1) items for the augmented grammar: $s' \rightarrow s;\ s \rightarrow Cc;\ c \rightarrow cC \mid d$. **(10 Marks)**

c. Write the yacc specification of a simple desk calculator with the following grammar for arithmetic expressions:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$. **(06 Marks)**

PART – B

5 a. Define the following with examples:
i) Synthesized attribute
ii) Inherited attribute
iii) S-Attributed definitions
iv) L-Attributed definitions. **(08 Marks)**

b. Explain the parser stack implementation of postfix STD with an example. **(08 Marks)**

c. Define syntax directed definition for a simple type declaration. **(04 Marks)**

6 a. List any four common three address instruction forms. **(04 Marks)**
b. Discuss quadruples, triples and indirect triples representation. **(06 Marks)**
c. Write syntax directed definition for flow of control statements. **(10 Marks)**

7 a. Explain the desirable properties of memory manager. **(06 Marks)**
b. Explain in detail, the strategy for reducing fragmentation in heap memory. **(08 Marks)**
c. Explain the design goals for garbage collector. **(06 Marks)**

8 a. Discuss the issues in the design of code generator. **(10 Marks)**
b. Write intermediate code for the following source code

```
for i from 1 to 10 do
for j from 1 to 10 do
    a [i, j] = 0.0;
for i from 1 to 10 do
    a [i, i] = 1.0.
```

(10 Marks)

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, Dec.2015/Jan.2016
Complier Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain the various phases of compiler with example. **(10 Marks)**

b. Explain the input buffering strategy used in lexical analysis phase. **(05 Marks)**

c. Write the regular definition for an unsigned number, also write the transition diagram. **(05 Marks)**

2 a. Define left recursion and left factoring? Consider the Grammar :

$E \rightarrow E + T \mid T$

$T \rightarrow id \mid id [\,] \mid id [X]$

$X \rightarrow E, E \mid E$

i) Eliminate left recursion

ii) For the obtained result of i) do the left factoring. **(10 Marks)**

b. Construct LL(1) parsing table for the grammar

$S \rightarrow aB \mid aC \mid Sd \mid Se$

$B \rightarrow bBc \mid f$

$C \rightarrow g$

and verify the above grammar is LL (1) or not. **(10 Marks)**

3 a. Define Handle and Handle pruning. Consider the grammar :

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

Indicate the handle for the following right sentential form $id_1 * id_2$. **(06 Marks)**

b. With a neat diagram, explain the general structure of LR parser. **(06 Marks)**

c. Consider the grammar :

$S \rightarrow L = R \mid R$

$L \rightarrow *R \mid id$

$R \rightarrow L$

Verify the grammar is SLR (1) or not. **(08 Marks)**

4 a. Given the grammar :

$A \rightarrow (A) \mid a$

Construct LR(1) set of items, parsing table and also parse the input string ((a)) using canonical LR parsing method. **(12 Marks)**

b. Write the procedure used to compute LR(1) items using LALR parser. **(04 Marks)**

c. Write a note on the parser generator - Yacc. **(04 Marks)**

PART – B

5 a. Define synthesized and inherited attributes with examples. **(04 Marks)**

b. Consider the grammar that is used for simple desk calculator. Obtain the semantic action and also the annotated parse tree for the string. (3 + 4) * (5 + 6)n

$L \rightarrow En$
$E \rightarrow E + T$
$E \rightarrow T$
$T \rightarrow T * F$
$T \rightarrow F$
$F \rightarrow (E)$
$F \rightarrow$ digit. **(10 Marks)**

c. Consider the grammar :

$T \rightarrow F\ T^1$
$T^1 \rightarrow * F\ T^1$
$T^1 \rightarrow E$
$F \rightarrow$ digit

Write the semantic action and obtain the dependency graph and the order of execution for the input string 3 * 5. **(06 Marks)**

6 a. Define DAG? Develop SDD to produce DAG for the expression :

$E \rightarrow E + T$
$E \rightarrow E - T$
$E \rightarrow T$
$T \rightarrow (E)$
$T \rightarrow$ id
$T \rightarrow$ num

and show the steps for constructing DAG for the expression $a + a * (b - c) + (b - c) * c$. **(12 Marks)**

b. Explain the quadruples, triples and indirect triples for the example a : = b * – c + b * c. **(08 Marks)**

7 a. Describe the general structure of a activation record. Explain the purpose of each field in the activation record and construct activation tree for Quicksort. **(10 Marks)**

b. Explain Heap management in detail. **(10 Marks)**

8 a. Briefly explain the main issues in code generation. **(10 Marks)**

b. Briefly explain any five kinds of code optimization with an example each. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, June/July 2015
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. Explain the various phases of compiler. Show the translations for an assignment statement position = initial + rate * 60, clearly indicate the output of each phase. **(12 Marks)**

b. Construct transition diagram for the following: i) Relational operators; ii) Unsigned number. **(08 Marks)**

2 a. Show that the following grammar is ambiguous: E → E + E / E * E / (E) / id. Write an equivalent unambiguous grammar for the same. **(06 Marks)**

b. Write a recursive descent parser for the grammar: S → cAd A → ab/a and for the input "cad" trace the parser. **(04 Marks)**

c. Consider the grammar:
E → 5 + T / 3 – T
T → V / V * V / V + V
V → a / b
i) Do the left factoring for the above grammar.
ii) Obtain FIRST and FOLLOW table for the above grammar.
iii) Construct predictive parsing table for the above grammar. **(10 Marks)**

3 a. What is handle pruning? Explain with the help of the grammar. S → SS + / SS * / a and input string aaa * a+, give a bottom-up parse of the given input string. **(10 Marks)**

b. For the following grammar S → 0S1 / 01 indicate the handle in the following right sentential form 000011 1. **(04 Marks)**

c. Show that the following grammar is not LL(1) without constructing parsing table:
S → iCtSS' / a
S' → eS / ∈
C → b **(06 Marks)**

4 a. Consider the following grammar
S → CC
C → cC
C → d
i) Obtain canonical collection of LR (0) items.
ii) Construct SLR (1) parsing table.
iii) Show the sequence of moves made by the parser for the string ccdd. **(12 Marks)**

b. Consider the following augmented grammar
S' → S
S → AA
A → Aa/b
Obtain LR(1) items. **(08 Marks)**

PART – B

5 a. Obtain SDD for simple type declaration. Construct a dependency graph for the declaration int a, b, c along with evaluation order. **(08 Marks)**

b. For the given productions shown below, write semantic rules and construct annotated parse tree for

3 * 5 + 4n

$L \rightarrow En$ $E \rightarrow E1 + T$ $E \rightarrow T$

$T \rightarrow T1 * F$ $T \rightarrow F$ $F \rightarrow (E)$ $F \rightarrow$ digit. **(08 Marks)**

c. Define S-attributed and L-attributed definitions with examples. **(04 Marks)**

6 a. Explain how DAG will help in intermediate code generation? Construct a DAG and a 3-address code for the expression a + a * (b – c) + (b – c) * d. **(08 Marks)**

b. Explain the following with an example:

i) Quadruples ii) Triples iii) Indirect triples. **(06 Marks)**

c. Explain syntax directed translation of switch statement. **(06 Marks)**

7 a. Describe the general structure of an activation record. Explain the purpose of each item in the activation record. **(10 Marks)**

b. What is garbage collection? Explain the design goals of garbage collector. **(10 Marks)**

8 a. Briefly discuss the various issues in code generation phase. **(10 Marks)**

b. Explain the following code optimization with examples:

i) Finding local common sub expression.

ii) Dead code elimination. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, Dec.2014/Jan.2015

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Describe the various phases of a compiler. Write down the translation for an assignment statement like: position = initial + rate * 60. Indicate the outputs of each phase. **(12 Marks)**

b. Construct the transition diagrams for a set of keywords like begin, end, if, then and else, and identifiers and constants along with a minimum set of relational operators. **(08 Marks)**

2 a. How left recursion can be eliminated from grammars? Write down the simple arithmetic expression grammar and rewrite the grammar after removing left recursion. **(05 Marks)**

b. What is left factoring? Rewrite the following grammar after "left factored":

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$ **(05 Marks)**

c. Broadly classify parsers. How top-down parsing and bottom-up parsing be subdivided? Briefly explain the merits of each. **(05 Marks)**

d. Briefly explain the implementation of a table driven predictive parser. **(05 Marks)**

3 a. Define LL(1) grammars. Test whether the following grammar is LL(1) or not, and construct a predictive parsing table for it.

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \in$

$B \rightarrow \in$ **(10 Marks)**

b. Construct the LR(0) item set for the grammar:

$E \rightarrow E + T / T$

$T \rightarrow T * F / F$

$F \rightarrow (E) \mid id$

Also draw the DFA for the et of LR(0) items found. **(10 Marks)**

4 a. Construct the canonical LR(1) items and the GOTO graph as well as the canonical LR(1) parsing table for the following augumented grammar:

$S' \rightarrow s$

$S \rightarrow CC$

$C \rightarrow cC \mid d$ **(16 Marks)**

b. For the table in question 4(a), construct the LALR parsing table. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax-directed definitions. What is synthesized attributes and inherited attributes? **(06 Marks)**

b. Construct the syntax tree, parse tree, and the annotated parse tree for the input string say 5 * 6 + 7; using suitable context-free grammar. **(06 Marks)**

c. Construct a dependency graph for the declaration float id1, id2, id3. **(08 Marks)**

6 a. Draw the syntax tree and DAG for the expression (a * b) + (c – d) * (a * b) + b. **(08 Marks)**

b. Represent the following assignment namely a = b * – c + b * – c; in its syntax tree form, three-address code, quadruples and triples representation. **(12 Marks)**

7 a. Discuss the general structure of activation record. **(08 Marks)**

b. What is meant by calling sequence and return sequence? List the calling sequence design principles. **(08 Marks)**

c. Write a note on garbage collection. **(04 Marks)**

8 a. List and briefly explain the design issues of a code generator. **(10 Marks)**

b. With example explain common subexpression and dead code elimination methods. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, June/July 2014
Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Draw diagram for language processing system. **(04 Marks)**

b. Explain general structure of a compiler. **(08 Marks)**

c. Write an algorithm for "look ahead code with sentinels". **(04 Marks)**

d. Draw a transition diagram for identifiers and keywords. How do you handle reserve words that look like identifiers? **(04 Marks)**

2 a. For the following grammar

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

Make necessary changes to make it suitable for LL(1) parser. **(02 Marks)**

b. After doing necessary changes in Q2(a) grammar check whether it is LL(1) grammar or not. If yes, parse the string (a, a). **(10 Marks)**

c. List all error recovery methods in LL(1) parser. Explain one method suitable for the grammar given below.

$S \rightarrow AbS \mid e \mid \in$

$A \rightarrow a \mid cAd$ string is "ceadb" **(08 Marks)**

3 a. What is a shift reduce parser? Explain the conflict that may occur during shift reduce parsing. **(04 Marks)**

b. What is handle pruning? Explain with the help of the grammar $S \rightarrow SS + \mid SS* \mid a$ and input string aaa*a++. **(08 Marks)**

c. Give Bottom-up parsing for the strings 000111 and grammar $S \rightarrow OS1 \mid 01$ and construct parse tree in each step of deviation. **(08 Marks)**

4 a. Write algorithm for construction of canonical LR(1) parsing table. **(10 Marks)**

b. Construct LALR parsing tables for the grammar shown below using LR(1) items.

$S \rightarrow CC$

$C \rightarrow c\,C \mid d$ **(10 Marks)**

PART – B

5 a. Write a SDD for desktop calculator. **(04 Marks)**

b. Assume suitable SDD to construct a syntax tree for the expression a – 4 + c and what are the steps involved in construction of that syntax tree. **(08 Marks)**

c. Construct annotated parse tree for 3*5 and write dependency graph for the constructed parse tree. **(08 Marks)**

6 a. List various 3 address instruction forms. Give one example for each. **(10 Marks)**

b. Construct DAG for the expression

$(x + y) - ((x + y) * (x - y))) + ((x + y) * (x - y))$ **(06 Marks)**

c. Write case 3 address code instructions used to translate a switch statement. **(04 Marks)**

7 a. Discuss about the various components and their use in an activation record. **(08 Marks)**

b. What do you mean by calling sequence? Explain the actions performed during
(i) function call (ii) return. **(08 Marks)**

c. Draw subdivision of run-time memory into code and data areas. **(04 Marks)**

8 a. For the following program segment generate intermediate code and flowgraph of that code.

```
for i from 1 to 10 do
    for j from 1 to 10 do
        a[i, j] = 0.0 ;
    for i from 1 to 10 do
        a[i, i] = 1.0 ;
```

(10 Marks)

b. What are the steps involved in optimization of Basic blocks. Explain any 2 steps. **(10 Marks)**

* * * * *

USN | | | | | | | | | | |

10IS662

Sixth Semester B.E. Degree Examination, June/July 2013

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. Explain, with a neat diagram, the phases of a compiler. **(10 Marks)**

b. Construct the transition diagram to recognize the tokens given below:
i) Identifier
ii) Relational operator
iii) Unsigned number **(10 Marks)**

2 a. What is left-recursion? Eliminate left recursion from the following grammar:
$E \rightarrow E + T/T$
$T \rightarrow T * F/F$
$F \rightarrow (E)/id$ **(08 Marks)**

b. Given the grammar:
$S \rightarrow (L)/Q$
$L \rightarrow L, S/S$
i) Make necessary changes to make it suitable for LL (1) parsing.
ii) Construct FIRST and Follow sets
iii) Construct the predictive parsing table
iv) Show the moves made by the predictive parser on the input (a, (a, a)). **(12 Marks)**

3 a. What is shift reduce parser? Explain the conflicts that may occur during shift reduce parsing. **(04 Marks)**

b. Given the grammar:
$A \rightarrow (A)/a$
i) Find LR (0) items
ii) Construct SLR parsing table.
iii) Write SLR parsing algorithm.
iv) Show the parsing of input string ((a)). **(16 Marks)**

4 a. Given the following grammar:
$S \rightarrow CC$
$C \rightarrow cC/d$
i) Construct sets of LR (1) items.
ii) Construct canonical LR (1) parsing table. **(12 Marks)**

b. Construct LALR parsing tables for the grammar shown in Q.No.4(a) using LR (1) items. **(08 Marks)**

PART – B

5 a. Explain the concept of syntax directed translation, with examples. **(06 Marks)**

b. Define inherited and synthesized attributes. **(04 Marks)**

c. Give SDD of a simple desk calculator. **(04 Marks)**

d. Write the annotated parse tree for 3*5 + 4n. **(06 Marks)**

6 a. Draw the DAG for the arithmetic expression, a + a * (b – c) + (b – c) * d. Show the steps for constructing the DAG. **(10 Marks)**

b. Translate the arithmetic expression a + –(b + c) into quadruples, triples and indirect triples. **(06 Marks)**

c. Write the tree address code for switch statement. **(04 Marks)**

7 a. What is an activation record? Explain the purpose of each item in the activation record, with an example. **(08 Marks)**

b. What is meant by calling sequence and return sequence? List calling sequence design principles. **(08 Marks)**

c. Write a note on Garbage collection. **(04 Marks)**

8 a. List and explain design issues of a code generator. **(10 Marks)**

b. With an example, explain common sub-expression and dead code elimination methods. **(10 Marks)**

* * * * *