SASTRA
DEEMED TO BE UNIVERSITY

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes Max Marks: 50

PART – A

ANSWER ALL THE QUESTIONS 10 X 2 = 20 Marks

1. A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2 and T_3 over the alphabet {a, b, c}.
   T_1: a?(b|c)*a
   T_2: b?(a|c)*b
   T_3: c?(b|a)*c
   If the string "baabbccccab" is processes by the analyzer, what will be the sequence of tokens it produces?
2. Consider the following segment of python program,
```
# float n
numbers = [11, 12, 13]
for n in numbers:
    print("{n} is a number")
```
   Specify the tokens recognized by the Lexical Analyser during scanning.
3. Given the Expression grammar of a recent high-level programming language,
   E -> E * F | F + E | F
   F -> F - F | id
   Find the precedence of operators with respect to the Grammar.
4. Consider the grammar
   E → E + n | E * n | n.
   For a sentence **n + n * n**, identify the "handles" in the right-sentential form of the reduction technique?
5. Build the derivation tree for the string **((a, a), a)** based on the following Context Free grammar
   S → (L) | a
   L → L , S | S
6. Illustrate the process of eliminating the Left Recursion from the given grammar:
   A → ABb | Bb |a
   B → Bc | Acc |b

SASTRA
DEEMED TO BE UNIVERSITY

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**

7. Write the **LEX specification** to recognize the string w described as:
   - w begins and ends with digit. |w| between 8 and 12 characters.
   - w contains atleast one Upper case letter and a special character (either # or @) in the middle.
8. Augment a Context Free Grammar to represent the syntactical correctness of **while and do-while statement in C++.**
9. Compare the behaviour of the Compiler and Interpreter upon detecting any error in the Source program during the translation.
10. Consider the language of all strings over {a, b, c} containing the substring "**abcabb**". Write a regular expression that describing this language.

PART – B

ANSWER ALL THE QUESTIONS **3 X 10 = 30 Marks**

11. Construct a LL(1) Parsing Table for the given Context Free Grammar.
    A → (B) | a
    B → B , A | A
12. Design a Lexical Analyser Generator to recognize the patterns in the following LEX specification.

```
%%
aba          {action sequence A1 for pattern P1}
ab*b         {action sequence A2 for pattern P2}
b+a+         {action sequence A3 for pattern P3}
%%
int main ( )
{ yylex();
return 0; }
int yywrap()
        { return 1;}
```

13. Show how the Deterministic Finite Automator is converted directly from an augmented regular expression **r=(0|1)*(0|1)**

--- END---

SASTRA
DEEMED TO BE UNIVERSITY

SCHOOL OF COMPUTING

Second CIA Exam – Sep 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90minutes Max. Marks: 50

PART A

ANSWER ALL THE QUESTIONS **10 MARKS**

1. Consider the grammar **S → aSbS | bSaS |ε**. Show that the grammar is ambiguous for the string abab. (2)
2. What are the Inherited and Synthesized attributes. (2)
3. Write the significance of Augmented Grammar. (2)
4. What conflicts do LR parsers handle during syntax analysis? (2)
5. Illustrate the Three-Address Code (TAC) for the following Python procedure call statement (2)

sum(x+y, n)

PART B

ANSWER ALL THE QUESTIONS **25 MARKS**

6. (a) Construct LALR Parsing table for the following grammar.

S → Aa | bAc | Bc | bBa
A → d
B → a (10)

OR

(b) Construct CLR Parsing table for the following grammar

S → (L) |id
L → L * S | S.
Also parse the input string **(id*id*id).** (10)

7. (a) Check whether the given context free grammar is SLR (1) Grammar or not.

A → B * C | C
B → *C | num
C → B (10)

OR

(b) Translate the given Boolean expression

(a<(c+d))&&((–(a+b)*(a+b)-(a+b)*d)>=MAX)

into the sequence of TAC Statements and represent with Quadruples, Triples and Indirect Triples. (10)

8.(a) Specify the various levels of Errors in programming paradigm. Discuss about the Error Recovery strategies adopted by the Parsers when transforming a code into a parse tree. (5)

OR

(b) Show the Stack implementation of Shift Reduce Parser to validate the string **(id+id)*id** from the grammar

E→ E + T | T

T→ T * id | (E) | id (5)

PART C

ANSWER ALL THE QUESTION **15 MARKS**

9. Write a LEX and YACC specification to evaluate the given expression involving only arithmetic operators in C++. Use the LEX code to validate the numbers and identifiers in the given input. With YACC, augment a context free grammar to validate the syntactical correctness of the expression then define the syntax directed translation rules to evaluate the resultant value of the given arithmetic expression. [note: Evaluation should be made as per the Rule of Precedence and Rule of associativity of Arithmetic operators defined in C++]. (15)

--- END---

SASTRA

SCHOOL OF COMPUTING
CIA – III Exam: Nov 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90minutes Max. Marks: 50

PART A

ANSWER ALL THE QUESTIONS **10 MARKS**

1. What do you understand from the code generated by a Cross Compiler. (2)
2. Give the structure of an activation record. (2)
3. When does dangling reference occur? (2)
4. Discuss the properties of an optimizing compilers. (2)
5. For the given expression, write the TAC statements and build DAG for the sequence. d=(a-b) + (a-c) + (a-c). (2)

PART B

ANSWER ALL THE QUESTIONS **25 MARKS**

6. (a) Construct CLR Parsing table for the following grammar and parse the input string "**abde**".

A → aB | Ad
B → bBC | d
C → e

(10)

OR

(b) Show how the minimized DFA is obtained directly from an augmented regular expression r=(0|1) 0* (0|1) (10)

7. (a) Write the Syntax directed translation scheme to generate the TAC statement for flow of control statements in C++. Illustrate your SDT with the following snippet of code

```
res[i][j] = mergeArrays (arr1, arr2);
  for (int i = 0; i < n1 + n2; i++)
    printArray (res[i][i+2]);
```

(10)

OR

(b) (i) Elaborate the issues in the design of code generator. (5)

(ii) Write short notes on register allocation and register assignment during code generation phase. (5)

8.(a) Explain in detail the various transformations used in peephole optimization to improve the efficiency of target code. (5)

OR

(b) Describe the various Storage allocation strategies used to manage the Run-time memory allocated for the program. (5)

PART C

ANSWER ALL THE QUESTION **15 MARKS**

9. Examine the sequence of Three Address Code statements given below and perform the following tasks.

```
(1)   i := m-1
(2)   j := n
(3)   t1 := 4*n
(4)   v := a[t1]
(5)   i := i+1
(6)   t2 := 4*i
(7)   t3 := a[t2]
(8)   if t3 < v goto (5)
(9)   j := j-1
(10)  t4 := 4*j
(11)  t5 := a[t4]
(12)  if t5 > v goto (9)
(13)  if i >= j goto (23)
(14)  t6 := 4*i
(15)  x := a[t6]
(16)  t7 := 4*i
(17)  t8 := 4*j
(18)  t9 := a[t8]
(19)  a[t7] := t9
(20)  t10 := 4*j
(21)  a[t10] := x
(22)  goto (5)
(23)  t11 := 4*i
(24)  x := a[t11]
(25)  t12 := 4*i
(26)  t13 := 4*n
(27)  t14 := a[t13]
(28)  a[t12] := t14
(29)  t15 := 4*n
(30)  a[t15] := x
```

(a) The Basic blocks of Instructions. (3)

(b) Construct a flowgraph, specify the loops. (3)

(c) For each variable, record its next-use and liveliness at the end of each basic block. (3)

(d) Apply the code improving transformations wherever possible and generate the optimized TAC sequence. (6)

SASTRA DEEMED UNIVERSITY

(A University under section 3 of the UGC Act, 1956)

End Semester Examinations

Nov 2024

Course Code: **CSE320**

Course: **COMPILER DESIGN**

QP No. **:U127-5**

Duration: **3** hours

Max. Marks:**100**

PART - A

Answer all the question **10 x 2 = 20 Marks**

1. Define Cross compiler.
2. Write a regular expression to represent all possible numbers (integer, float, exponential).
3. How and why input buffering is occurring?
4. Prove that the given grammar S → aSbS | bSaS |ε is ambiguous.
5. Write the Context free grammar for conditional statements in C++.
6. Give the structure of an activation record.
7. Write down the SDD for the design of simple desk calculator.
8. How is the liveliness of the variable is calculated by the compiler?
9. Draw the Syntax tree and DAG for the expression a=b* -c + b * -c.
10. Can DAG be used for Optimization? Give Example.

PART – B

Answer all the question **4 x 15 = 60 Marks**

11. Elaborate on the different phases of compiler with a neat sketch. Show the output of each phase of the compiler when the following statement is passed as an input:
SI = (p*n*r)/100; where n should be an integer, p and r could be floating point numbers.

(OR)

12. Explain step by step procedure of constructing a minimized DFA directly from the given regular expression. Illustrate your procedure with the following regular expression (a|b) *a(a|b).

13. Check whether the following grammar can be implemented using a LL (1) parser. Also show the validation of the input string "ia<btd" by using the predictive parser.
S → iEtS | iEtSeS |d
E → id R id
R → < | >

(OR)

14. Construct LALR Parsing table for the following context free grammar. Parse the input string (id+id)*id is accepted by the LALR parser or not.
S → S + R | R
R → R * T | T
T → (S) | id

15. a) Write short notes on the various storage allocation strategies used by Run-Time memory management system during the translation and execution process. (9)

b) Discuss about the Error Recovery strategies adopted by the Parser and Error handler when transforming a code from the token stream to parse tree. (6)

(OR)

16. Define TAC Statement. Specify the different type of TAC Statements. Transform the following Boolean expression ((–(a+b)*(a+b)-(a+b)*d)>MAX) into the sequence of Three Address Code Statements and also represent the sequence with Quadruples, Triples and Indirect Triples structure.

17. a) Write notes on the various issues in the design of Code Generation phase during Compilation process. (8)

b) What are the characteristics of Peephole? Explain the transformations that can be performed on a Peephole window. (7)

(OR)

18. Translate the following snippet of PYTHON code into a sequence of Three Address Code Statements based on the Syntax Directed Translation schemes. Convert the sequence into a flow graph representing the flow of control between different basic blocks.

```
c=0
while((x+y)<=MAX-1):
    sum = sum + c
    if (x>0 && x<10):
            c=c+sum
            display(c)
    c=c+10
display(sum)
```

PART – C

Answer the following **1 x 20 = 20 Marks**

19. a) An online shopping site has set up the following criteria for setting the password for viewing their products and its features:
   i) Password should be 5 to 8 characters long using alphabets and numerals.
   ii) It should start with an alphabet.
   iii) Two special symbols (% and #) are permitted.
   iv) The last character should be a capital letter.
   Write the LEX specification to validate the password given by the user of the site. (5)

   b) Consider the following sequence of TAC statements in a Basic Block, in which all variables are integers and ** denotes exponentiation operation.

```
a=x**2
b=3
c=x
d=c*c
e=b*2
f=a+d
g=e*f
```

   Apply the following optimization techniques to the above basic block in order to compute the result of each transformation. (10)
   i) Algebraic simplification
   ii) Copy propagation
   iii) Common Sub expression elimination
   iv) Constant folding
   v) Specify the Nextuse information for every statement.

   c) Generate the target assembly code for the optimized code. (5)

*****