Sardar Vallabhbhai Patel Institute of Technology, Vasad

Assignment 3 (Examples) (System Programming-2150708)

1. Eliminate left recursion from following grammar.
   S → A
   A → Ad | Ae | aB | aC
   B → bBC | f
   C → g
2. Construct LL(1) parsing table for following grammar.
   S → iCtSeS | iCtS | a
   C → b
3. Given a grammar, E →TA, A → +TA | ε T →VB B →*VB | ε V →id | (E)
   Develop an LL (1) parser table and parse following strings using the parsing table.
   (a) id * (id + id)
   (b) (id*id) + (id*id)
4. Construct predictive parsing table for following grammar:
   E->BA
   A-> &BA|€
   B->true|false
5. Write a regular expression for a language containing a binary string which does not contain two consecutive 0s or two consecutive 1s anywhere.
6. Construct an optimized DFA : (a) 0*1*(0/1)# (b) (0|1)*011
7. Construct NFA and DFA for following regular expression: (0 | 1)*001#
8. Develop regular expression and DFAs for the following kind of strings:
   1. a real number with optional integer and fraction part
   2. a comment string in the C++ language.
9. Show that following regular expressions are equivalent by constructing optimized DFA.
   (0/1)*
   (0*/1*)*
10. Write operator precedence table for arithmetic operators "+", "*", "-", "/" "(", ")". Parse the following expression using the table. id * (id + id)/ (id *id)
11. What is operator precedence parsing? Show operator precedence matrix for following operators: +, - , *, (,). Parse following string: |- <id> + <id> * <id> -|
12. Draw the expression tree for the string f+(x+y)*((a+b)/(c-d)) by their evaluation order and mention register required label in each node.
13. Generate Quadruple, Triple, Indirect Triple for following expression: ans=a+b*c/2.0
14. Given following expression: x = -a * b + -a * b
    (1) Write three address codes for the expression.
    (2) Optimize the three address code if it is possible to do so
15. (3) Give triple implementation for the three address code of the expression
16. Explain operand descriptor and register descriptor for a*b.

17. Given the source program:

```
             START      100

   A         DS          3

    L1       MOVER       AREG, B

             ADD         AREG, C
             MOVEM       AREG, D
   D        EQU         A+1
   L2        PRINT       D
             ORIGIN      A-1
   C        DC          '5'
             ORIGIN       L2+1
             STOP
   B        DC          '19'
             END         L1
```

(a) Show the contents of the symbol table at the end of Pass I.

(b) Explain the significance of EQU and ORIGIN statement in the program and explain how they are processed by the assembler.

(c) Show the intermediate code generated for the program.

16. Write the data structure, intermediate code of following assembly program. Write the assembly program output if value of N = 5.

```
START    101
READ   N
MOVER   BREG, ONE
MOVEM   BREG, TERM    AGAIN
MULT    BREG, TERM
MOVER  CREG, TERM
ADD   CREG, ONE
MOVEM  CREG, TERM
COMP  CREG, N
BC  LE, AGAIN
MOVEM   BREG, RESULT
PRINT    RESULT
STOP
N                DS     1
RESULT           DS     1
ONE              DC    '1'
```

```
            TERM            DS      1
            END
```

17. Consider following assembly language program:
 Show (i) Contents of Symbol Table (ii) Intermediate codes using Variant I representation.

```
            START 101
            READ N
            MOVER           BREG, ONE
            MOVEM           BREG, TERM
AGAIN       MULT BREG, TERM
            MOVER CREG, TERM
            ADD CREG, ONE
            MOVEM CREG, TERM
            COMP CREG, N
            BC LE, AGAIN
            MOVEM BREG, AGAIN
            PRINT RESULT
            STOP
            N DS 1
            RESULT DS 1
            ONE DC ‘1’
            TERM DS 1
            END
```

Instruction opcode: STOP – 00, ADD – 01, MULT – 03, MOVER – 04, MOVEM – 05, COMP – 06, BC – 07, READ – 09, PRINT – 10, LE – 02 Assembler directives: START – 01, END – 02 Declaration statements: DC – 01, DS – 02 Register code: BREG – 02, CREG – 03

Prepared by:
Vibhavari Patel(CC)
Assistant Professor
Information Technology