1st Semester 2020-21; PPL (CSF301)

Comprehensive examination

Part-B Max marks : 45 (90 Mins) Date 23-Dec-2020

On Google Forms

Q1. Give a CFG for the language L over ∑= {a, b} such that any string contains exactly two ***a***s.

[5M]

Q2. Prove that the following CFG is ambiguous. [4M]

S→ a | abSb | aAb

A→ bS | aAAb

Note: Non-terminals={S, A} terminals={ a,b}

Q3: Convert the following EBNF to BNF as discussed in Textbook. [4]

S→ M{aa}$^+$ [M]

M→ b[(a|b)]N

N→ c(d | e)[ab]

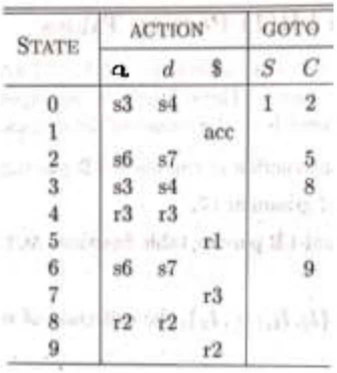
Set of Non-terminals={S, M, N} Set of terminals={a, b, c, d, e}

starting Non-terminal=S.

Q4. Consider the have Grammar G with following rules-

1. **S→CC**
2. **C→aC**
3. **C→d**

And the below LR Parser table.



STATE	ACTION			GOTO	
	a	d	$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

Using the above Grammar (with rule numbers) and the LR parsing table given above, give the process of parsing the valid string ***daaad*** with table entries with relevant columns as discussed in the class.

You are expected to give the complete set of table entries represented in the below format.

<row number; stack; input string; action> .

For example, the first entry of the table is- ***<R1; 0; daaad$; S4 >***

Do not forget to include row number. The GOTO / SHIFT **Action** component should be elaborate as given in the textbook. [7]

Q5. Give the computation for the weakest pre-condition of the Following. [2.5+2.5=5]

```
(i)     if (a==d)
            b=(3*a)+2;
        else
            b=4*(a-3);
        {b>8}

(ii)    a=2*(2*b-1);
        b=(3*a)-2;
        {b>4}
```

Q6. A grammar G has rules – { S→ 0S | 1AA; A→ 0|1A|0B; B→ 1 | 0BB}. [2.5+2.5=5]

Note: the semicolon ‘;’ is a separator between rules and ‘|’ is for alternation, they are not part of the grammar.

Give the *handle* for the following sentential Forms.

(i) 01A100BB

(ii) 1A1100B1

Q7. Assume that in Prolog program we have stored the following facts about employees.

```
// facts about employee name and basic pay
Basicpay(Kiran, 8000).
Basicpay(Tom, 6000).
Basicpay(Monica, 12000).
Basicpay(Abdul, 6000).
// facts about working place of employee and the house rent % on the basic pay of the employee
working in that city
Houserent(Mumbai, 20).
Houserent(Delhi, 30).
// facts about Employee work place
Workcity(Kiran, Delhi).
Workcity(Tom, Delhi).
Workcity(Monica, Mumbai).
```

Workcity(Abdul, Mumbai).

Note: Houserent(Delhi, 30). Implies that house rent for employees working in city Delhi is 30% of his/her basic pay.
Now write/complete the 'Salary' computation (for a given employee) statement-
Salary(empname, empsalary):- ________________________________.
(Give the complete Salary computation statement in box provided)
Apply following logic for salary computation
Salary= (basic + (basic * rent percentage for the city/100))
Ex: salary of Kiran = 8000+ (8000*(30/100)) =102400
Hence if we give the query   **Salary (Kiran, sal)** it should display **sal** and its **value** computed for kiran.
[5 marks]

Q8. Consider the following skeletal C program.   [2.5+2.5=5]

```
void fun1(void);
void fun2(void);
void fun3(vaid);
void fun4(void);
 void main() {
    int a, c, d;
    . . .
  }
  void fun1(void){
   int a, b, c;
   .  .  .
}
  void fun2(void){
   int a, c, d, e;
   .  .  .
}
void fun3(void){
   int  b, c, d;
   .  .  .
}
void fun4(void){
   int  a, d, e, f;
   .  .  .
}
```

Given the following calling sequences, and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it is defined.

(i) main calls fun3, fun3 calls fun1, fun1 calls fun4 and fun4 calls fun2.

(ii) main calls fun1, fun1 calls fun3, fun3 calls fun2, and fun2 calls fun4

Q9. Assume call-by-reference is used in the C-style pseudo code shown below.
In what order should the functions P, Q and R be called in *main()* in the blanks specified with comments *1st function call , 2nd function call, and 3rd function call ,* to get the output –
"10 22 22 22 22 20 20" . Just give the order in which P, R, and Q are called. [5]
Note: You need not look for minor syntax errors like ';' etc. Focus on the actual concepts.

```
int main ()
{
        int x;
        int Q (int z)  {                // body of function Q
         z = z + x;
         printf ("%d\n", z);
        }
        int P (int y)  {                // body of function P
         int x;
         x = y + 2;
         Q (x);
         printf ("%d\n", x);
        }
        int R (int w)  {                // body of function R
         w = w + x;
         printf ("%d\n", w);
         P(x);
        }
        x = 5;                 //body of function main
        ____  ;  // 1st  function call

       _____ ; // 2nd function call

       _____ ;  // 3rd  function call

        printf ("%d\n", x);
        return 0;
}    //end of main
```