Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Compiler Theory: CS31003
3rd year CSE, 5th Semester

Theory Quiz *Marks: 25*
Date: October 15, 2020

1. Identify the correct statement/(s) regarding FIRST and FOLLOW sets of the given grammar. [2]

```
S -> aBDh

B -> cC

C -> bC / ∈

D -> EF

E -> g / ∈

F -> f / ∈
```

   a) FIRST(D) = {g}
   b) FIRST(S) = {a, c, b, g, f }
   c) FIRST(D) = {g, f, ∈}
   d) FOLLOW(D) = {h}

   Answer: c), d)

2. Consider the following grammar [2]

```
0) S' -> S
1) S -> XX
2) X  -> aX
3) X -> b
```

   The LR(1) states are as follows

I0 :
S' –> •S, $
S –> •XX, $
X –> •aX, a/b
X –> •b, a/b

I1 :
S' –> S•, $

I2:
S –> X•X, $
X –> •aX, $
X –> •b, $

I3:
X –> a•X, a/b
X –> •aX, a/b
X –> •b, a/b

I4 :
X –> b•, a/b

I5:
S –> XX•, $

I6:
X –> a•X, $
X –> •aX, $
X –> •b, $

I7:
X –> b•, $

I8:
X –> aX•, a/b

I9:
X –> aX•, $

Identify the correct parsing table

a)

	Action			Goto	
State on stack top	a	b	$	S	X
0	s3	s4		1	2
1			acc		
2	s3	s3			5
3	s6	s7			8
4	r3	r3			
5			r1		
6	s4	s4			9
7			r3		
8	r2	r2			
9			r2		

b)

	Action			Goto	
State on stack top	a	b	$	S	X
0	s3	s4		1	3
1			acc		
2	s6	s7			8
3	s3	s4			9
4	r3	r3			
5			r1		
6	s6	s7			5
7			r3		
8	r2	r2			
9			r2		

c)

	Action			Goto	
State on stack top	a	b	$	S	X
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		

d)

	Action			Goto	
State on stack top	a	b	$	S	X
0	s3	s4		1	9
1			acc		
2	s3	s3			2
3	s6	s7			5
4	r3	r3			
5			r1		
6	s3	s4			8
7			r3		
8	r2	r2			
9			r2		

Answer: c)

3. Consider the following grammar [2]

```
S'-> S
S -> aBc | bCc | aCd | bBd
B -> e
C -> e
```

The LR(1) states are as follows

I0:
S' –> •S, $
S –> •aBc, $
S –> •bCc, $
S –> •aCd, $
S –> •bBd, $

I1:
S' –> S•, $

I2:
S –> a•Bc, $
S –> a•Cd, $
B –> •e, c
C –> •e, d

I3:
S –> b•Cc, $
S –> b•Bd, $
C –> •e, c
B –> •e, d

I4:
S –> aB•c, $

I5:
S –> aC•d, $

I6:
B –> e•, c
C –> e•, d

I7:
S –> bC•c, $

I8:
S –> bB•d, $

I9:
B –> e•, d
C –> e•, c

I10:
S –> aBc•, $

I11:
S –> aCd•, $

I12:
S –> bCc•, $

I13:
S –> bBd•, $

Identify any two states, which if merged leads to a **reduce-reduce conflict**. Mention the states in ascending order and only the numbers. Do not append the answer with the word 'state' or 'I' or 'S'. Like if state 1 can be merged with state 5, then mention the answer as `15` only (without any space in between). Answers like `1 5` or `51` or `I1I5` or `state1 state5` will not be accepted.

Answer: *69*

4. Consider the following LALR(1) grammar [2]

```
S' -> S
S -> Bbb | aab | bBa
B -> a
```

Some of the states are given below

I0:
S' –> •S, $
S –> •Bbb, $
S –> •aab, $
S –> •bBa, $
B –> •a, b

I1:
S' –> S•, $

I2:
S –> B•bb, $

I3:
S –> a•ab, $
B –> a•, b

.....
.....

There is a shift reduce conflict in State 3 of LALR(1). State TRUE or FALSE.

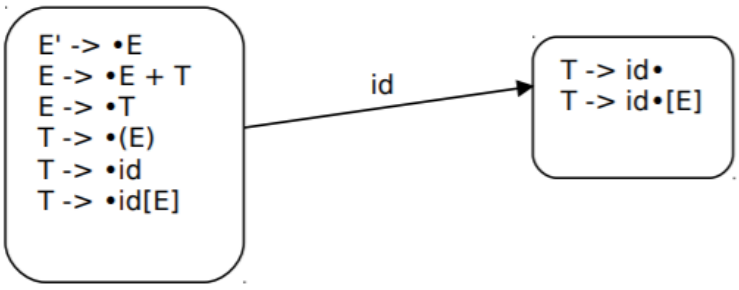
(a) TRUE
(b) FALSE

Answer: *FALSE*

5. Consider the following grammar [2]

```
E'-> E
E -> E + T | T
T -> (E) | id | id[E]
```

The first two states are shown below



Identify the correct statement/s

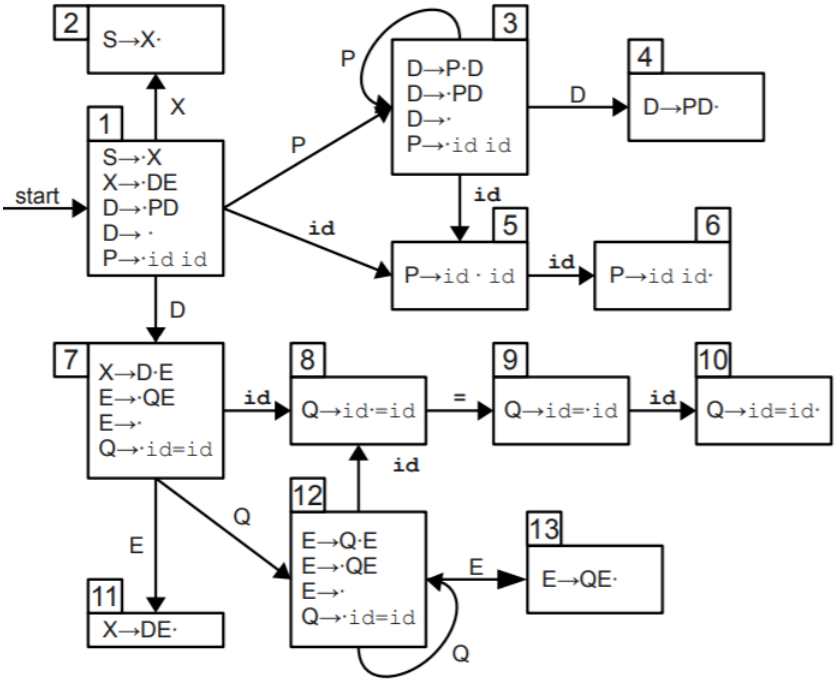
a) The grammar is LR(0)
b) The grammar is SLR(1)
c) The grammar is both LR(0) and SLR(1)
d) The grammar is LR(1)

Answer: b), d)

In an LR(0) parser, the set on the right has a shift-reduce conflict. However, an SLR(1) will compute Follow(T) = { +)] $ } and only enter the reduce action on those tokens. The input [will shift and there is no conflict. Thus this grammar is SLR(1) even though it is not LR(0).

6. Consider the following grammar [2]

```
S -> X
X -> DE
D -> PD
D -> ∈
E -> QE
E -> ∈
P -> id id
Q -> id = id
```



Identify the correct statement(s)

a) State 1 contains a shift reduce conflict

b) The grammar is not SLR(1)

c) FOLLOW(D) does not contain `id`

d) The grammar is LR(0)

Answer: a), b)

7. Consider the following grammar [2]

```
S->   A        (1)
A  -> aBE      (2)
B  -> bCD      (3)
C  -> c        (4)
```

```
D -> d        (5)
E -> eFG      (6)
F -> f        (7)
G -> g        (8)
```

Identify the correct rightmost derivation of the string `abcdefg`

a) 4 5 7 3 8 6 2 1

b) 4 5 7 3 6 2 8 1

c) 1 2 6 8 7 3 4 5

d) 1 2 6 8 7 3 5 4

Answer: d)

8. Identify the correct statement [2]

   a) SLR(1) is a subset of all LR(0) grammars

   b) LR(1) considers more than 1 lookahead symbols

   c) Conversion of LR(1) to LALR(1) may introduce new shift/reduce conflicts

   d) In LALR(1) states, two or more LR(1) states having the same set of core LR(0) items may be merged into one by combining the look-ahead symbols for every item

   Answer: d)

9. Let A denote a 2×3 array of integers, and let c, i, and j denote integers. Assume the size of integer is 4 bytes. What will be the three-address translation for the expression `c + A[i][j]`? [1]

   a)
   ```
   t1 = i * 4
   t2 = j * 12
   t3 = t1 + t2
   t4 = A[t3]
   t5 = c + t4
   ```

   b)
   ```
   t1 = i * 12
   t2 = j * 4
   t3 = t1 + t2
   t4 = A[t3]
   t5 = c + t4
   ```

   c)
   ```
   t1 = i
   t2 = j
   t3 = t1 + t2
   t4 = A[t3]
   t5 = c + t4
   ```

   d)
   ```
   t1 = i * 12
   t2 = j
   t3 = t1 + t2
   t4 = A[t3]
   t5 = c + t4
   ```

Answer: b)
Reference: Dragon Book Example 6.12

10. Which of the following is NOT used for intermediate code representation? [1]

    a) Control Flow Graph.

    b) Program Dependence Graph.

    c) DJ-Graph

    d) Bipartite Graph.

    Answer: d)
    Reference: Module 5 - IC Translation Slide no. 7

11. Consider the below program. [2]

```
int a[10], b[10], dot_prod, i;
int *a1, *b1;
dot_prod = 0, a1 = a, b1 = b;
for(i=0;i<10;i++) dot_prod += *a1++ * *b1++;
```

    Identify the correct three-address code representation (not optimised). Assume that the declarations and initialization have already been processed.

    a)
```
i=0
L1: if(i>=10) goto L2
T3 = *a1
T4 = a1+1
a1 = T4
T5 = *b1
T6 = b1+1
b1 = T6
T7 = T3*T5
T8 = dot_prod+T7
dot_prod = T8
T9 = i+1
i = T9
goto L1
L2:
```

    b)
```
i=0
L1: if(i>=10) goto L2
T3 = *a1
T4 = a1+1
a1 = T4
T5 = *b1
T6 = b1+1
b1 = T6
T7 = T4*T6
T8 = dot_prod+T7
```

```
dot_prod = T8
i = i+1
goto L1
L2:
```

c)
```
i=0
L1: if(i>=10) goto L2
T3 = *a1
T4 = a1+1
a1 = T4
T5 = *b1
T6 = b1+1
b1 = T6
T7 = T3*T5
T8 = dot_prod+T7
dot_prod = T8
i = i+1
goto L1
L2:
```

d)
```
i=0
L1: if(i>=10) goto L2
T3 = *a1
T4 = a1+4
a1 = T4
T5 = *b1
T6 = b1+4
b1 = T6
T7 = T3*T5
T8 = dot_prod+T7
dot_prod = T8
T9 = i+1
i = T9
goto L1
L2:
```

Answer: a)
Reference: https://nptel.ac.in/content/storage2/courses/106108113/module5/Lecture17.pdf

12. Consider the below code segment.

```
int main(int){
    int d;
    {
        int a=10;
        {
            int b = 1;
            while(a){
                d++;
                a--;
```

```
            }
        }
    }
    printf("%d", d);
}
```

How many symbol tables will be created in order to generate three-address code of the above code segment? [1]

(a) 2

(b) 5

(c) 4

(d) 3

Answer: 5
Explanation: Symbol table for main, local block, nest local block, while loop and global.

13. Fill in the blank below. [1]
In case of _____ scoping, symbol table may be needed at compile time as well as run time. (Use one word answer)

Answer: dynamic
Reference: Module 5: IC Translation Slide no. 22

14. Consider the following expression grammar with actions.

```
L -> LS \n      {}
L -> S \n       {}
S -> id = F     {emit(id.loc = F.loc);}
F -> F1 - F2    {F.loc = gentemp(); emit(F.loc = F1.loc - F2.loc);}
F -> F1 + F2    {F.loc = gentemp(); emit(F.loc = F1.loc + F2.loc);}
F -> (F1)       {F.loc = F1.loc;}
F -> -F1        {F.loc = gentemp(); emit(F.loc = -F1.loc);}
F -> number     {F.loc = gentemp(); emit(F.loc = number.val);}
F -> id         {F.loc = id.loc;}
```

How many temporary variables are needed (or gentemp() calls are used) are needed to convert below expression into intermediate three address code (not optimized code)? Consider that the addition and subtraction operations are left associative. [1]

```
x = 12 - (2 - (-3) - 4)
```

a) 6

b) 7

c) 8

d) 9

Answer: c)
Explanation:

```
t0 = 12
t1 = 2
t2 = 3
t3 = -t2
t4 = 4
t5 = t1 - t3
t6 = t5 - t4
t7 = t0 - t6
x = t7
```

15. Consider the following grammar.

S -> while M1 (B) M2 S1
M -> ϵ

Where B is any boolean expression. What will be the action statements for the grammar? Standard attributes and methods should be used. [2]

a) *backpatch(S1.nextlist, M1.instr);*
backpatch(B.truelist, M2.instr);
S.nextlist = B.falselist;
emit('goto', M1.instr);

b) *backpatch(S1.nextlist, M2.instr);*
backpatch(B.truelist, M1.instr);
S.nextlist = B.falselist;
emit('goto', M1.instr);

c) *backpatch(S1.nextlist, M1.instr);*
backpatch(B.truelist, M2.instr);
S.nextlist = B.truelist;
emit('goto', M2.instr);

d) *backpatch(S1.nextlist, M1.instr);*
backpatch(B.falselist, M2.instr);
S.nextlist = B.truelist;
emit('goto', M1.instr);

Answer: a)
Reference: Module 5: Backpatching - control construct grammar with actions