CMR
INSTITUTE OF
TECHNOLOGY

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

Internal Assessment Test II – May, 2017

Sub:	Compiler Design							Code:	10CS63
Date:	9/05/2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	CSE

Answer Any FIVE FULL Questions

		Marks	OBE	
1.	Consider the grammar **S → AA A → A a | b** (i) Compute sets of LR (1) items (ii) Construct canonical LR(1) parsing table.	[5+5]	CO1	L1
2.	Construct the parsing table for LALR (1) parser using the following grammar. **S → CC C → a C C → d**	[10]	CO2	L3
3. (a)	Define the following with examples: (i) Quadruples (ii) Triples (iii) Indirect Triples	[4]	CO3	L3
(b)	Construct a DAG and a three address-code for the expression **a+2*(a-3)+(a-3)*3**	[6]	CO3	L3
4.	Obtain SDD for simple type declaration. Construct a dependency graph for the declaration **int a,b,c** along with evaluation order.	[10]	CO3	L3
5.(a)	Generate 3-address statement for the following programming construct. i=1 , sum=0 do { sum+= a[i] * b[i] i++ }while(i<=20)	[6]	CO3	L3
(b)	Write 3-address code for **switch statement**.	[4]		
6.	For the given productions shown below, write semantic rules and construct **annotated parse tree for 20*3+5** L → E, E → E+T, E → T, T → T*F,T → F, F → (E),F → digit	[10]	CO3	L3

CMR

Scheme of Evaluation & Solution

Internal Test – II, May 2017

Sub: Compiler Design (10CS63)

1. ① $S \rightarrow AA$

② $A \rightarrow Aa$

③ $A \rightarrow b$

Sets of LR(1) items (5M)

I_0: $S' \rightarrow .S, \$$

$S \rightarrow .AA, \$$

$A \rightarrow .Aa, b/a$

$A \rightarrow .b, b/a$

I_1: GOTO (I_0, S)

$S' \rightarrow S., \$$

I_2: GOTO (I_0, A)

$S \rightarrow A.A, \$$

$A \rightarrow A.a, b/a$

$A \rightarrow .Aa, \$/a$

$A \rightarrow .b, \$/a$

I_3: GOTO (I_0, b)

$A \rightarrow b., b/a$

I_4: GOTO (I_2, A)

$S \rightarrow AA., \$$

$A \rightarrow A.a, \$/a$

I_5: GOTO (I_2, a)

$A \rightarrow Aa., b/a$

I_6: GOTO (I_2, b)

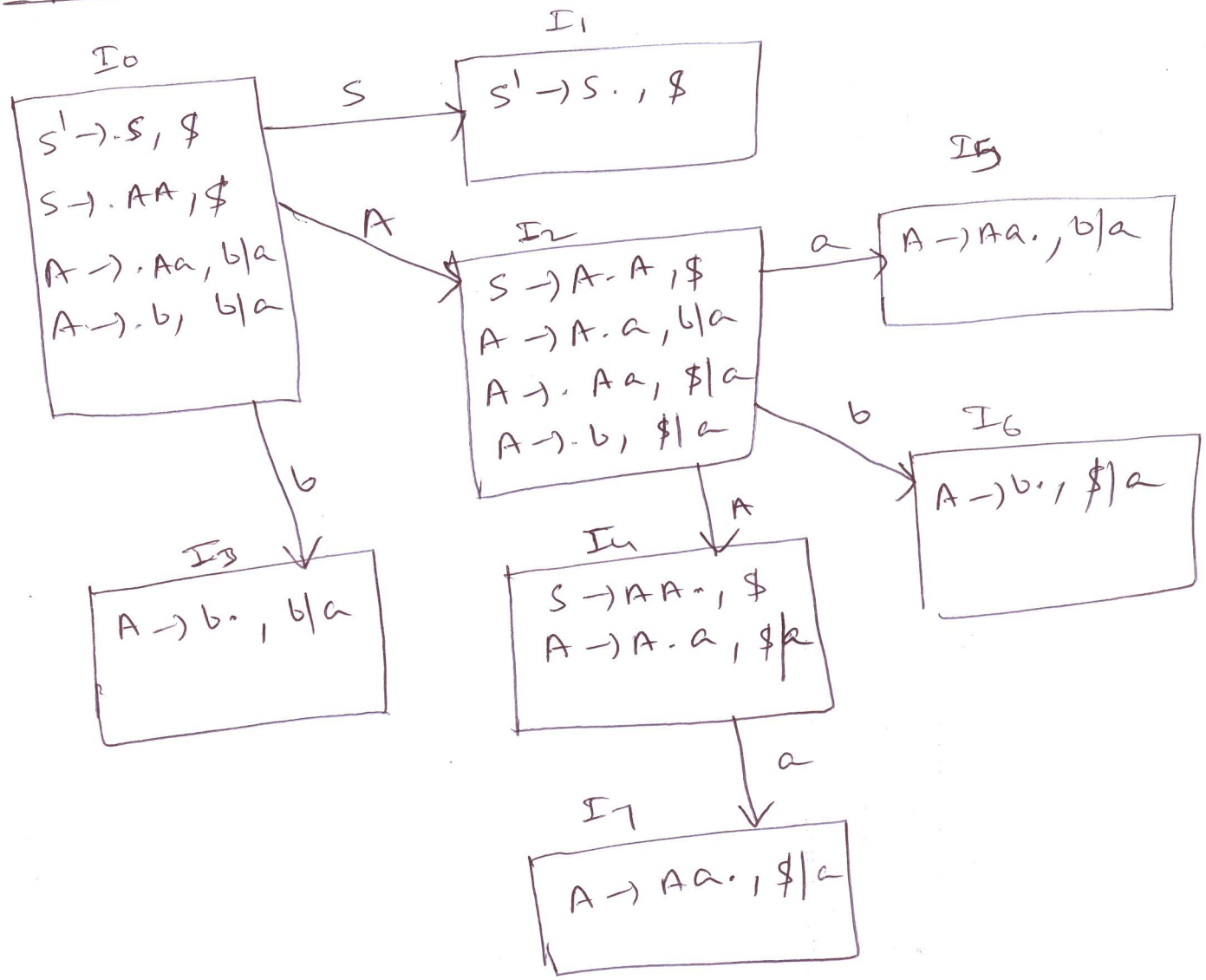
$A \rightarrow b., \$/a$

I_7: GOTO (I_4, a)

$A \rightarrow Aa., \$/a$

Canonical LR(1) Parsing table

Transition diagram



Parsing table

Canonical LR(1) Parsing table (5M)

States	Action a	Action b	Action $	GOTO S	GOTO A
0		S3		1	2
1			Accept		
2	S5	S6			4
3	r3	r3			
4	S7		r1		
5	r2	r2			
6	r3	r3			
7	r2		r2		

Q.2

① $S \rightarrow CC$

② $C \rightarrow aC$

③ $C \rightarrow d$

LR(1) items

I_0

$S' \rightarrow .S, \$$

$S \rightarrow .CC, \$$

$C \rightarrow .aC, a/d$

$C \rightarrow .d, a/d$

I_1 : GOTO(I_0, S)

$S' \rightarrow S., \$$

I_2 : GOTO(I_0, C)

$S \rightarrow C.C, \$$

$C \rightarrow .aC, \$$

$C \rightarrow .d, \$$

I_3 : GOTO(I_0, a)

$C \to a \cdot C$, a/d

$C \to \cdot aC$, a/d

$C \to \cdot d$, a/d

I_4 : GOTO(I_0, d)

$C \to d \cdot$, a/d

I_5 : GOTO(I_2, C)

$S \to CC \cdot$, \$

I_6 : GOTO(I_2, a)

$C \to a \cdot C$, \$

$C \to \cdot aC$, \$

$C \to \cdot d$, \$

I_7 : GOTO(I_2, d)

$C \to d \cdot$, \$

I_8 : GOTO(I_3, C)

$C \to aC \cdot$, a/d

I_9 : GOTO(I_5, a)

$C \to a \cdot C$, a/d

$C \to \cdot aC$, a/d

GOTO(I_3, a) = I_3

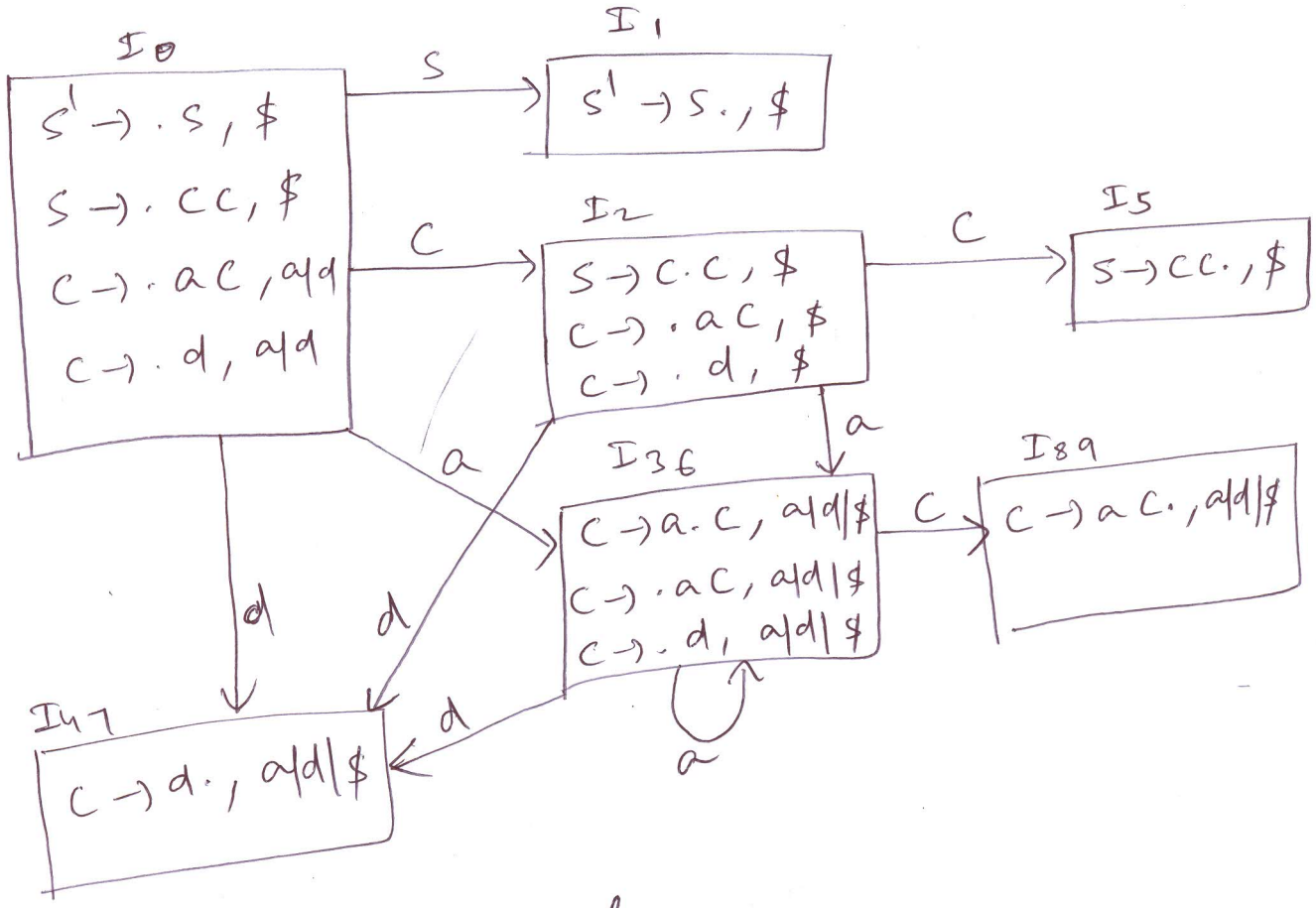
GOTO(I_3, d) = I_4

I_9 : GOTO(I_6, C)

$C \to aC \cdot$, \$

GOTO(I_6, a) = I_6

GOTO(I_6, d) = I_7



LALR(1) Parsing table

States	ACTION a	ACTION d	ACTION $	GOTO S	GOTO C
0	S36	S47		1	2
1			Accept		
2	S36	S47			5
36	S36	S47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

Q.3. (a)

Ex $a = b + c * 5$

(1 + 3) Mark

3-add code

$T_1 = c * 5$

$T_2 = b + T_1$

$a = T_2$

Quadruple

Record index	Operator	Opnd 1	Opnd 2	Result
0	*	c	5	T_1
1	+	b	T_1	T_2
2	=	T_2	—	a

Triple

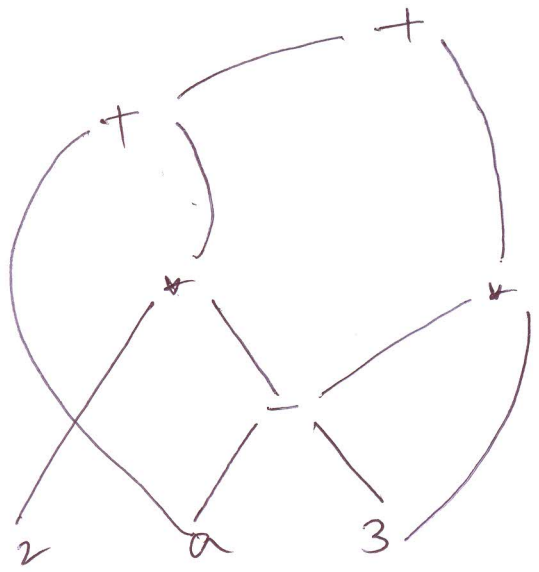
Record index	Operator	Opnd 1	Opnd 2
0	*	c	5
1	+	b	[0]
2	=	[1]	—

Indirect Triple

Address	Pointer to the Record
100	[0]
101	[1]
102	[2]

(b) $a + 2 * (a-3) + (a-3) * 3$

DAG



[3M]

3 - add code

[3M]

$T_1 = a - 3$

$T_2 = 2 * T_1$

$T_3 = T_1 * 3$

$T_4 = a + T_2$

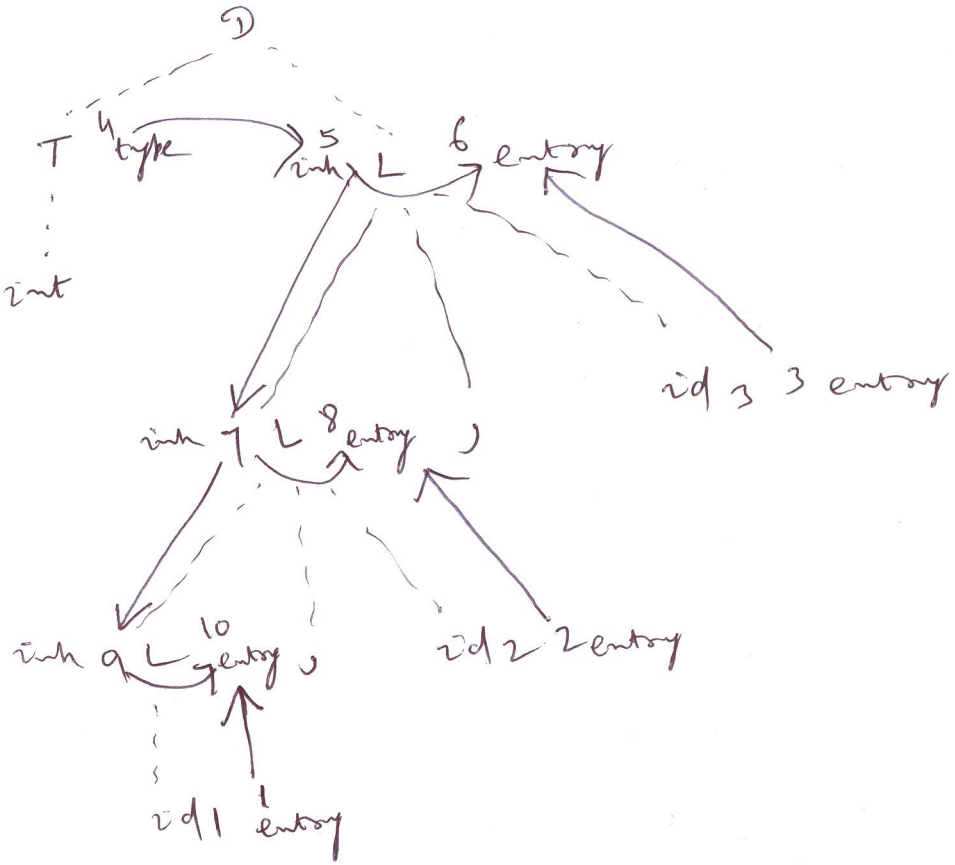
$T_5 = T_4 + T_3$

Q.4 SDD for simple type declaration (5M)

Prod^n	Semantic Rules
1) $D \rightarrow TL$	L.inh = T.type
2) $T \rightarrow$ int	T.type = integer
3) $T \rightarrow$ float	T.type = float
4) $L \rightarrow L_1$, id	L_1.inh = L.inh addType(id.entry, L.inh)
5) $L \rightarrow$ id	addType(id.entry, L.inh)

Dependency graph for int a, b, c

(5M)



Order of evaluation : ~~10 2 8 3 6~~ ~~5~~

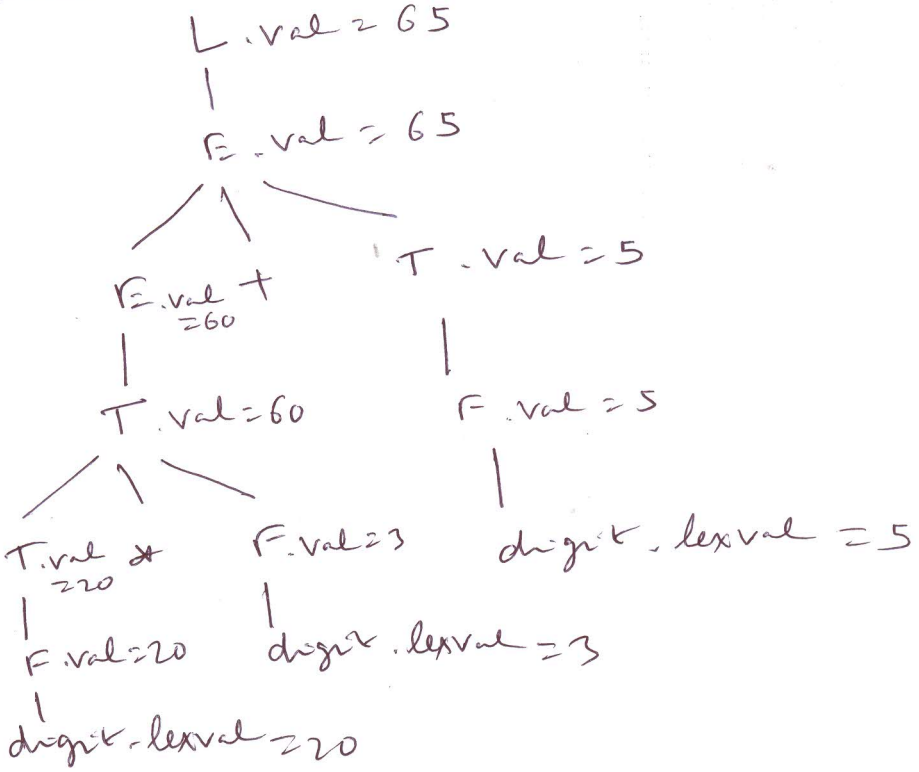
1 2 3 4 5 6 7 8 9 10

CMR

Q.5. (5M)

Productions	Semantic Rules
(1) L → E	L.val = E.val
(2) E → E + T	E.val = E.val + T.val
(3) E → T	E.val = T.val
(4) T → T * F	T.val = T.val * F.val
(5) T → F	T.val = F.val
(6) F → (E)	F.val = E.val
(7) F → digit	F.val = digit.lexval

Annotated Parse tree for 20*3 +5 (5M)



CMR

6.(a)

```
i=1, sum=0
do
{
 sum += a[i] * b[i]
  i++
}while(i<=20)
```

3-add code for the above program segment

```
    i=1
    sum=0
L1: T1 = baseadd(a)
    T2 = i*4
    T3 = T1[T2]
    T4 = baseadd(b)
    T5 = T4[T2]
    T6 = T3 * T5
    T7 = sum + T6
    sum = T7
    T8 = i+1
    i = T8
    if i<=20 goto L1
L2:
```

(6M)

(b) 3 - add code for switch statement

```
switch (expr)
{
 case V1:  S1 ; break ;
 case V2 :  S2; break ;
   :
case Vn-1 :  Sn-1 ; break;
default :  Sn ; break;
}
```

(4M)

3-add code

```
Evaluate expr (result in T)
if T == V1  goto L1
if T == V2  goto L2
   :
if T == Vn-1  goto Ln-1
goto Ln

L1: code for S1
    goto next
L2: code for S2
    goto next
   :
Ln-1: code for Sn-1
      goto next
Ln: code for Sn
    goto next

next :
```