18101061 19.5

Department of Computer Science & Engineering
University of Asia Pacific (UAP)

Class Test **Fall 2021** **4th Year 2nd Semester**

Course Code: CSE 429 **Course Title: Compiler Design** **Credits: 3**

Full Marks: 20 **Duration: 30 minutes**

1. Consider the following Context-Free Grammar (CFG):

   STMT→ TERM | xb
   TERM→ a TERM b | FACTOR
   FACTOR→ x

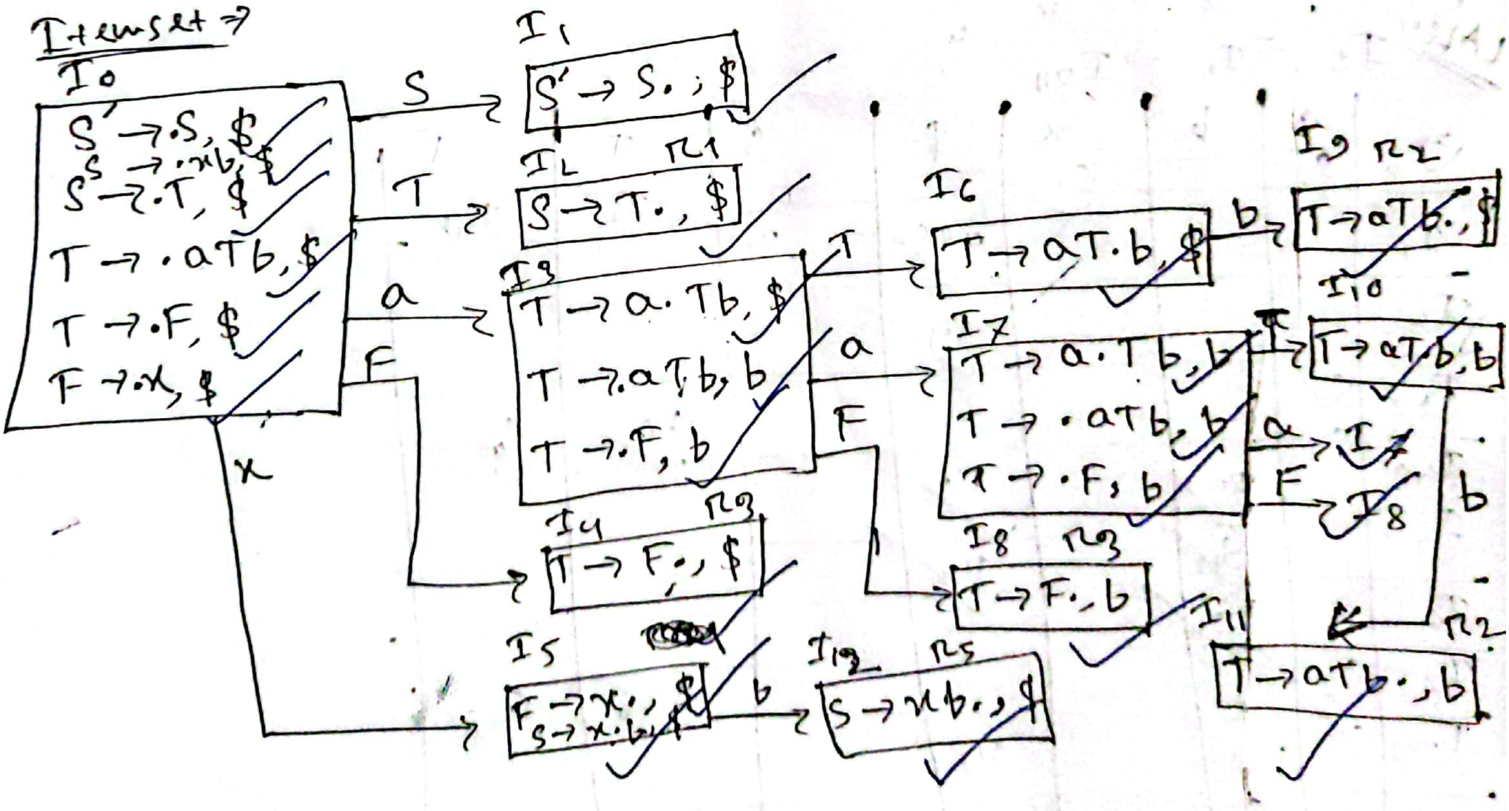
   a) Find out the LR(1) item sets for the above stated grammar.
   b) Construct two parse tables for i) CLR(1) ii) LALR(1)

Let,
S = STMT
T = TERM
F = FACTOR

a) S → T | xb
T → aTb | F
F → x

Modified →
0. S' → S 2. T → aTb 4. F → x
1. S → T 3. T → F 5. S → xb

Itemset →



Scanned with CamScanner

(b) (i) CLR(1) ⇒

	a	b	x	$	S	T	F
I_0	S_3		S_5		I_1	I_2	I_4
I_1				acc			
I_2				r_1			
I_3	S_7					I_6	I_8
I_4				r_3			
I_5		S_{12}		r_4			
I_6		S_9				I_{10}	I_8
I_7	S_7						
I_8		r_3					
I_9				r_2			
I_{10}		S_{11}					
I_{11}		r_2					
I_{12}	~~LALR~~			r_5			

(ii) $I_4 \cong I_8 \Rightarrow I_{48}$

LALR ⇒ $I_9 \cong I_{11} \Rightarrow I_{9,11}$

	a	b	x	$	S	T	F
I_0	S_3		S_5		I_1	I_2	I_{48}
I_1				acc			
I_2				r_1			
I_3	S_7					I_6	I_{48}
I_{48}		r_3		r_3			
I_5		S_{12}		r_4			
I_6		S_{911}					
I_7	S_7					I_{10}	I_{48}
I_{911}		r_2		r_2			
I_{10}		~~S_{11}~~					
I_{12}				r_5			

Original Grammar:

$EXPR1 \rightarrow EXPR1 + EXPR2 \mid EXPR2 \mid EXPR1 = EXPR2$

$EXPR2 \rightarrow (EXPR1) \mid id$

$EXPR3 \rightarrow id$

Let, $EXPR1 = E$

$EXPR2 = F$

$EXPR3 = T$

Now, the grammar becomes,

$E \rightarrow E+F \mid F \mid E=F$	$First(E) = \{(, id\}$
$F \rightarrow (E) \mid id$	$First(F) = \{(, id\}$
$T \rightarrow id$	$First(T) = \{id\}$

Modified Grammar for LR(0) itemset,

$E' \rightarrow E$

$E \rightarrow E+F$

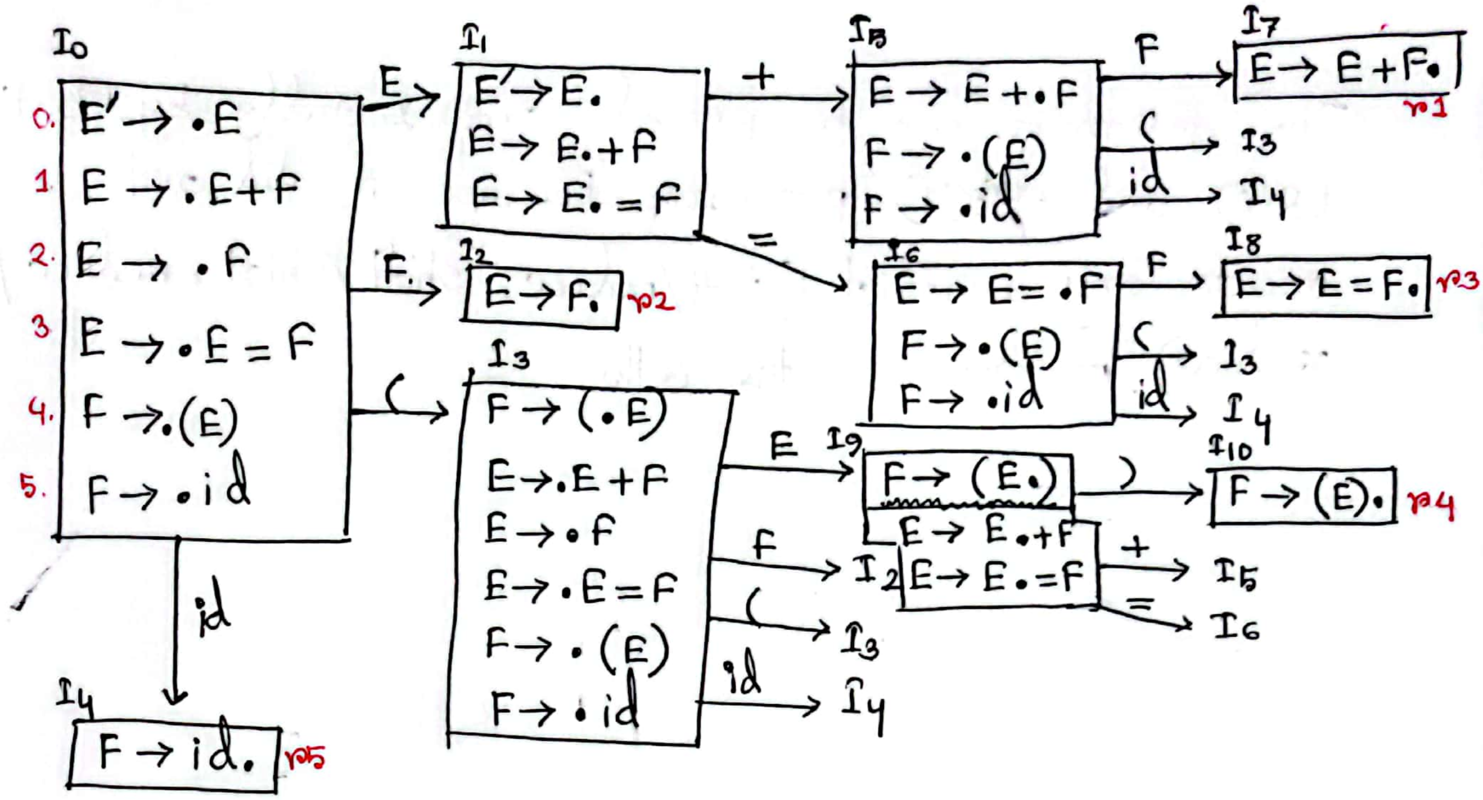
$E \rightarrow F$

$E \rightarrow E=F$

$F \rightarrow (E)$

$F \rightarrow id$

$\boxed{T \rightarrow id}$ $\longrightarrow$ We are not going to add the production of T in our initial itemset. Please think why?



	+	=	(	)	id	$	E	F	–
I0			S3		S4		I1	I2	
I1	S5	S6				acc			
I2	r2	r2	r2	r2	r2	r2			
I3	~~S3~~	8·	S3		S4		I9	I2	
I4	r5	r5	r5	r5	r5	r5			
I5			S3		S4			I7	
I6			S3		S4			I8	
I7	r1	r1	r1	r1	r1	r1			
I8	r3	r3	r3	r3	r3	r3			
I9	S5	S6		S10					
I10	r4	r4	r4	r4	r4	r4			

←Action→←Go-to→

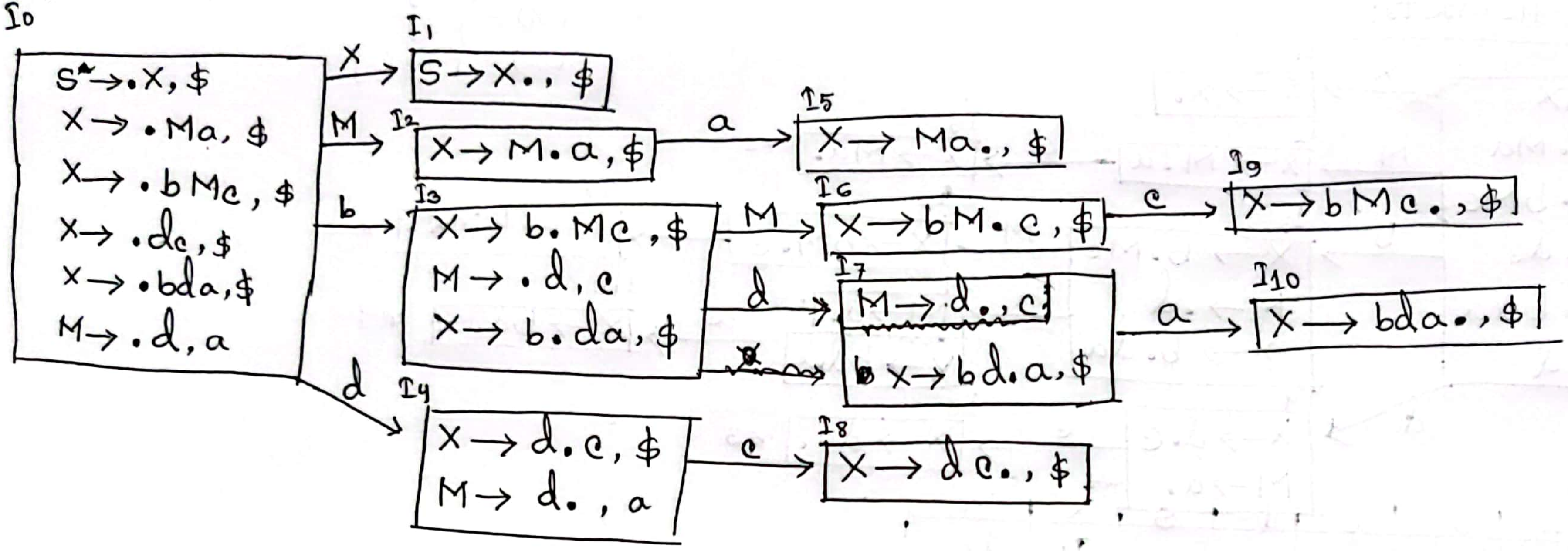
	+	=	(	)	id	$	E	F
I0			S3		S4		I1	I2
I1	S5	S6				acc		
I2			r2		r2			
I3			S3		S4		I9	I2
I4			r5		r5			
I5			S3		S4			I7
I6			S3		S4			I8
I7			r1		r1			
I8			r3		r3			
I9	S5	S6		S10				
I10			r4		r4			

←Action→←Go-to→

Scanned with CamScanner

The grammar can be parsed successfully by both LR(0) and SLR(1) parsers. Because it did not create any conflict (shift/reduce, shift/shift, reduce/reduce). in any of the cells.

LR (1) itemsets



This is LALR(1), we can say ~~we~~ even without creating the parsing table. How?

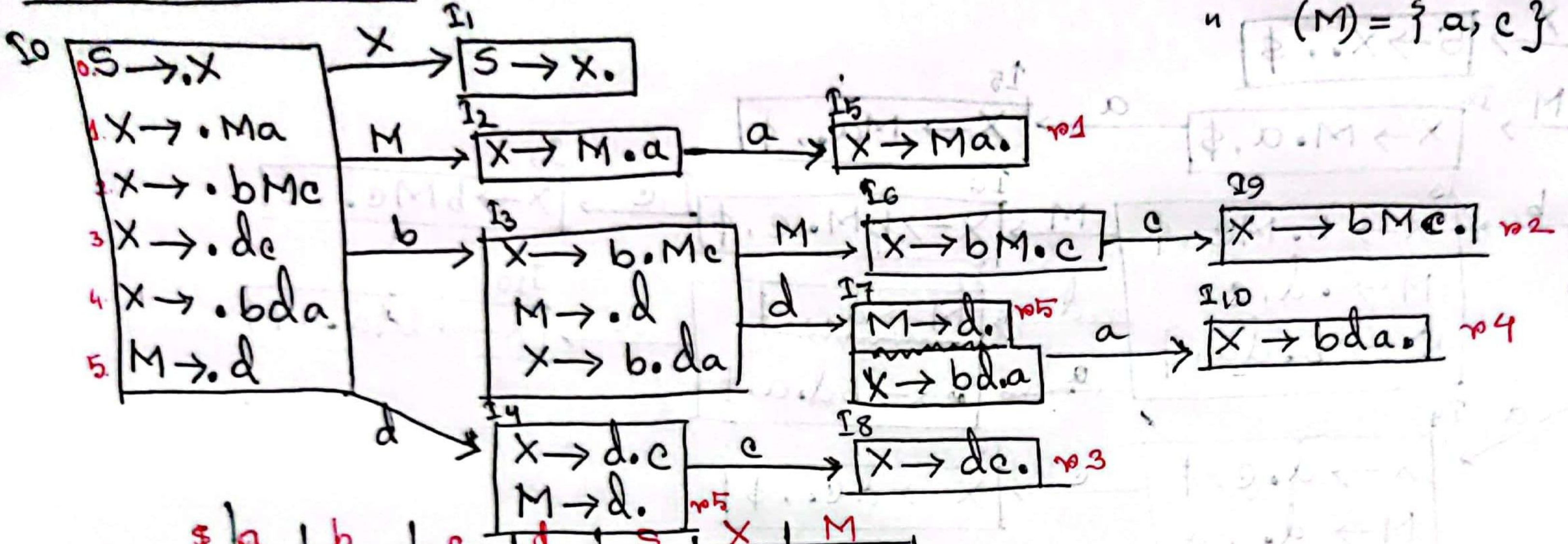
Look closly the itemsets. There are no two states that we can marge. When can we marge two itemsets while creating the LALR(1) parsing table? When two itemsets have identical transition (including the dot(•)) but they have different look ahed operators.

But that is not sufficient to tell if it is LALR or not, we need more specific reasons. ~~Like for example~~ We can have any conflicts in between the itemsets itself. For example, we can see that, in itemset I4, there is one shift operation (S8) and another reduce operation (r5) both. But look closely, the shift operation has lookahed operator $, where as the reduce operation has different look ahead a. So, there will be no conflict. Same goes for itemset I7. For all these reasons, we can say it is LALR(1).

OR, YOU CAN JUST CREATE THE PARSING TABLE !!!

LR(0) itemsets:

Follow (S) = { $ }
" (X) = { $ }
" (M) = { a, c }



SLR(1) parsing table

	$	a	b	c	d	S	X	M
I0			S3		S4		I1	I2
I1	acc							
I2		S5						
I3					S7			I6
I4		r5	r5	S8				
I5	r1							
I6				S9				
I7		S10/r5	r5					
I8	r3							
I9	r2							
I10	r4							

Here we can see that we got a cell in the row of I7 which has two types of actions — shift S10 and reduce r5. So, we can clearly see a conflict in that specific cell. For that reason, we can say this grammar is not SLR(1).