Brent Nix   HW 2

1)

STACK	INPUT	ACTION
0	Id*id+id*id$	
0 id4	*id+id*id$	S4
0 F3	*id+id*id$	r6
0 F3 *6	id+id*id$	S6
0 F3 *6 id4	+id*id$	S4
0 F3 *6 F3	+id*id$	r6
0 F3 *6 T8	+id*id$	r5
0 T2	+id*id$	r4
0 T2 +5	id*id$	S5
0 T2 +5 id4	*id$	S4
0 T2 +5 F3	*id$	r6
0 T2 +5 F3 *6	id$	S6
0 T2 +5 F3 *6 id4	$	S4
0 T2 +5 F3 *6 F3	$	r6
0 T2 +5 F3 *6 T8	$	r5
0 T2 +5 T2	$	r4
0 T2 +5 E7	$	r3
0 E1	$	r2
		Accept

2) a.

S0
1) S' -> ·S
2) S -> ·abS
3) S ->·ab

S1
S' -> S·   {$}

S2
S -> ·abS
S ->·ab

S3
S -> ab·S
  -> ab·   {$}
S -> ·abS
  -> ·ab

S4
S -> abS·   {$}
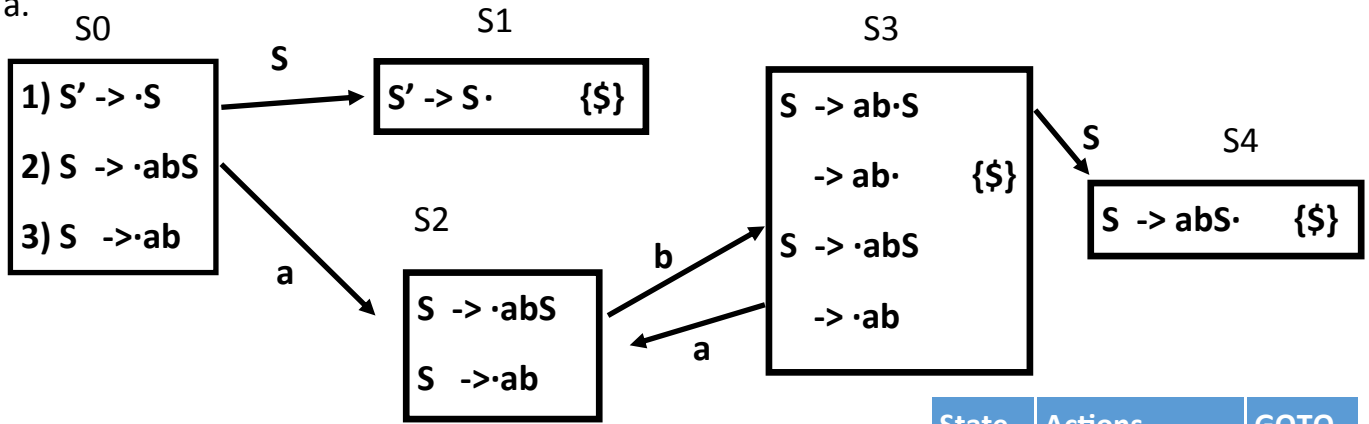
State	Actions			GOTO
	a	b	$	S
S0	S2			1
S1			!	
S2		S3		
S3	S2		r3	4
S4			r2	

This is an SLR

2) b.

S0
1) S' -> ·S
2) S -> ·AaAb
3) S -> ·BbBa
4) A -> ·c
5) B -> ·c

S --> S1
S' -> S·   {$}

A --> S2
S -> A·aAb

a --> S6
S -> Aa·Ab

A --> S5
S -> AaA·b

b --> S9
S -> AaAb·   {$}

S10
A -> c·   {a,b}

B --> S3
S -> B·bBa

b --> S7
S -> Bb·Ba

c --> S11
B -> c·   {a,b}

B --> S8
S -> BbB·a

a --> S12
S -> BbBa·   {$}

c --> S4
A -> c·   {a,b}
B -> c·   {a,b}
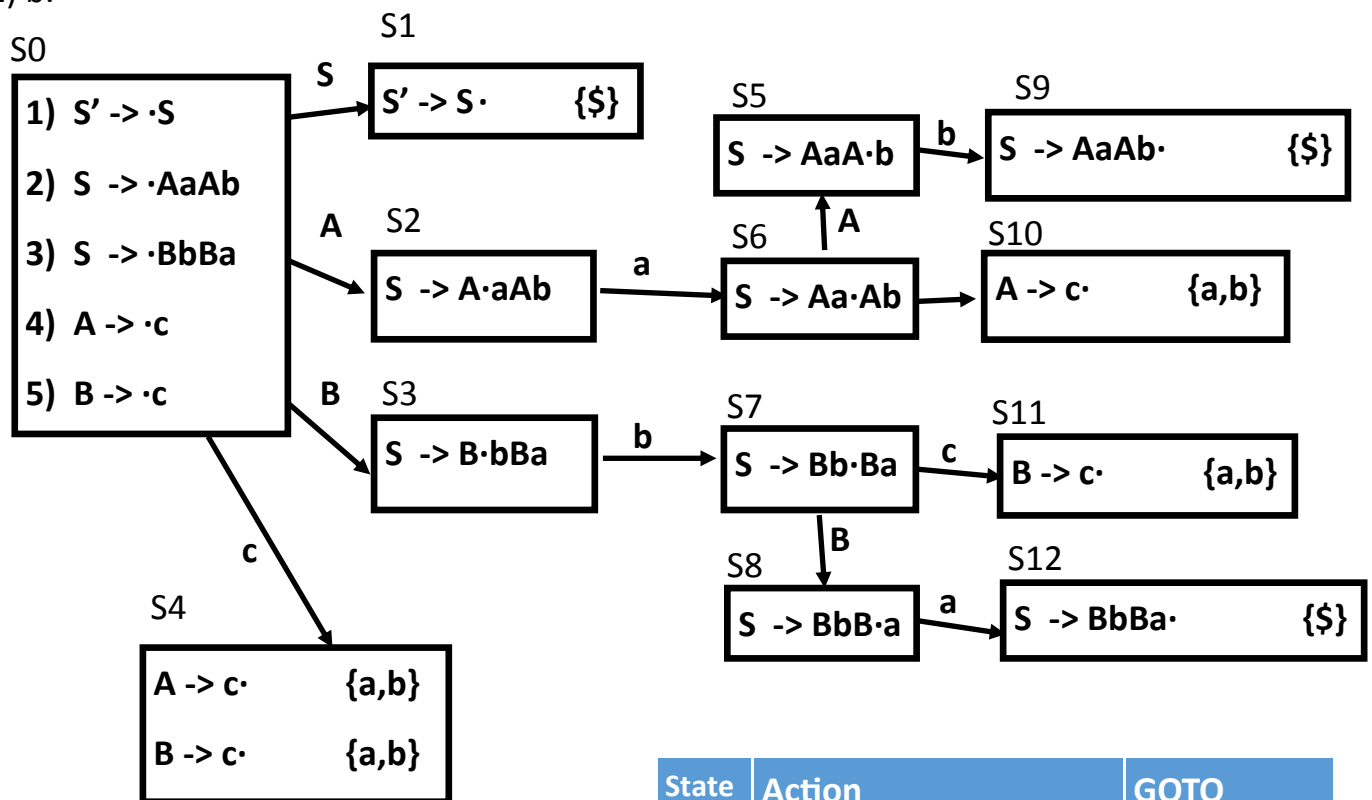
This is not an SLR

State	Action				GOTO		
	a	b	c	$	S	A	B
S0			S4		1	2	3
S1				!			
S2	S6						
S3		S7					
S4	r4/r5						
S5		S9					
S6			S10			5	
S7			S11			8	
S8	S12						
S9				r2			
S10	r4	r4					
S11	r5	r5					
S12				r3			

2) c.

S0
1) S' -> ·S
2) S -> ·ASB
3) S -> ·ab
4) A -> ·a

S1
S' -> S·   {$}

S2
S -> A·SB
S -> ·ASB
S -> ·ab
A -> ·a

S3
S -> a·b
A -> a·   {a,b}

S4
B -> b·   {$}

S5
S -> AS·B
5) B ->·b

S6
S -> ab·   {b,$}

S7
S -> ASB·   {b,$}
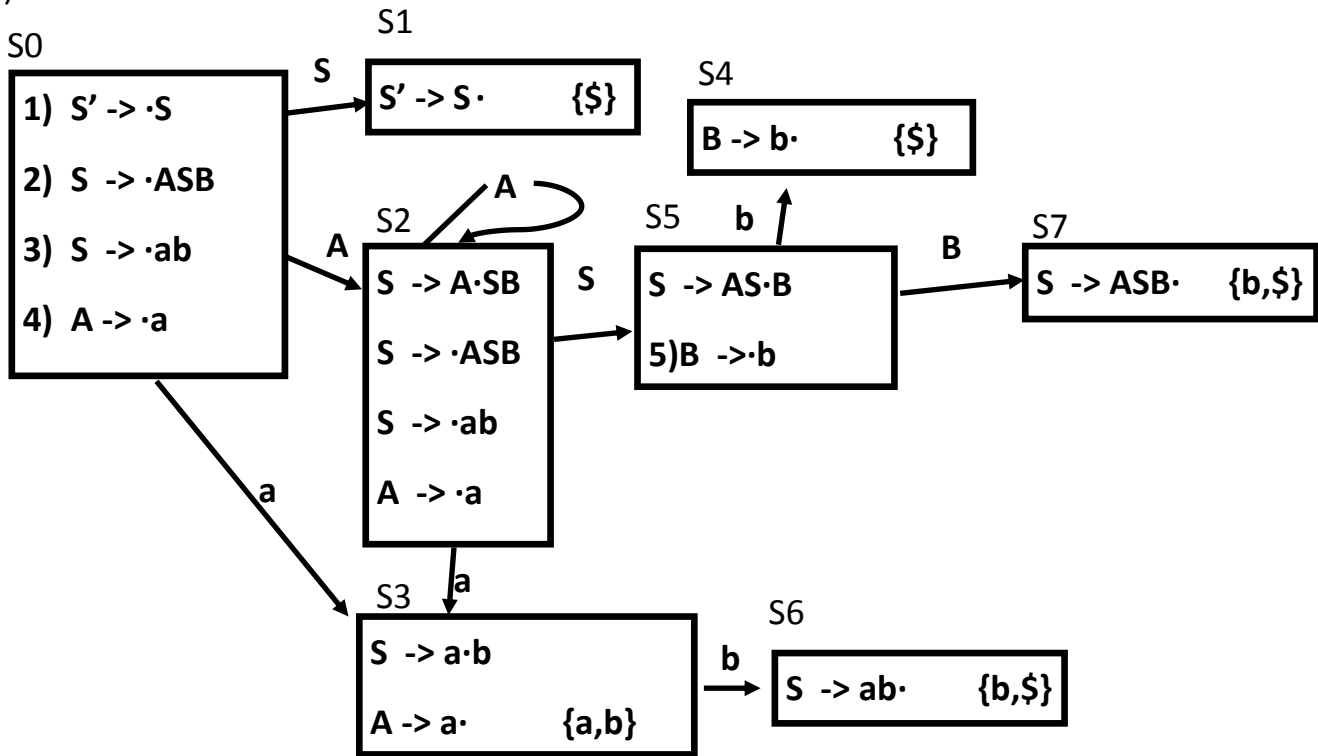
State	Action			GOTO		
	a	b	$	S	A	B
S0	S3			1	2	
S1			!			
S2	S3			5	2	
S3	r4	S6/r4				
S4			r5			
S5		S4				7
S6		r3	r3			
s7		r2	r2			

This is not an SLR

3) First(Es) = { atom, ', (}

Follow(E) = {atom, ', (,), $ }

Predict(Es -> ε) = {),$ }

4)

B -> T B′	Predict(B -> T B′) = { not, (, true, false }
B′ -> or T B′	Predict(B′ -> or T B′) = { or }
B′ -> ε	Predict(B′ -> ε) = {),$ }
T -> C T′	Predict(T -> C T′) = { not, (, true, false }
T′ -> and C T′	Predict(T′ -> and C T′) = { and }
T′ -> ε	Predict(T′ -> ε) = {),$ }
C -> not C	Predict(C -> not C) = { not }
C -> (B)	Predict(C -> (B)) = { (}
C -> true	Predict(C -> true) = { true }
C -> false	Predict(C -> false) = { false }

5)

```
S(){
        If (token == PLUS || token == MINUS || token == A) then{
                if( match(A) ) then
                        return T();
                else return S();
        }
        else  return ERROR;
}
 T(){
        if( token == DIV || token == MULT || token == B ) then{
                If(match(B)
                        return OK;
                else return S();
        }
        else return ERROR;
}
```