A2 sol

Q1 (a) G_1: 1. $P \to S\ \$\$$
2. $S \to (S)\ S$
3. $S \to [S]\ S$
4. $S \to \varepsilon$

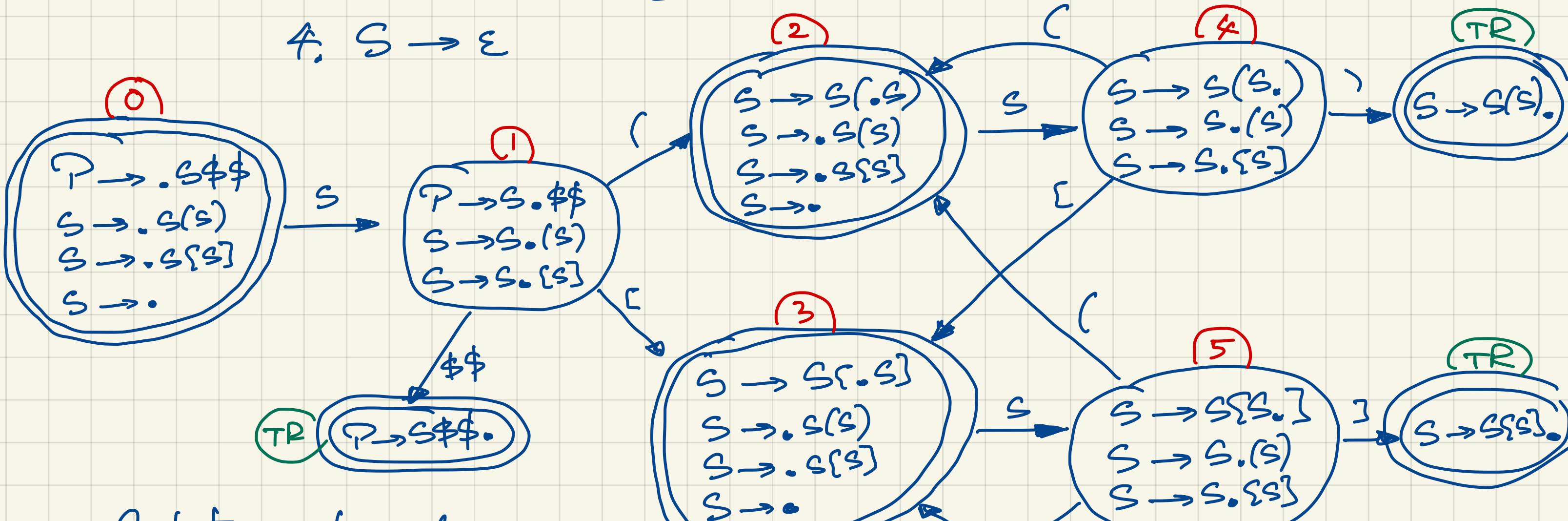
	FIRST	FOLLOW
P	(, [	∅
S	(, [	$\$\$$,),]

Parse table:

nonterm. \ input	(	)	[	]	$\$\$$
P	1		1		1
S	2	4	3	4	4

G_1 is LL(1) because each cell in the parse table has at most one entry.

(b) G_2: 1. $P \to S\ \$\$$
2. $S \to S(S)$
3. $S \to S[S]$ } → not LL(1)
4. $S \to \varepsilon$



-trivial states not used (TR)

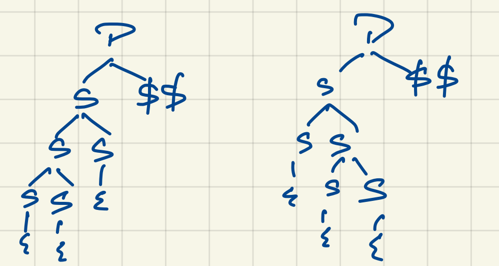
Parse table:

state \ input	S	(	)	[	]	$\$\$$
0	s1	r4	r4	r4	r4	r4
1		s2		s3		b1
2	s4	r4	r4	r4	r4	r4
3	s5	r4	r4	r4	r4	r4
4		s2	b2	s3		
5		s2		s3	b3	

There is no conflict, so G_2 is SLR(1)

(c) G_3: $P \to S\ \$\$$
$S \to SS$
$S \to (S)$
$S \to \varepsilon$

G_3 is ambiguous: therefore not SLR(1)

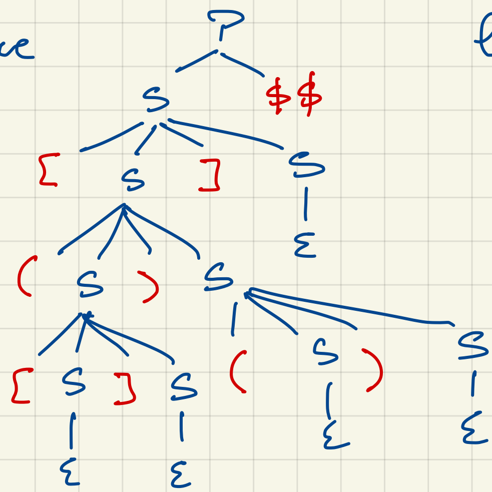


Alternative proof for not SLR(1):

$$\boxed{\begin{array}{l} P \to .S\$\$ \\ S \to .SS \\ S \to .(S) \\ S \to . \end{array}}$$

- shift/reduce conflict on '('
  - shift: $S \to (.S)$
  - reduce: $S \to .$, $'(' \in FOLLOW(S)$

(d) G, parse tree



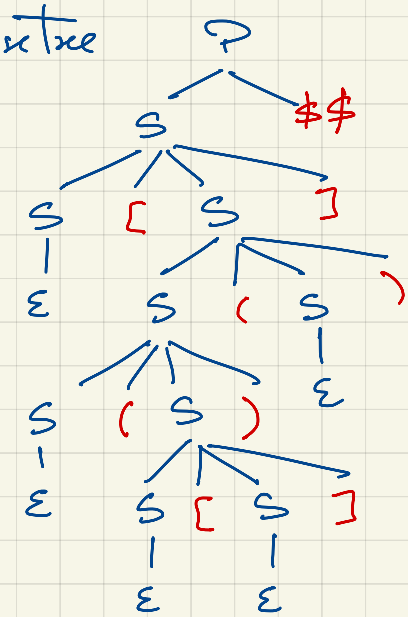
left derivation:

$P \Rightarrow S\$\$ \Rightarrow [S]S\$\$ \Rightarrow [(S)S]S\$\$$
$\Rightarrow [([S]S)S]S\$\$ \Rightarrow [([]S)S]S\$\$$
$\Rightarrow [([])S]S\$\$ \Rightarrow [([])(S)S]S\$\$$
$\Rightarrow [([])()S]S\$\$ \Rightarrow [([])()]S\$\$$
$\Rightarrow [([])()]\$\$$

(e)

Parse stack	Input stream	Comment
P	[([])()]$$	
S$$	[([])()]$$	predict P → S$$
[S]S$$	[([])()]$$	predict S → [S]S
S]S$$	([])()]$$	match [
(S)S]S$$	([])()]$$	predict S → (S)S
S)S]S$$	[])()]$$	match (
[S]S)S]S$$	[])()]$$	predict S → [S]S
S]S)S]S$$	])()]$$	match [
]S)S]S$$	])()]$$	predict S → ε
S)S]S$$	)()]$$	match]
)S]S$$	)()]$$	predict S → ε
S]S$$	()]$$	match
(S)S]S$$	()]$$	predict S → (S)S
S)S]S$$	)]$$	match (
)S]S$$	)]$$	predict S → ε
S]S$$	]$$	match)
]S$$	]$$	predict S → ε
S$$	$$	match]
$$	$$	predict S → ε
		match $$

(f) G_2 parse tree



right derivation:

$P \Rightarrow S\$\$ \Rightarrow S[S]\$\$$

$\Rightarrow S[S(S)]\$\$ \Rightarrow S[S()]\$\$$

$\Rightarrow S[S(S)()]\$\$ \Rightarrow S[S(S[S])()]\$\$$

$\Rightarrow S[S(S[])()]\$\$ \Rightarrow S[S([])()]\$\$$

$\Rightarrow S[([])()]\$\$ \Rightarrow [([])()]\$\$$

(g)

Parse stack	Input stream	Comment
0	[([])()]$$	
0	S[([])()]$$	reduce by S → ε
0S1	[([])()]$$	shift S
0S1[3	([])()]$$	shift [
0S1[3	S([])()]$$	reduce by S → ε
0S1[3S5	([])()]$$	shift S
0S1[3S5(2	[])()]$$	shift (
0S1[3S5(2	S[])()]$$	reduce by S → ε
0S1[3S5(2S4	[])()]$$	shift S
0S1[3S5(2S4[3	])()]$$	shift [
0S1[3S5(2S4[3	S])()]$$	reduce by S → ε
0S1[3S5(2S4[3S5	])()]$$	shift S
0S1[3S5(2	S)()]$$	shift and reduce by S → S[S]
0S1[3S5(2S4	)()]$$	shift S
0S1[3	S()]$$	shift and reduce by S → S(S)
0S1[3S5	()]$$	shift S
0S1[3S5(2	)]$$	shift (
0S1[3S5(2	S)]$$	reduce by S → ε
0S1[3S5(2S4	)]$$	shift S
0S1[3	S]$$	shift and reduce by S → S(S)
0S1[3S5	]$$	shift S
0	S$$	shift and reduce by S → S[S]
0S1	$$	shift S
0	P	shift and reduce by P → S$$

Q2 (a) We use one synthesized attribute s, for the string associated with a node.

$E_1 \to E_2 + T$ ▷ $E_1.s = \text{concat}(E_2.s, T.s, '+')$

$E_1 \to E_2 - T$ ▷ $E_1.s = \text{concat}(E_2.s, T.s, '-')$

$E \to T$ ▷ $E.s = T.s$

$T_1 \to T_2 * F$ ▷ $T_1.s = \text{concat}(T_2.s, F.s, *)$

$T_1 \to T_2 / F$ ▷ $T_1.s = \text{concat}(T_2.s, F.s, '/')$

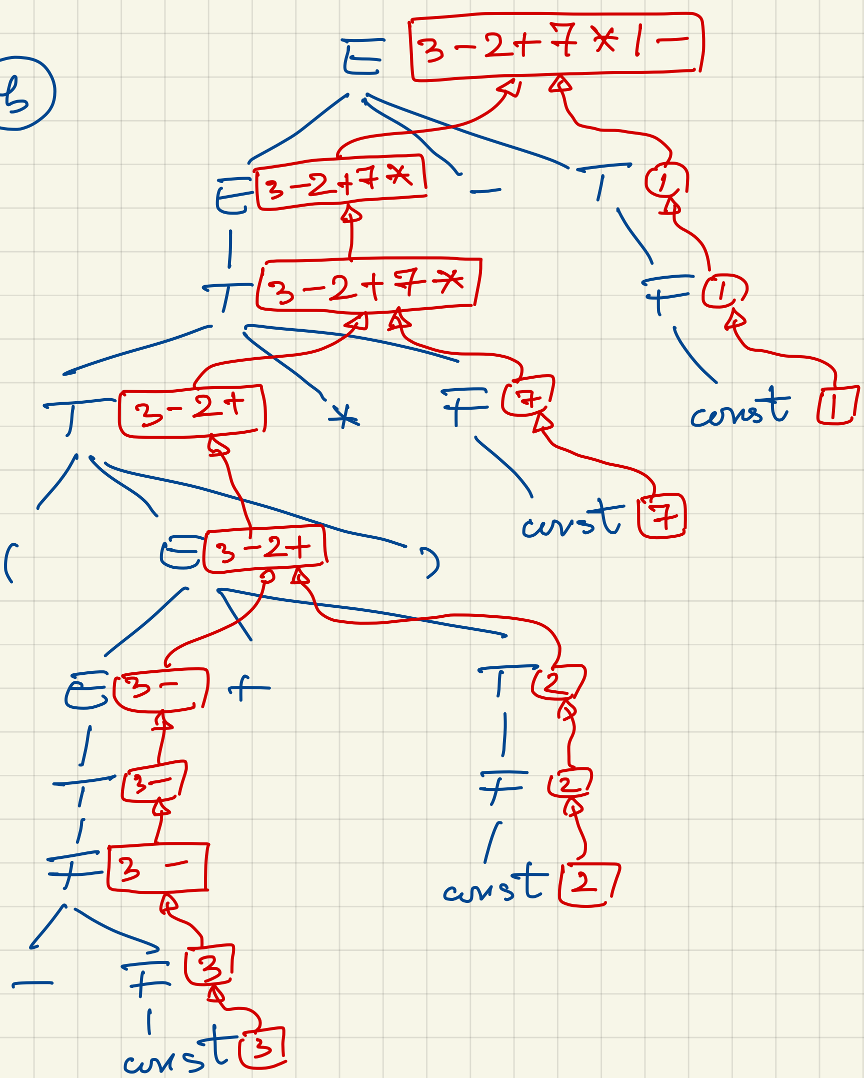
$T \to F$ ▷ $T.s = F.s$

$F_1 \to -F_2$ ▷ $F_1.s = \text{concat}(F_2.s, '-')$

$F \to (E)$ ▷ $F.s = E.s$

$F \to \text{const}$ ▷ $F.s = \text{str}(\text{const})$ (or just const)

(b)



(c) We use one synthesized attribute, s, for string and an inherited one, p, for partial string.

$E \rightarrow T\ TT \qquad \triangleright\ TT.p = T.s \qquad \triangleright\ E.s = TT.s$

$TT_1 \rightarrow +\ T\ TT_2 \qquad \triangleright\ TT_2.p = \text{concat}(TT_1.p, T.s, \text{'+'}) \qquad \triangleright\ TT_1.s = TT_2.s$

$TT_1 \rightarrow -\ T\ TT_2 \qquad \triangleright\ TT_2.p = \text{concat}(TT_1.p, T.s, \text{'-'}) \qquad \triangleright\ TT_1.s = TT_2.s$

$TT \rightarrow \varepsilon \qquad \triangleright\ TT.s = TT.p$

$T \rightarrow F\ FT \qquad \triangleright\ FT.p = F.s \qquad \triangleright\ T.s = FT.s$

$FT_1 \rightarrow *\ F\ FT_2 \qquad \triangleright\ FT_2.p = \text{concat}(FT_1.p, F.s, \text{'*'}) \qquad \triangleright\ FT_1.s = FT_2.s$

$FT_1 \rightarrow /\ F\ FT_2 \qquad \triangleright\ FT_2.p = \text{concat}(FT_1.p, F.s, \text{'/'}) \qquad \triangleright\ FT_1.s = FT_2.s$

$FT \rightarrow \varepsilon \qquad \triangleright\ FT.s = FT.p$

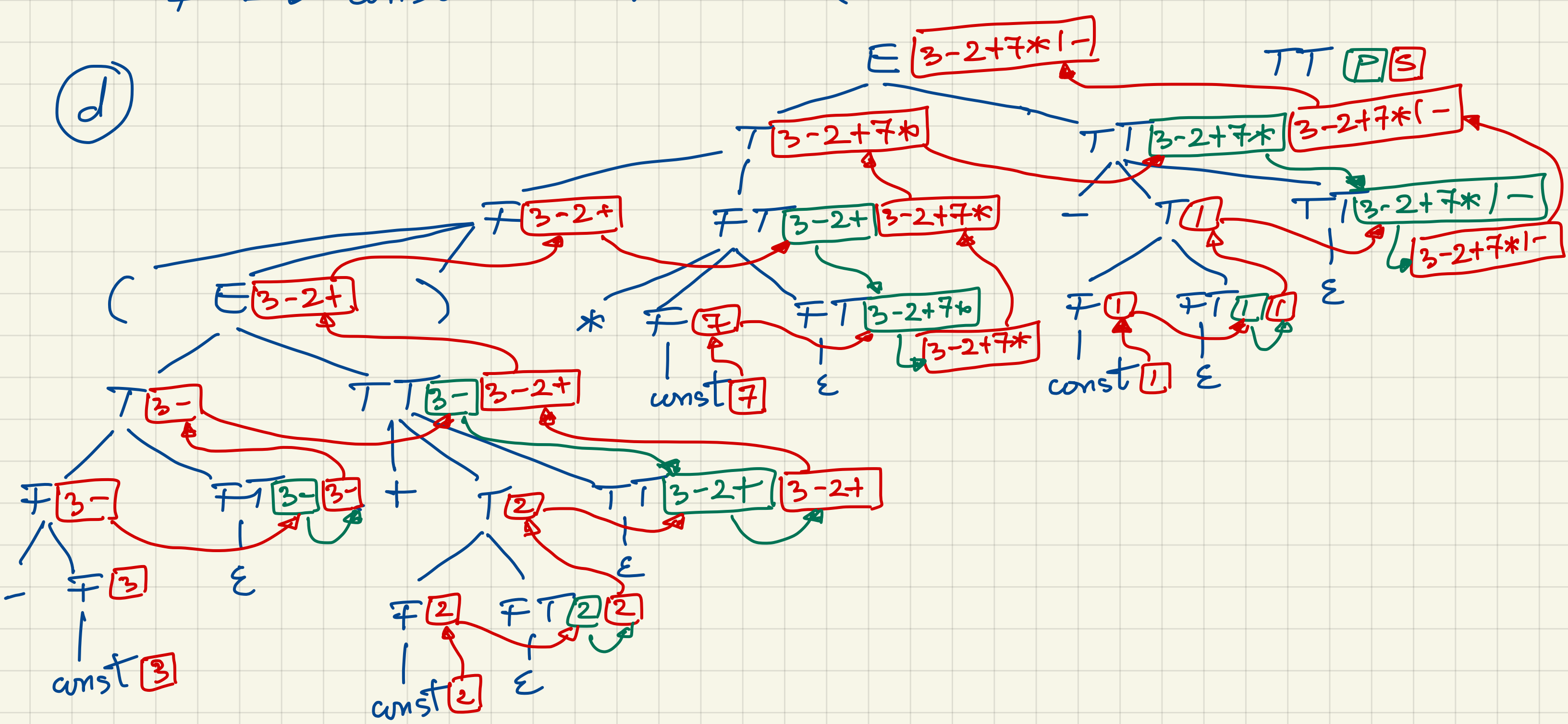
$F_1 \rightarrow -\ F_2 \qquad \triangleright\ F_1.s = \text{concat}(F_2.s, \text{'-'})$

$F \rightarrow (\ E\ ) \qquad \triangleright\ F.s = E.s$

$F \rightarrow \text{const} \qquad \triangleright\ F.s = \text{str}(\text{const})$

$(-3+2)*7-1$

(d)



Q_3 ⓐ Besides the val attribute, we'll use pos for the position of the digit.

Float → Left . Right	▶ Float.val = Left.val + Right.val Right.pos = 0
Left → Digit Left_more	▶ Left.val = Digit.val + Left_more.val Left.pos = Left_more.pos + 1 Digit.pos = Left_more.pos
Left_more → Left	▶ Left_more.val = Left.val Left_more.pos = Left.pos
Left_more → ε	▶ Left_more.val = Left_more.pos = 0
Right → Digit Right_more	▶ Right.val = Digit.val + Right_more.val Right_more.pos = Right.pos − 1 Digit.pos = Right_more.pos
Right_more → Right	▶ Right_more.val = Right.val Right.pos = Right_more.pos
Right_more → ε	▶ Right_more.val = 0
Digit → i	▶ Digit.val = $i \cdot 10^{\text{Digit.pos}}$

ⓑ