Two

Compiler final Exam   Kishor Kumar Andekar

Exercise 3.1, 3.2, 3.3, 3.4 Q4   Derivation, parse tree and ambiguity

(10 pts)   Given the grammar $A \to AA \mid (A) \mid \varepsilon$, show it is ambiguous.

Note: Find one string and provide two different left most derivations OR two different Parse trees. But Do not do both in test.

There exist two different left most derivations for string (())

1) $A \Rightarrow \overset{\text{Rule 1}}{A}A \Rightarrow (A)A \Rightarrow ((A))A \Rightarrow (())A \Rightarrow (())$

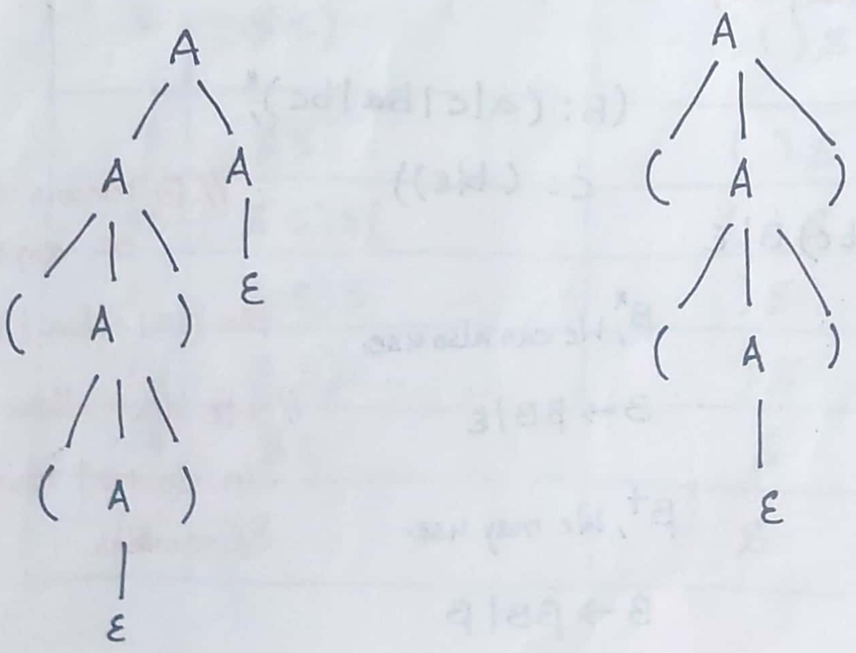
// Rule 1 on first non terminal from Left

2) $A \Rightarrow (A) \Rightarrow ((A)) \Rightarrow (())$

// Rule No 2 twice and Rule no 3 once.

Therefore, it is ambiguous.

We can also show two different Parse trees for the string (())



Exercise 3.20

a) Write a regular expression that generates the same language as the following CF grammar
(10 pts)

$A \rightarrow aA \mid B \mid \varepsilon$

$B \rightarrow bB \mid A$

First, try to list a few strings or find Patterns of strings.

a^*, b^* are in the language

What about $a^*b^*a^*b^*a^*b^* \ldots$?

$(a|b)^*$ — all strings of a's and b's

But not a^*b^*

b) Write a Context Free grammar that generates the same language as following regular Expression
(10 pts)

$(a|c|ba|bc)^* \, (b|\varepsilon)$

$A \rightarrow B\,C$

(B: $(a|c|ba|bc)^*$,
C: $(b|\varepsilon)$)

$B \rightarrow (a|c|ba|bc)\,B \mid \varepsilon$

// β means String of anything

$C \rightarrow b \mid \varepsilon$

β^*, We can also use

$B \rightarrow \beta B \mid \varepsilon$

$\beta = (a|c|ba|bc)$

// We dont allow +, * in Context Free Grammar

β^+, We may use

$B \rightarrow \beta B \mid \beta$

// β can be any sequence of terminals.

Exercise 4.5, 4.6 (10 pts)

Show the action of LL(1) parse that uses the table below to recognize string (())().

M[N,T]	(	)	$
S	S→(S)S	S→ε	S→ε

SNo	Parsing Stack	Input	Action
1.	$S	(())()$	S→(S)S
2.	$S)S(	(())()$	Match
3.	$S)S	())()$	S→(S)S
4.	$S)S)S(	())()$	Match
5.	$S)S)S	))()$	S→ε
6.	$S)S)	))()$	Match
7.	$S)S	)()$	S→ε
8.	$S)	)()$	Match
9.	$S	()$	S→(S)S
10.	$S)S(	()$	Match
11	$S)S	)$	S→ε
12	$S)	)$	Match
13	$S	$	S→ε
14.	$	$	Accept

Exercise 4.8 (10 pts)

a) Remove Left Recursion

$\underbrace{\text{lexp-seq}}_{A} \rightarrow \underbrace{\text{lexp-seq}}_{A}\ \underbrace{\text{lexp1}}_{\alpha} \mid \underbrace{\text{lexp2}}_{\beta}$

From $A \rightarrow A\alpha \mid \beta$

To $A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \mid \varepsilon$

$\text{lexp-seq} \rightarrow \text{lexp2}\ \text{lexp-seq}'$

$\text{lexp-seq}' \rightarrow \text{lexp1}\ \text{lexp-seq}' \mid \varepsilon$

(b) Remove Left recursion, Be careful with the case that a

(15 pts) substitution is needed first

// Left recursion means Left associativity

// put A first because A is start symbol.

$A \rightarrow Ba \mid Aa \mid c$

$B \rightarrow Bb \mid Ab \mid d$

Rewrite A Rule: $A \rightarrow A\underbrace{a}_{\alpha} \mid \underbrace{(Ba \mid c)}_{\beta}$

// Anything behind A is α

1. Remove Left recursion for A rules

$A \rightarrow (Ba \mid c)A' \Rightarrow A \rightarrow BaA' \mid cA'$

$A' \rightarrow aA' \mid \varepsilon$

2. Substitution (applying $A \rightarrow BaA' \mid cA'$ in B Rules)

$A \rightarrow BaA' \mid cA'$

$A' \rightarrow aA' \mid \varepsilon$

$B \rightarrow Bb \mid BaA'b \mid cA'b \mid d$ // Compare as $A \rightarrow A\alpha \mid \beta$)

$B \rightarrow B\ (\underbrace{b \mid aA'b}_{\alpha}) \mid \underbrace{cA'b \mid d}_{\beta}$

3. Remove Left recursion for B rules

$A \rightarrow BaA' \mid cA'$

$A' \rightarrow aA' \mid \varepsilon$

$B \rightarrow (cA'b \mid d)B' \Rightarrow B \rightarrow cA'bB' \mid dB'$

$B' \rightarrow (b \mid aA'b)B' \mid \varepsilon \Rightarrow B' \rightarrow bB' \mid aA'bB' \mid \varepsilon$

Exercise 4.9 (10 pts)

(a) Left Factor the grammar

Rules: $A \rightarrow \alpha\beta \mid \alpha\delta$

$A \rightarrow \alpha A'$

$A' \rightarrow \beta \mid \delta$

$$\underbrace{\text{lexp-seq}}_{A} \rightarrow \underbrace{\text{lexp}}_{\alpha}\ \underbrace{\text{lexp-seq}}_{\beta} \mid \underbrace{\text{lexp}}_{\alpha}\ \delta$$

$\text{lexp-seq} \rightarrow \text{lexp lexp-seq}'$

$\text{lexp-seq}' \rightarrow \text{lexp-seq} \mid \varepsilon$

SLR(1) Parsing

For these two CFG's, Solve following four problems

1. $S \rightarrow S(S) \mid \varepsilon$

2. $S \rightarrow (A) \mid AS$, $A \rightarrow S \mid \varepsilon$

1. Convert the grammar to a augmented Grammar with a new (10pts) Start symbol and List all LR(0) items.

Augmented Grammar

$S' \rightarrow S$, $S \rightarrow S(S) \mid \varepsilon$

LR(0) Items:

$S' \rightarrow \cdot S$

$S' \rightarrow S \cdot$

$S \rightarrow \cdot S(S)$

$S \rightarrow S \cdot (S)$

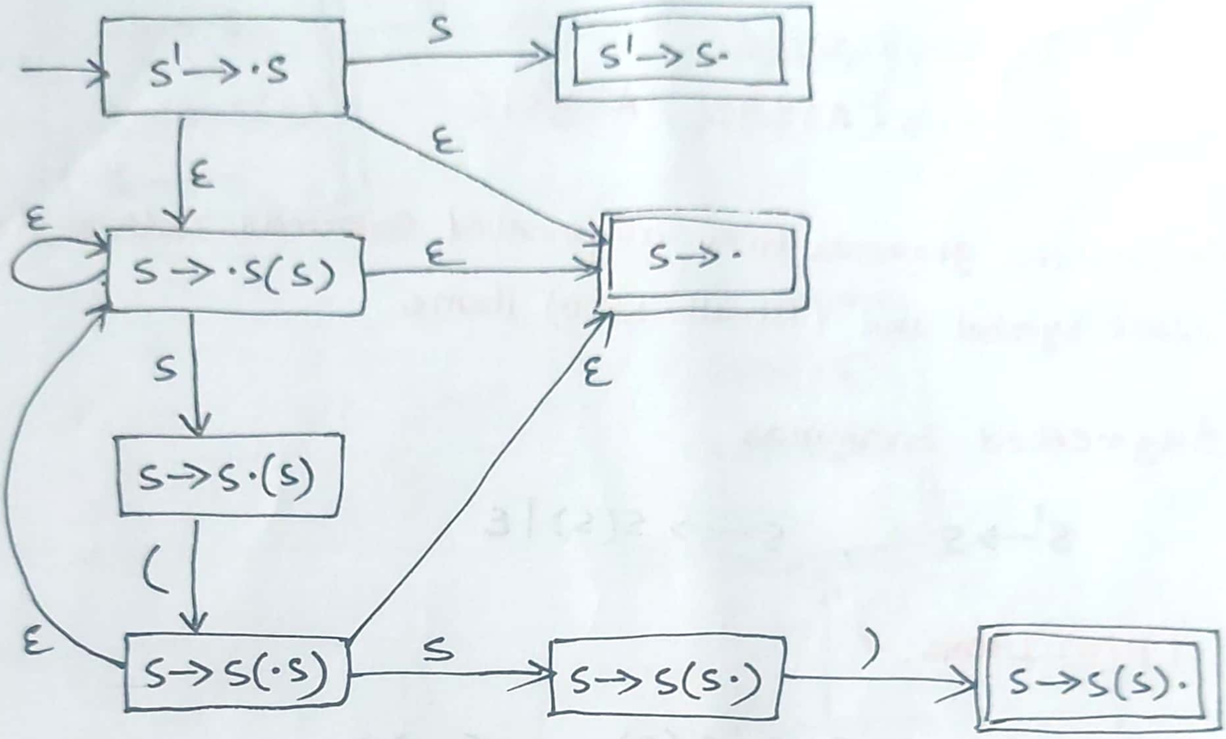
$S \rightarrow S(\cdot S)$

$S \rightarrow S(S \cdot)$

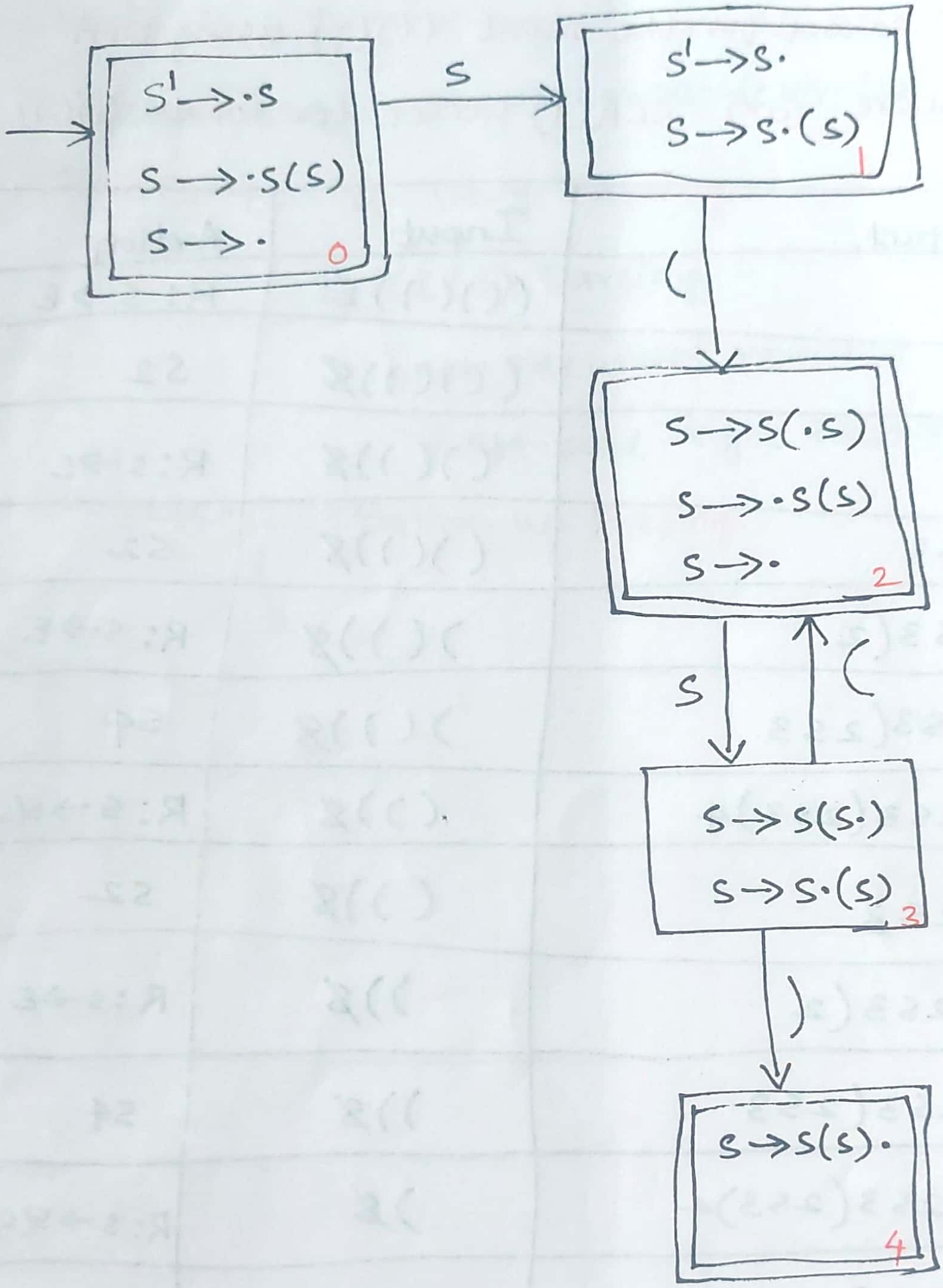
$S \rightarrow S(S) \cdot$

$S \rightarrow \cdot$

2) Construct NFA of LR(0) items for the grammar



3) Construct the DFA from NFA



4. Show Parsing stack and the action (shift and reduce) of an SLR(1) parser for the input (()()) using DFA Constructed above. (or) SLR(1) parser for Input ()())

SNo	Parsing stack	Input	Action
1.	$0	(()())$	R: S→ε
2.	$0S1	(()())$	S2
3.	$0S1(2	()())$	R: S→ε
4.	$0S1(2S3	()())$	S2
5.	$0S1(2S3(2	)())$	R: S→ε
6.	$0S1(2S3(2S3	)())$	S4
7.	$0S1(2S3(2S3)4	())$	R: S→S(S)
8.	$0S1(2S3	())$	S2
9	$0S1(2S3(2	))$	R: S→ε
10.	$0S1(2S3(2S3	))$	S4
11	$0S1(2S3(2S3)4	)$	R: S→S(S)
12.	$0S1(2S3	)$	S4
13.	$0S1(2S3)4	$	R: S→S(S)
14.	$0S1	$	R: S'→S
15.	$0S'	$	Accept

৬০

Final 1 Kishor Kumar Andekar

1. Regular Expression $S \rightarrow aSA \mid B$

$A \rightarrow c$

$B \rightarrow Bb \mid b$

$S \rightarrow aSA \Rightarrow aaSAA \Rightarrow aabcc$ or $aaBbcc$

$aabbcc$

$S \rightarrow B \Rightarrow Bb \Rightarrow bb$

$S \rightarrow asA \Rightarrow aBA \Rightarrow aBbA \Rightarrow abbc$

There is no proper Regular Expression

2. RE $(a|b)(a|cb)^+(c|\varepsilon)$

$A \rightarrow BCD$

$B \rightarrow a|b$ $\qquad \beta = (a|cb)$

$C \rightarrow (a|cb)c \mid (a|cb)$ $\qquad \beta^* = \beta B|\varepsilon$

$D \rightarrow c|\varepsilon$

$\beta^+ = \beta B \mid \beta$

3. Left Recursion

exp → exp addop term | term

$A \rightarrow A\alpha \mid \beta$

exp → term exp' $\qquad A \rightarrow \beta A'$

exp' → addop term exp' | ε $\qquad A' \rightarrow \alpha A' \mid \varepsilon$

4. Left factor

lexp → atom list | atom term1 term2

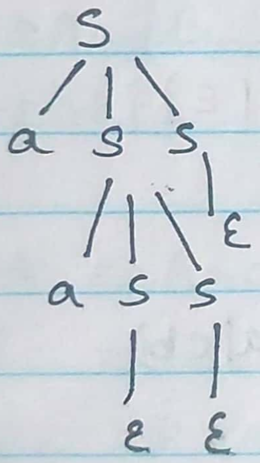
lexp → atom lexp' $\qquad A \to \alpha A'$

lexp' → list | term1 term2 $\qquad A' \to \beta | \delta$

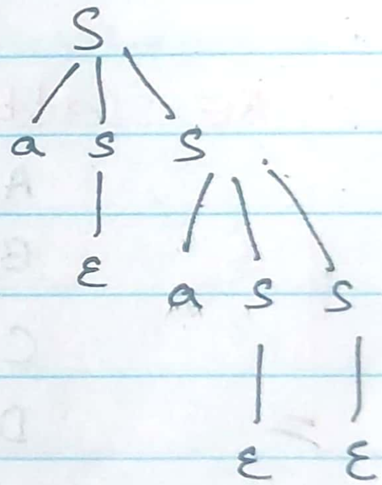
5. CF Grammar $S \Rightarrow aSS | \varepsilon$ String aa

Parse Tree

1)



2)



Ambiguous.

6. LL(1) ()()

Sno	Parsing stack	Input	Action
1.	$ S	()()$	S→(S)S
2.	$ S)S(	()()$	match
3.	$ S)S	)()$	S→ε
4.	$ S)	)()$	match
5.	$ S	()$	S→(S)S
6.	$ S)S(	()$	match
7.	$ S)S	)$	S→ε
8.	$ S)	)$	match
9.	$ S	$	S→ε
10.	$	$	Accept

SLR

$S \rightarrow (AS) | \varepsilon$

$A \rightarrow S | \varepsilon$

Augmented Grammar $S' \rightarrow S$

$S \rightarrow (AS) | \varepsilon$

$A \rightarrow S | \varepsilon$

LR(0) Items

$S' \rightarrow \cdot S$ $S \rightarrow \cdot(AS)$ $S \rightarrow \cdot$ $A \rightarrow \cdot S$ $A \rightarrow \cdot$

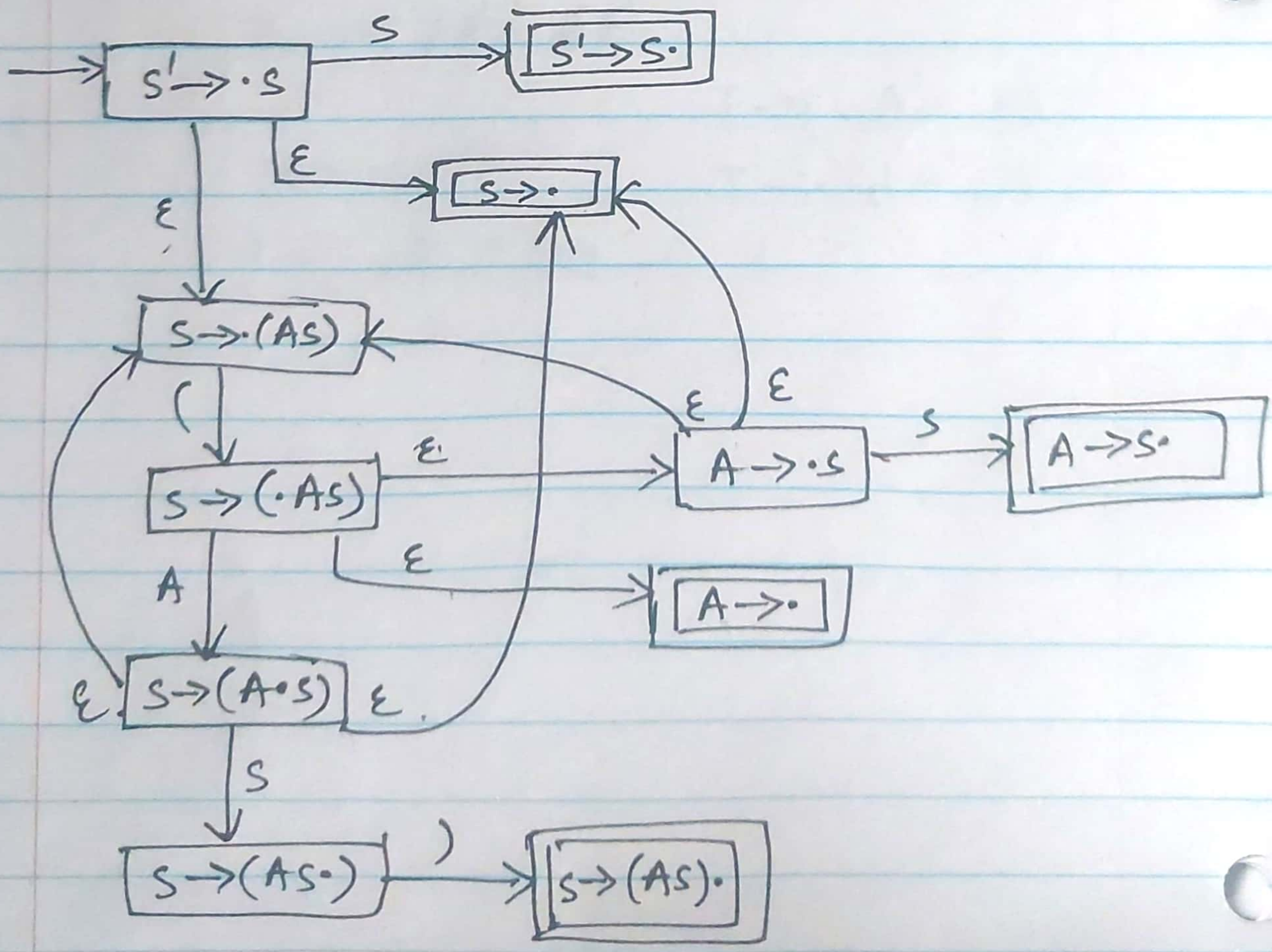
$S' \rightarrow S\cdot$ $S \rightarrow (\cdot AS)$ $A \rightarrow S\cdot$

$S \rightarrow (A\cdot S)$

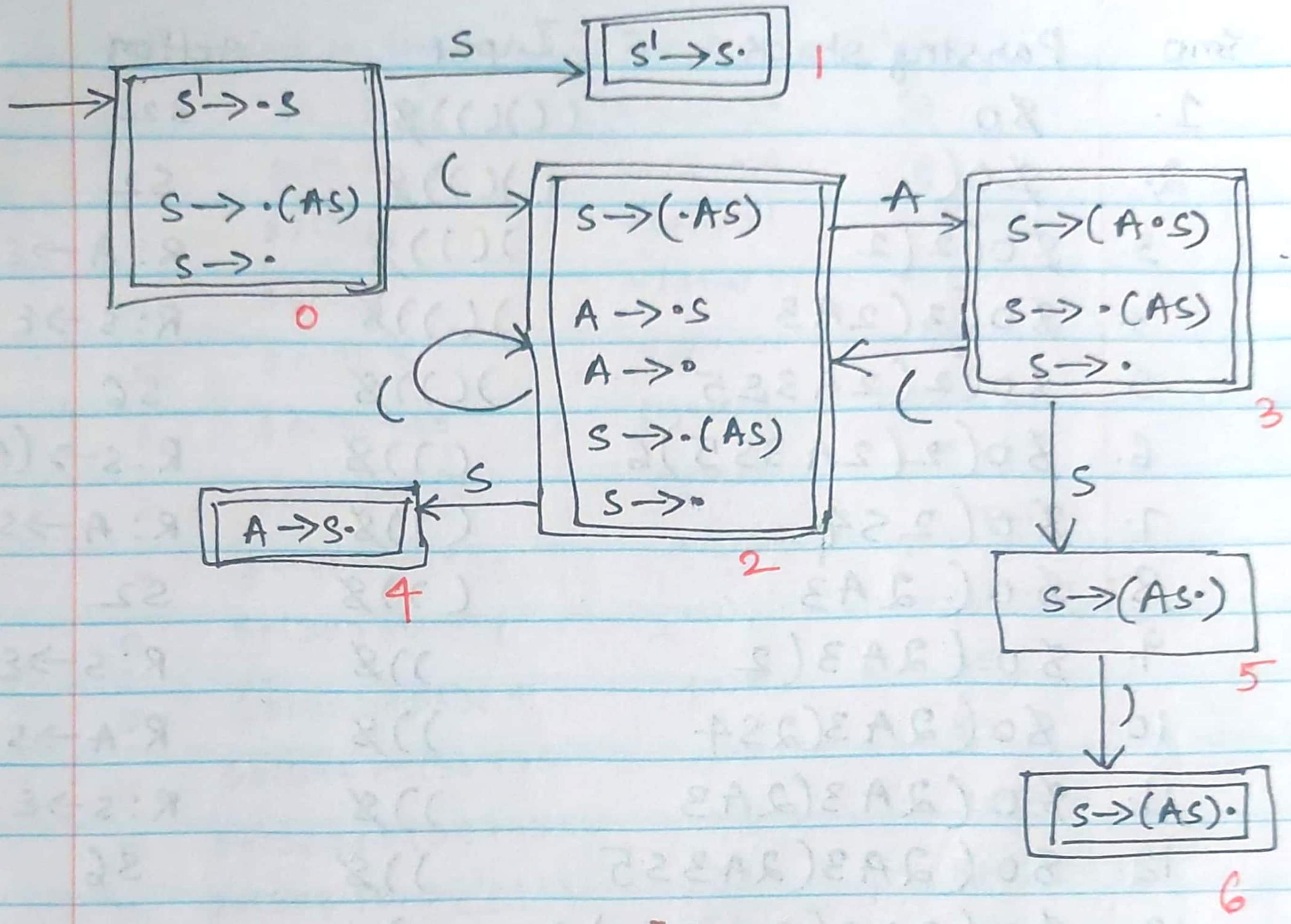
$S \rightarrow (AS\cdot)$

$S \rightarrow (AS)\cdot$

6 NFA



DFA



SLR (()())

Sno	Parsing stack	Input	Action
1.	$0	(()())$	S2
2.	$0(2	()())$	S2
3.	$0(2(2	)())$	R: A→ε
4.	$0(2(2A3	)())$	R: S→ε
5.	$0(2(2A3S5	)())$	S6
6.	$0(2(2A3S5)6	())$	R: S→(AS)
7.	$0(2S4	())$	R: A→S
8.	$0(2A3	())$	S2
9.	$0(2A3(2	))$	R: S→ε
10.	$0(2A3(2S4	))$	R: A→S
11.	$0(2A3(2A3	))$	R: S→ε
12.	$0(2A3(2A3S5	))$	S6
13.	$0(2A3(2A3S5)6	)$	R: S→(AS)
14.	$0(2A3S5	)$	S6
15.	$0(2A3S5)6	$	R: S→(AS)
16.	$0S1	$	R: S'→S
17.	$0S'	$	Accept

LMD: $S \to aSS \mid \varepsilon$ $\underline{aa}$

1) $S \Rightarrow aSS \Rightarrow aa\underline{SSS} \Rightarrow aaSS \Rightarrow aaS \Rightarrow aa$

2) $S \Rightarrow aSS \Rightarrow aS \Rightarrow aaSS \Rightarrow aaS \Rightarrow aa$

⑦ LR: $T \to Aa \mid b$

$A \to Ac \mid Td \mid \varepsilon$

① $A \to A\underbrace{c}_{\alpha} \mid \underbrace{(Td \mid \varepsilon)}_{\beta}$

$A \to (Td \mid \varepsilon)A'$

$A' \to cA' \mid \varepsilon$

② sub A in T Rules

$T \to (Td \mid \varepsilon)A'a \mid b$

$T \to T\underbrace{dA'a}_{\alpha} \mid \underbrace{A'a \mid b}_{\beta}$

$T \to (A'a \mid b)T'$

$T' \to dA'aT' \mid \varepsilon$

ⓐ $(a|b)(a|cb)^{+}(c|\varepsilon)$

$A \to BCD$

$B \to (a|b)$

$C \to (a|cb)C \mid (a|cb)$

$D \to (c|\varepsilon)$