

b) LR table

State	ACTION				VOTO					
	a	b	c	d	*	S'	s	A	B	C
0	s_2							1	3	
1										
2		s_8	s_b	s_7						
3		s_{10}		s_h	s_7			5	4	
4									9	
5			s_{12}							
6				s_m	s_b				13	
7						r_8				
8			s_8	s_{17}					16	
9							r_2			
10		r_4		r_4	r_4					
11				s_{11}	s_7				18	
12		r_3		r_3	r_3					
13			r_6			r_7				
14				s_{14}	s_{15}				19	
15			r_8			r_8				
16			r_5							
17				s_{21}	s_{22}				20	
18							r_7			
19			r_1				r_7			
20			r_6							
21				s_{21}	s_{22}				23	
22			r_8							
23			r_7							

a.) Input: abbbcccd

Trace

Step	Stack	Input	Action
1	0	a bbb cc d \$	S2
2	0 a z	bbb cc d \$	S3
3	0 a z b 8	b b cc d \$	S8
4	0 a z b 8 b 8	b c c d \$	S8
5	0 a z b 8 b b 8 b 8	c c d \$	S17
6	0 a z b 8 b 8 b 8 C 17	C 17	S21
7	0 a z b 8 b 8 b 8 C 17 C 21	C 21	S22
8	0 a z b 8 b 8 b 8 C 17 C 21 b 22	b 22	

b.)

trace

Step	Stack	Input	Action
1	0	accd \$	S2
2	0 a z	cc d \$	S6
3	0 a z c b	C d \$	S14
4	0 a z c b c 14	C 14	S15
5	0 a z c b c 14 d 15	D 15	R8
6	0 a z c b c 14 C	C	GOTO 14
7	0 a z c b c 14 C 14	C 14	R7
8	0 a z c b C	C	GOTO 13
9	0 a z c b C 13	C 13	R9
10	0 a z C	C	GOTO 4
11	0 a z C 4	C 4	R1
12	0 S	S	GOTO 1
13	0 S 1	1	accept

input: a c d b a a c d

trace			
Step	Stack	INPUT	Action
1	O	a c d b a a c d \$	S ₂
2	O a ₂	c d b a a c d \$	S ₆
3	O a ₂ c b	d b a a c d \$	S ₁₅
4	O a ₂ c b d 15	b a a c d \$	F ₈
5	O a ₂ c b l	b a a c d \$	goto 13
6	O a ₂ c b c 13	b a a c d \$	F ₆
7	O a ₂ B	b a a c d \$	goto 5
8	O a ₂ B 5	b a a c d \$	S ₁₂
9	O a ₂ B 5 b 12	a a c d \$	F ₃
10	O A	a a c d \$	goto 3
11	O A 3	a a c d \$	S ₁₀
12	O A 3 a 10	a c d \$	F ₄
13	O A	a c d \$	goto 3
14	O A 3	a c d \$	S ₁₀
15	O A 3 a 10	c d \$	F ₄
16	O A	c d \$	goto 3
17	O A 3	c d \$	S ₁₁
18	O A 3 C 11	d	S ₇
19	O A 3 C 11 d 7	\$	F ₈
20	O A 3 C 11 C	\$	goto 18
21	O A 3 C 11 C 18	\$	G
22	O A 3 C	\$	goto 4
23	O A 3 C 4	\$	F ₂
24	O S	\$	goto 1
25	O S 1	\$	accept

Input: acdbd

d.)

trace

Step	Stack	Input	Action
1	0	a c d b d #	S ₂
2	0a2	c d b d #	S _b
3	0a2cb	d b d #	S ₁₅
4	0a2cb15	b d #	R ₈
5	0a2cbC	b d #	goto 13
6	0a2cb13	b d #	R ₆
7	0a2B	b d #	goto 5
8	0a2B5	b d #	S ₁₂
9	0a2B5 b12	d #	R ₅
10	0A	d #	goto 3
11	0A3	d #	S ₇
12	0A3d7	#	R ₈
13	0A3c	#	goto 4
14	0A3c4	#	R ₅
15	0S	#	goto 1
16	0S1	#	ACCEPT

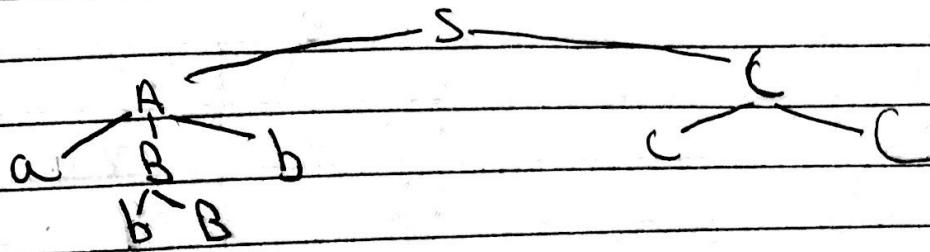
input: abCdbad

e.)

trace

Step	Stack	Input	Action
1	0	abCdbad	S_2
2	0a2	bCd bad	S_8
3	0a2b8	Cbdad	S_{17}
4	0a2b8C17	d bad	S_{22}
5	0a2b8C17d22	bad	r_8
6	0a2b8C17C	bad	goto 20
7	0a2b8C17C20	bad	r_6
8	0a2b8B	bad	goto 16
9	0a2b8B16	bad	r_5
10	0a2B	bad	goto 5
11	0a2B5	bad	S_{17}
12	0a2B5b12	ad	r_3
13	0A	ad	goto 3
14	0A3	ad	S_{10}
15	0A3a10	a	r_4
16	0A	a	goto 3.
17	0A3	a	S_7
18	0A3d7	d	r_8
19	0A3C	\$	goto 4
20	0A3C4	\$	r_2
21	0S	\$	goto 1
22	0S1	\$	accept

3.) a) input: abBbCC



Phrases: abBbCC, abBb, bB, CC

Simple phrases: bB, CC

Handle: bB

Right most derivation

S

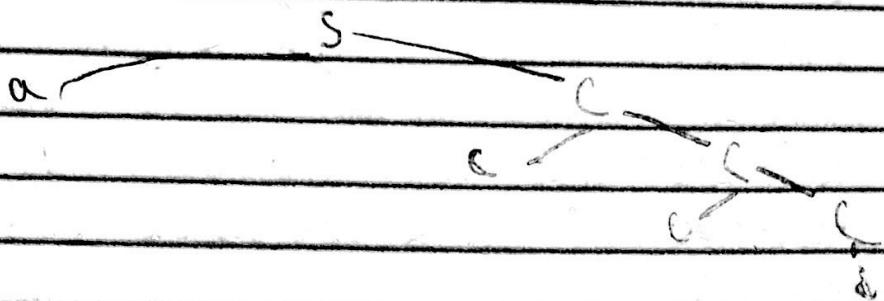
AC

A:CC

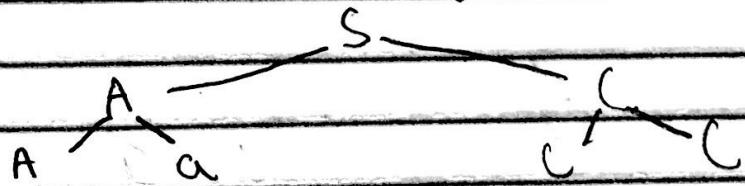
abBbCC

abBbCC

b) Input: accld



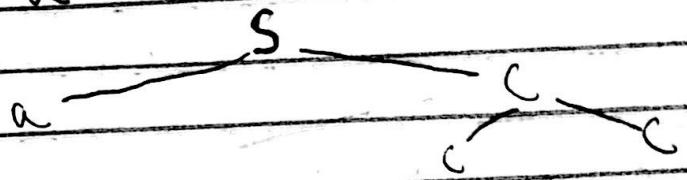
If we begin by selecting a C, to begin, we won't be able to produce 2 lower case c's and a capital C, with a d at the end. As seen, if we create 2 lower case c's then a d, we'll be missing the capital C. We can't use that last capital C to produce another c b/c we'd then have 3 lower case c's.



If we begin by selecting A C, we can't do anything with the first non terminal, if we make A, A a, we'll have too many a's in our string, and if we make it a B b, we'll have a lowercase b, which isn't in our input string.

Conclusion: This string is not in our grammar

(ii) Input: aCbbaac



- We can't begin with "a C", because we won't ever be able to derive a lower case b from the non-terminal C.
- We can't begin with "A C", because if we convert A into "a B b", we won't be able to get a capital 'C' into the 2nd spot for our string, and if we convert A into "A a", we run into a similar issue with getting a capital C into that 2nd spot.

Conclusion! This string is not in the grammar

d.) INPUT: a c d a b d

- We can't begin with $S \rightarrow a C$, because we won't ever be able to derive a lowercase b from the non-terminal C.
- When we take $A C$, we can't convert A into "A a", because we will not be able to have a lowercase b in front of our d at the end (a c d a b d). But if we convert A into "a B b", we won't be able to get the lowercase a in the 4th spot (a c d a b d), because it's impossible to derive lowercase 'a' from the Non-terminal B.

Conclusion: This string is not in our grammar.

e) Input: abCbad

- we can't start with "a C", because then we'll never be able to derive a lowercase 'b'. If we start with "A C", we'll have to convert the non-terminal 'C' into d, because our string doesn't have a lowercase c, so we can't do $C \rightarrow cC$.
- The other problem we run into is that we cannot create a capital 'C' from nonterminal A, because we would then create a lowercase 'c'. B can be converted into CL, and C can be converted into CL. So as seen, we can't create 'C' without creating a lowercase c, which isn't in our string. Thus, this string isn't in our grammar.

Conclusion: This string is not in our grammar.