# CSE103 Homework 9

Isai Lopez Rodas

ilopezro | 1605542

 $March\ 12,\ 2020$ 

Language being referenced in this homework assignment comes from Canvas under Files  $\rightarrow$  Handouts  $\rightarrow$  CKY Algorithm Notes.pdf

Language A:

$$S \to AB \mid BA \mid SS \mid AC \mid BD$$

$$A \rightarrow a$$

$$B \to b$$

$$C \to SB$$

$$D \to SA$$

## Problem #1: Is string $babaab \in L(A)$ ?

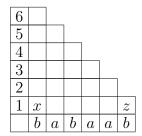


Table 1: Initial Unfilled Table

I will denote each row as an array row[x][y], where x is the row number and y is the index of that array. For example, row[1][0] refers to the bottom left corner marked with an x while row[1][5] is marked with z.

row[1] will contain all possible derivations for the letter right below it.

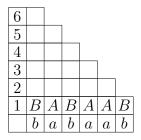


Table 2: Row 1 filled

row[2] contains all possible derivations for substrings of length 2.

• row[2][0] will look at string ba. ba contains substrings b and a, which are derived from B and A in that order. The cartesian product is BA. BA is found in grammar A through S. Therefore, row[2][0] = S.

- row[2][1] will look at string ab. ab contains substrings a and b, which are derived from A and B in that order. The cartesian product is AB. AB is found in grammar A through S. Therefore, row[2][1] = S.
- row[2][2] will look at string ba. ba contains substrings b and a, which are derived from B and A in that order. The cartesian product is BA. BA is found in grammar A through S. Therefore, row[2][2] = S.
- row[2][3] will look at string aa. aa contains substrings a and a, which are derived from A and A in that order. The cartesian product is AA. AA is found in grammar A through  $\emptyset$ . Therefore, row[2][3] =  $\emptyset$ .
- row[2][4] will look at string ab. ab contains substrings a and b, which are derived from A and B in that order. The cartesian product is AB. AB is found in grammar A through S. Therefore, row[2][4] = S.

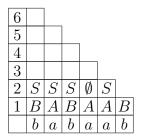


Table 3: Row 2 filled

row[3] will follow similar procedure as row[2]. We will look at substrings of length 3.

• row[3][0] will look at substring bab. This string can be further split up into two substrings,  $s_1 = b$ , ab and  $s_2 = ba$ , b. I will look at  $s_1$  first. b can be derived from B and ab can be derived from S. The cartesian product is BS. BS is not found in grammar A. We will now look at  $s_2$ . ba is derived from S and S is derived from S. Cartesian product

SB can be found in grammar  $\mathcal{A}$  through C. Therefore, since  $s_1$  cannot be derived, but  $s_2$  can, row[3][0] = C.

- row[3][1] will look at substring aba. This string can be further split up into two substrings, s<sub>1</sub> = a, ba and s<sub>2</sub> = ab, a. I will look at s<sub>1</sub> first. a can be derived from A and ba can be derived from S. Cartersian product AS is not found in grammar A. I will look at s<sub>2</sub> now. ab is derived from S and a is derived from a. Cartesian product SA is derived through D. Therefore, since s<sub>1</sub> cannot be derived, but s<sub>2</sub> can, row[3][1] = D.
- row[3][2] will look at substring baa. This string produces  $s_1 = b$ , aa and  $s_2 = ba$ , a. There are no derivations for substring aa in grammar  $\mathcal{A}$  as denoted in row[2][3]. For substring  $s_2$ , we can obtain ba from S and a from A. The cartesian product is SA, which can be obtain through D. Since  $s_1$  does not contain a derivation, we do not take that into account and look onto at  $s_2$ 's derivation. Therefore, row[3][2] = D.
- row[3][3] will look at substring aab. This string produces  $s_1 = a, ab$  and  $s_2 = aa, b$ . We can discard taking into consideration  $s_2$  because it does not have a derivation.  $s_1$  can be derived from A and S. The cartesian product is AS, which is not in the grammar A. Therefore, there are no derivations for any of these substrings so row[3][3] is  $\emptyset$ .

6						
5						
4						
3	$\overline{C}$	D	D	Ø		
2	S	S	S	Ø	S	
1	B	A	B	A	A	B
	b	a	b	a	a	b

Table 4: Row 3 filled

#### row[4] will look at substrings of length 4.

- row[4][0] will look at substring baba. This string can be broken down into  $s_1 = b, aba, s_2 = ba, ba$ , and  $s_3 = bab, a$ . For string  $s_1, b$  is derived from B and aba is derived from D. Cartersian product is BD. This can be obtained through S. For string  $s_2$ , ba can be obtained from S. Since we have two of ba, the cartesian product is SS, which can be obtained through S. For string  $s_3$ , bab can be obtained through C and C are obtained through C and C are obtained through C and C are obtained through the grammar at all. Therefore, since  $c_1$  and  $c_2$  can both be derived by  $c_3$ , we say that  $c_3$  and  $c_4$  and  $c_5$  can be obtained through  $c_5$  and  $c_6$  can be obtained through the grammar at all.
- row[4][1] will look at substring abaa. This string can be split up into:  $s_1 = a, baa, s_2 = ab, aa$ , and  $s_3 = aba, a$ . For string  $s_1, a$  is obtained through A and baa is obtained through D. Cartersian product AD is not in the grammar. For string  $s_2$ , aa has no derivation. For string  $s_3$ , aba is derived through D and a is derived through A. Carteisan product DA is also not in the grammar. Therefore, row[4][1] =  $\emptyset$ .
- row[4][2] will look at substring baab. This string produces substrings  $s_1 = b, aab, s_2 = ba, ab,$  and  $s_3 = baa, b$ . For string  $s_1$ , aab has no derivation. For string  $s_2$ , ba is obtained through S and ab is obtained through S. Cartesian product SS is obtained through S. For string  $s_3$ , baa is obtained through S and S are string S are string S and S are string S and S are string S are string S and S are string S are string S are string S are string S and S are string S and S are string S are string S and S are string S and S are string S are string S and S are string S are string S and S are strin

6						
5						
4	S	Ø	S			
3	C	D	D	Ø		
2	S	S	S	Ø	S	
1	B	A	B	A	A	B
	b	a	b	a	a	b

Table 5: Row 4 filled

#### row[5] will look at substrings of length 5

- row[5][0] will look at substring babaa. This string can be split into:  $s_1 = b, abaa, s_2 = ba, baa, s_3 = bab, aa$ , and  $s_4 = baba, a$ . For string  $s_1$ , abaa has no derivation. For string  $s_2$ , ba is derived from S and baa is derived from D. The cartesian product, SD, is not in the grammar. For string  $s_3$ , aa has no derivation. For string  $s_4$ , baba is derived through S and S at through S and S are cartesian product, S and S are found through S at S and S are found through S are found through S and S are found through S and S are found through S are found through S are found through S and S are found through S are found through S and S are found through S and S are found through S and S are found through S are found through S and S are found through S are found through S are found through S and S are found through S are found through S and S are found through S and S are found through S and S are found through S are found through S and S are found through S are found through S and S are found through S are found through S are found throu
- row[5][1] will look at substring abaab. This string produces substrings:  $s_1 = a, baab, s_2 = ab, aab, s_3 = aba, ab$ , and  $s_4 = abaa, b$ . String  $s_1$  can be derived from A and S; however, cartesian product AS is not in the grammar. String  $s_2$  has no derivation because of aab. String  $s_3$  can be derived from D and S; however, DS is not in the grammar.  $s_4$  has no derivation because abaa does not. Therefore, row[5][1] has no derivation.

6						
5	D	Ø				
4	S	Ø	S			
3	C		D	Ø		
2	S	S	S	Ø	S	
1	B	A	В	A	A	В
	b	a	b	a	a	b

Table 6: Row 5 filled

### row[6] will look at substrings of length 6

• row[6][0] will look at string babaab. We can have substrings as follows:  $s_1 = b, abaab, s_2 = ba, baab, s_3 = bab, aab, s_4 = baba, ab,$  and  $s_5 = babaa, b.$   $s_1$ . In  $s_1$ , abaab has no derivation. In  $s_2$ , you can derivate the string through S and S. Thus the cartesian product SS can be derived

from S. In  $s_3$ , aab has no derivation. In  $s_4$ , you can derive it through S and S. The cartesian product is SS, which is derived through S. In  $s_4$ , you derive it through D and B, but DB is not in the grammar. Since  $s_2$  and  $s_4$  both get their derivations from S, then row[6][0] = S.

6	S					
5	D	$\emptyset$				
4	S	$\emptyset$	S			
3	C	D	D	Ø		
2	S	S	S	Ø	S	
1	B	$\overline{A}$	В	A	A	B
	b	a	b	a	a	b

Table 7: Row 6 filled

In conclusion, since row[6] contains the start variable S, then the string  $babaab \in L(A)$ .

# Problem #2: Is string $bababb \in L(A)$ ?

I will follow the same procedure as I did for problem #1. For the sake of simplicity, I will give only important and relevant information.

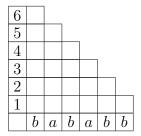


Table 8: Initial Unfilled Table

Again, row[1] contains the derivations of the letters directly below each box.

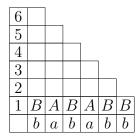


Table 9: First Row Filled

The following table are as follows:

	S	substring	derivated	Cartesian	In grammar?	
		of s	$\operatorname{from}$	Product	Where?	
row[2][0]	ba	b	В	BA	{S}	
100 [2][0]	Da	a	A	DIL	(6)	
row[2][1]	ab	a	A	$_{ m AB}$	{S}	
	ab	b	В	AD	ری	
row[2][2]	ba	b	В	$_{ m BA}$	{S}	
100 [2][2]	Da	a	A	DIL	[0]	
row[2][3]	ab	a	A	AB	{S}	
10W[2][0]	ab	b	В	AD	[2]	
row[2][4]	bb	b	В	BB	Ø	
10w[2][4]	55	b	В	מם	V	

	S	substring	derivated	Cartesian	In grammar?	
		of s	from	Product	Where?	
row[3][0]	bab	b, ab	B, S	{BS}	{C}	
10w[3][0]	Dab	ba, b	S, B	${SB}$	\ \_\	
row[3][1]	aba	a, ba	A, S	{AS}	{D}	
10w[3][1]	aba	ab, a	S, A	$\{SA\}$	[ ՄՄ	
row[3][2]	bab	b, ab	B, S	{BS}	{C}	
10w[0][2]	Dab	ba, b	S, B	${SB}$	{\cdot\}	
row[3][3]	abb	a, bb	A, Ø	Ø	{C}	
10w[9][9]	abb	ab, b	S, B	{SB}	[∪∫	

	S	substring of s	derivated from	Cartesian Product	In grammar?	Where?
		b, aba	B, D	{BD}	{S}	
row[4][0]	baba	ba, ba	S, S	{SS}	{S}	{S}
		bab, a	C, A	{CA}	{∅}	
		a, bab	A, C	{AC}	{S}	
row[4][1]	abab	ab, ab	S, S	{SS}	{S}	{S}
		aba, b	D, B	{∅}	{∅}	
		b, abb	В, С	{BC}	{∅}	
row[4][2]	babb	ba, bb	S, Ø	{∅}	{Ø}	{∅}
		bab, b	C, A	{CA}	{∅}	

	S	substrings of s	derivated from	Carteisan Product	In Grammar?	Final			
		b, abab	B, S	{BS}	{∅}				
row[5][0]	babab	ba, bab	S, C	{SC}	{∅}	$\mathbf{C}$			
$ \operatorname{row}[5][0] $ bal	Dabab	bab, ab	C, S	{CS}	{∅}	C			
		baba, b	S, B	{SB}	{C}				
		a, babb	A, Ø	$\{\emptyset\}$	$\{\emptyset\}$				
novv[E][1]	ababb	-1-11	-1-11	ahahh	ab, abb	S, C	{SC}	{∅}	C
row[5][1]		aba, bb	D, Ø	{∅}	{∅}				
		abab, b	S, B	{SB}	{C}				

	s	substrings of s	derivated from	Carteisan Product	In Grammar?	Final
		b, ababb	В, С	{BC}	{∅}	
		ba, babb	S, Ø	{∅}	{∅}	
row[6][0]	bababb	bab, abb	С, С	{CC}	{∅}	{∅}
		baba, bb	S, Ø	{∅}	{∅}	
		babab, b	С, В	{CB}	{∅}	

6	$\emptyset$					
5	C	C				
4	S	S	Ø			
3	C	D	C	C		
2	S	S	S	S	Ø	
1	B	A	B	A	B	B
	b	$\overline{a}$	b	$\overline{a}$	b	b

Table 10: Final Table

In conclusion, since row[6] does not contain start variable S, then string  $bababb \notin L(A)$ .

### Problem #3:

Give Turing machine transition table for a Turing Machine that accepts language:

 $\{x \in \{0,1\}^* \mid x \text{ begins with } 0 \text{ and has as many } 1 \text{ to } 0 \text{ transitions as } 0 \text{ to } 1 \text{ transitions}\}$ 

	0	1	В
$p_0$	R	$p_1 R$	Accept
$p_1$	$p_0 R$	R	Reject

This transition table shows that the turing machine will read each symbol and move states only when there is a change in input. It will accept if there is never a transition or if there is transition from 0-to-1 and then from 1-to-0. If there is a transition from 0-to-1 and the string ends there, the machine rejects such string.