AI665: Natural Language Processing

Hadeel Abdullah AlShehri

444008923

Assignment 3 - SLR(1)

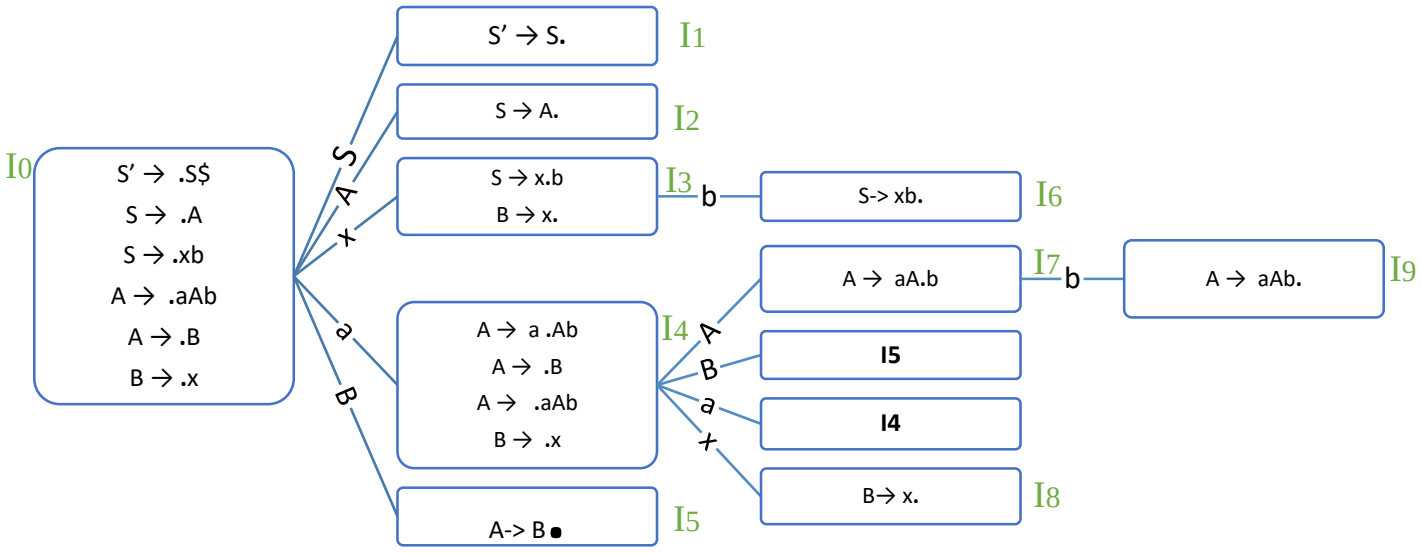
Q1: Please construct canonical Collection and parsing table of SLR(1) the following Grammer

S → A
S → xb
A → aAb
A → B
B → x

Step1: Add augment as first rule (S' → S$)

(1) S' → S$
(2) S → A
(3) S → xb
(4) A → aAb
(5) A → B
(6) B → x

Step2: Context-free grammar



Step3: Parsing table

1- Find follow for each Non Terminal

- Follow(S) = {$}
- Follow(A) = {$,b}
- Follow(B) = {$,b}

Status #	Activation Table				GO_TO		
	a	b	x	$	S	A	B
0	S4		S3		1	2	5
1				Accept			
2				r2			
3		S6/r6					
4	S4		S8			7	5
5		r5	r5				
6				r3			
7		S9					
8		r6		r6			
9		r4		r4			

Step4: Parse input (aaxbxb)

Stack	input	Action
0	aaxbxb $	-
0a	axbxb $	S4
0a4a4	xbxb $	S4
0a4a4x8	bxb $	S8
0a4a4	bxb $	(r6) Reduce by B → x (*2)
0a4a4B5	bxb $	Add B (NT) and (5)
0a4a4	bxb $	(r5) Reduce by A → B (*2)
0a4a4A7	bxb $	Add A (NT) and (7)
0a4a4A7b9	xb $	S9
0a4a4A7b9	xb $	No action (ERROR)

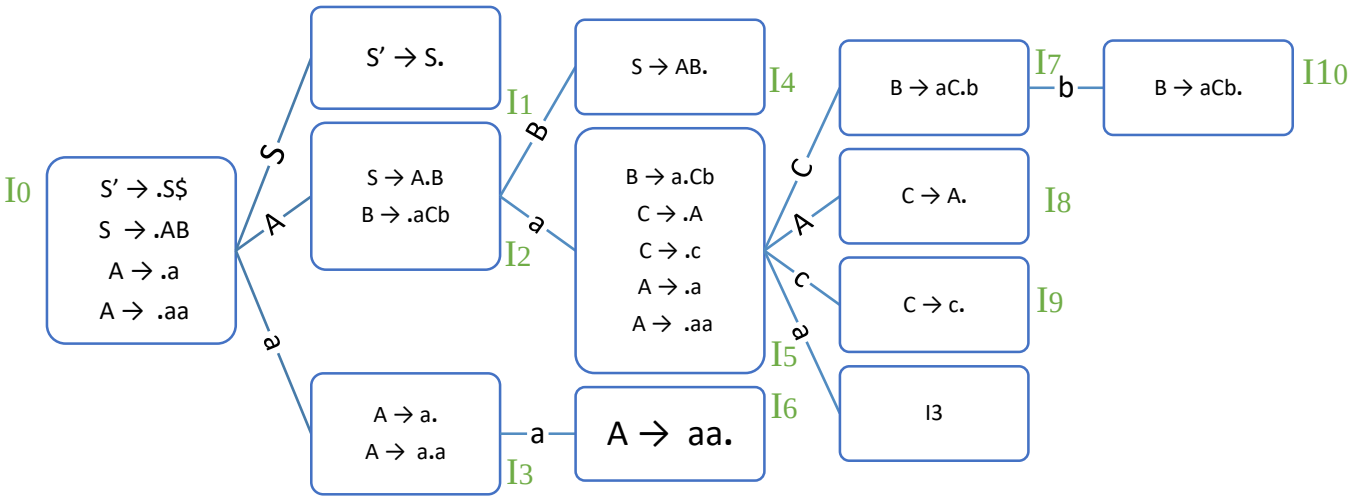
Q2: Please construct canonical Collection and parsing table of SLR(1) the following Grammer

S → AB
A → a
A → aa
C → A
C → c
B → aCb

Step1: Add augment as first rule (S' → S$)

(1) S' → S$
(2) S → AB
(3) A → a
(4) A → aa
(5) C → A
(6) C → c
(7) B → aCb

Step2: Context-free grammar



Step3: Parsing table

2- Find follow for each Non Terminal

- Follow(S) = {$}
- Follow(A) = {a, b}
- Follow(B) = {$}
- Follow(C) = {b}

Status #	Activation Table				GO_TO			
	a	c	b	$	S	A	B	C
0	S3				1	2		
1				Accept				
2	S5						4	
3	S6/r3		r3					
4				r2				
5	S3	S9				8		7
6	r4		r4					
7			S10					
8			r5					
9			r6					
10				r7				

Step4: Parse input (aaacb)

1- Stack and action

Stack	input	Action
0	aaacb $	-
0a3	aacb $	S3
0a3a6	acb $	S6
0	acb $	(r4) Reduce by A → aa (*4)
0A2	acb $	Add A (NT) and (2)
0A2a5	cb $	S5
0A2a5c9	b $	S9
0A2a5	b $	(r6) Reduce by C → c (*2)
0A2a5C7	b $	Add C (NT) and (7)
0A2a5C7b10	$	S10
0A2	$	(r7) Reduce by B → aCb (*6)
0A2B4	$	Add B (NT) and (4)
0	$	(r2) Reduce by S → AB (*4)
0S1	$	Add S (NT) and (1)

0S1$		Accept

2- Input Tree: