CS335 - Compilers
Assignment 2
Date: 12/02/2020

Name: Sarthak Singhal Roll No.: 170635

1.

$$S \rightarrow (L)|a$$
$$L \rightarrow L, S|LS|b$$

Left factoring the grammar gives:

$$S \rightarrow (L)|a$$
$$L \rightarrow LX|b$$
$$X \rightarrow , S|S$$

Removing left recursion gives:

$$S \rightarrow (L)|a$$
$$L \rightarrow bL^{'}$$
$$L^{'} \rightarrow XL^{'}|\epsilon$$
$$X \rightarrow , S|S$$

$FIRST(S) = \{(, a\}$
$FIRST(L) = \{b\}$
$FIRST(L^{'}) = \{`,', (, a, \epsilon\}$
$FIRST(X) = \{`,', (, a\}$

$FOLLOW(S) = \{\$, (, a, `,',)\}$
$FOLLOW(L) = \{)\}$
$FOLLOW(L^{'}) = \{)\}$
$FOLLOW(X) = \{`,', (, a,)\}$

Predictive parsing table for this grammar is:

Non-terminal	**a**	**b**	**(**	**)**	**,**	**$**
S	$S \rightarrow a$		$S \rightarrow (L)$			
L		$L \rightarrow bL^{'}$				
$L^{'}$	$L^{'} \rightarrow XL^{'}$		$L^{'} \rightarrow XL^{'}$	$L^{'} \rightarrow \epsilon$	$L^{'} \rightarrow XL^{'}$	
X	$X \rightarrow S$		$X \rightarrow S$		$X \rightarrow , S$	

2.

$$S^{'} \rightarrow S$$
$$S \rightarrow Lp|qLr|sr|qsp$$
$$L \rightarrow s$$

Canonoical collection of set of LR(0) items:

I_0= Closure($S^{'} \rightarrow .S$) ={
$S^{'} \rightarrow .S$,
$S \rightarrow .Lp$,
$S \rightarrow .qLr$,
$S \rightarrow .sr$,
$S \rightarrow .qsp$,
$L \rightarrow .s$
}

I_1= Goto(I_0, S) ={
$S^{'} \rightarrow S.$
}

I_2= Goto(I_0, L) ={
$S \rightarrow L.p$
}

I_3= Goto(I_0, q) ={
$S \rightarrow q.Lr$,
$L \rightarrow .s$,
$S \rightarrow q.sp$
}

I_4=Goto(I_0, s) ={
$S \rightarrow s.r$,
$L \rightarrow s.$
}

In I_4 there is shift action on seeing r.
$FOLLOW(L) = \{p, r\}$
There can also be a reduce step in I_4 on seeing r as it belongs to FOLLOW(L). Therefore, there is a shift-reduce conflict in I_4 so there is no need to make more items further. Hence, due to the S-R conflict the grammar is not SLR(1).

Canonical collection of set of LR(1) items:

I_0= Closure($S^{'} \rightarrow .S, \$$) ={
$S^{'} \rightarrow .S, \$$,
$S \rightarrow .Lp, \$$,
$S \rightarrow .qLr, \$$,
$S \rightarrow .sr, \$$,
$S \rightarrow .qsp, \$$,
$L \rightarrow .s, p$
}
I_1= Goto(I_0, S) ={
$S^{'} \rightarrow S., \$$
}

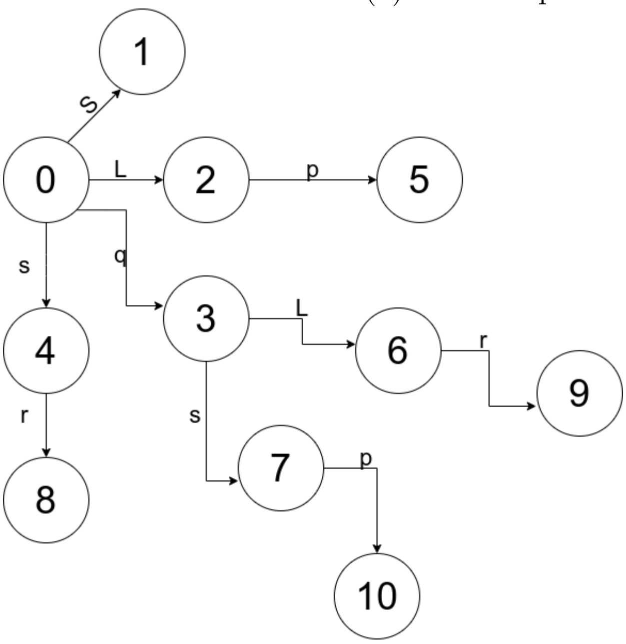
I_2= Goto(I_0, L) ={
$S \rightarrow L.p, \$$
}

I_3= Goto(I_0, q) ={
$S \rightarrow q.Lr, \$$,
$L \rightarrow .s, r$,
$S \rightarrow q.sp, \$$
}

I_4=Goto(I_0, s) ={
$S \rightarrow s.r, \$$,
$L \rightarrow s., p$
}
I_5=Goto(I_2, p) ={
$S \rightarrow Lp., \$$
}
I_6=Goto(I_3, L) ={
$S \rightarrow qL.r, \$$
}

I_7=Goto(I_3, s) ={
$S \rightarrow qs.p, \$$,
$L \rightarrow s., r$
}
I_8=Goto(I_4, r) ={
$S \rightarrow sr., \$$
}
I_9=Goto(I_6, r) ={
$S \rightarrow qLr., \$$
}
I_{10}=Goto(I_7, p) ={
$S \rightarrow qsp., \$$
}

All items have different cores therefore LALR(1) will be equal to LR(1).



LR(1) automaton:

Let us number the rules to display reduce action:
1. $S \to Lp$
2. $S \to qLr$
3. $S \to sr$
4. $S \to qsp$
5. $L \to s$

LALR(1) parsing table:

State	**Action**					**Goto**	
	p	**q**	**r**	**s**	**$**	**S**	**L**
0		s3		s4		1	2
1					accept		
2	s5						
3				s7			6
4	r5		s8				
5					r1		
6			s9				
7	s10		r5				
8					r3		
9					r2		
10					r4		

As there are no S-R and R-R conflicts the given grammar is LALR(1).

3.

$$R' \to R$$
$$R \to R|R$$
$$R \to RR$$
$$R \to R^*$$
$$R \to (R)$$
$$R \to a|b$$

Canonical collection of set of LR(0) items:

I_0= Closure($R' \to .R$) ={
$R' \to .R,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R^*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
}

I_1= Goto(I_0, R) ={
$R' \to R.,$
$R \to R.|R,$
$R \to R.R,$
$R \to R.^*,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R^*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
}

I_2= Goto($I_0, ($) ={
$R \to (.R),$
$R \to .R|R,$
$R \to .RR,$
$R \to .R^*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
}
I_3= Goto(I_0, a) ={
$R \to a.$
}

I_4=Goto(I_0, b) ={
$R \to b.$
}

$I_5 = \text{Goto}(I_1, R) = \{$
$R \to RR.,$
$R \to R.|R,$
$R \to R.R,$
$R \to R.*,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
$\}$

$I_6 = \text{Goto}(I_1, |) = \{$
$R \to R|.R,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
$\}$

$I_7 = \text{Goto}(I_1, {}^*) = \{$
$R \to R*.$
$\}$

$I_2 = \text{Goto}(I_1, ()$
$I_3 = \text{Goto}(I_1, a)$
$I_4 = \text{Goto}(I_1, b)$
$I_2 = \text{Goto}(I_2, ()$
$I_3 = \text{Goto}(I_2, a)$
$I_4 = \text{Goto}(I_2, b)$

$I_8 = \text{Goto}(I_2, R) = \{$
$R \to (R.),$
$R \to R.|R,$
$R \to R.R,$
$R \to R.*,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
$\}$

$I_5 = \text{Goto}(I_5, R)$
$I_6 = \text{Goto}(I_5, |)$
$I_7 = \text{Goto}(I_5, {}^*)$
$I_2 = \text{Goto}(I_5, ()$
$I_2 = \text{Goto}(I_6, ()$
$I_3 = \text{Goto}(I_5, a)$
$I_4 = \text{Goto}(I_5, b)$
$I_3 = \text{Goto}(I_6, a)$
$I_4 = \text{Goto}(I_6, b)$

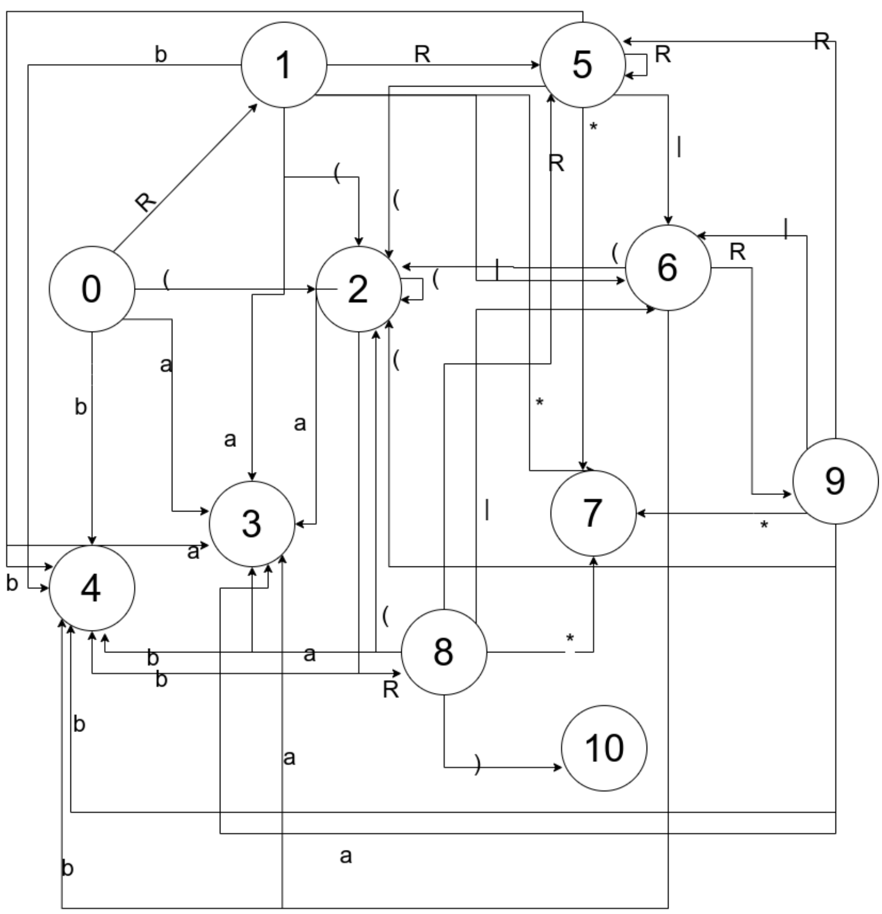
$I_9 = \text{Goto}(I_6, R) = \{$
$R \to R|R.,$
$R \to R.|R,$
$R \to R.R,$
$R \to R.*,$
$R \to .R|R,$
$R \to .RR,$
$R \to .R*,$
$R \to .(R),$
$R \to .a,$
$R \to .b$
$\}$

$I_6 = \text{Goto}(I_8, |)$
$I_7 = \text{Goto}(I_8, {}^*)$
$I_2 = \text{Goto}(I_8, ()$
$I_5 = \text{Goto}(I_8, R)$

$I_{10} = \text{Goto}(I_8,))\{$
R $\to (R).$
$\}$

$I_5 = \text{Goto}(I_9, R)$
$I_7 = \text{Goto}(I_9, {}^*)$
$I_2 = \text{Goto}(I_9, ()$
$I_6 = \text{Goto}(I_9, |)$

$I_3 = \text{Goto}(I_8, a)$
$I_4 = \text{Goto}(I_8, b)$
$I_3 = \text{Goto}(I_9, a)$
$I_4 = \text{Goto}(I_9, b)$



LR(0) automaton:

FOLLOW(R) $= \{\$,|,^*,),(,a,b\}$
Let us number the rules to display reduce action:
1. $R \rightarrow R|R$
2. $R \rightarrow RR$
3. $R \rightarrow R^*$
4. $R \rightarrow (R)$
5. $R \rightarrow a$
6. $R \rightarrow b$

SLR(1) parsing table:

State	**Action**							**Goto**
	a	**b**	$\vert$	**(**	**)**	*	**\$**	**R**
0	s3	s4		s2				1
1	s3	s4	s6	s2		s7	accept	5
2	s3	s4		s2				8
3	r5	r5	r5	r5	r5	r5	r5	
4	r6	r6	r6	r6	r6	r6	r6	
5	s3 r2	s4 r2	s6 r2	s2 r2	r2	s7 r2	r2	5
6	s3	s4		s2				9
7	r3	r3	r3	r3	r3	r3	r3	
8	s3	s4	s6	s2	s10	s7		5
9	s3 r1	s4 r1	s6 r1	s2 r1	r1	s7 r1	r1	5
10	r4	r4	r4	r4	r4	r4	r4	

The precedence in decreasing order is: $()^*.|$

Rules to remove ambiguation:

1. State 5 on a:
It says either to shift on seeing a or reduce using $R \rightarrow RR$ but as . is left associative we will reduce.

2. State 5 on b:
It says either to shift on seeing b or reduce using $R \rightarrow RR$ but as . is left associative we will reduce.

3. State 5 on $|$:
It says either to shift on seeing $|$ or reduce using $R \rightarrow RR$ but as . has more precedence than $|$ we will reduce.

4. State 5 on (:
It says either to shift on seeing (or reduce using $R \rightarrow RR$ but as () has more precedence than . we will shift.

5. State 5 on *:
It says either to shift on seeing * or reduce using $R \rightarrow RR$ but as * has more precedence than . we will shift.

6. State 9 on a:
It says either to shift on seeing a or reduce using $R \rightarrow R|R$ but as . has more precedence than $|$ we will shift.

7. State 9 on b:
It says either to shift on seeing b or reduce using $R \rightarrow R|R$ but as . has more precedence than $|$ we will shift.

8. State 9 on $|$:
It says either to shift on seeing $|$ or reduce using $R \rightarrow R|R$ but as $|$ is left associative we will

reduce.

9. State 9 on (:
It says either to shift on seeing (or reduce using $R \rightarrow R|R$ but as () has more precedence than | we will shift.

10. State 9 on *:
It says either to shift on seeing * or reduce using $R \rightarrow R|R$ but as * has more precedence than | we will shift.

New SLR(1) parsing table:

State	**Action**							**Goto**
	a	**b**	|	(	)	*	**$**	**R**
0	s3	s4		s2				1
1	s3	s4	s6	s2		s7	accept	5
2	s3	s4		s2				8
3	r5	r5	r5	r5	r5	r5	r5	
4	r6	r6	r6	r6	r6	r6	r6	
5	r2	r2	r2	s2	r2	s7	r2	5
6	s3	s4		s2				9
7	r3	r3	r3	r3	r3	r3	r3	
8	s3	s4	s6	s2	s10	s7		5
9	s3	s4	r1	s2	r1	s7	r1	5
10	r4	r4	r4	r4	r4	r4	r4	

Now it has no S-R or R-R conflicts.

4. Flex was used to generate the tokens and Bison was used to create the parser.
Make utility was also used to compile and link the source files.
To compile simply write:

- make

To execute write:

- ./parser < input_file