CS335A: Assignment 2

Deepak Sangle, 200860
sangleds20@iitk.ac.in

February 14, 2023

Problem 1

For the following grammar, design a predictive parser and show the predictive parsing table. Perform desired processing like removing left-recursion and left-factoring on the grammar if required.

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

Solution

The grammar is left recursive because of the second production rule. It also contains the left factoring. The grammar after removing left recursion is

$$S \rightarrow (L) \mid a$$
$$L \rightarrow bL_1$$
$$L_1 \rightarrow , SL_1 \mid SL_1 \mid \epsilon$$

(Here the ϵ denotes an empty terminal). This also removes the left factoring. Thus the grammar is now SLR grammar.

The FIRST Set and FOLLOW Set of all the non-terminals are

Non Terminal	FIRST	FOLLOW
S	(a	, (a) $
L	b	)
L_1	, (a ϵ	)

The predictive parsing table for the above grammar is

Non Terminal	$	,	b	a	(	)
S				$S \to a$		$S \to (L)$
L			$L \to bL_1$			
L_1		$L_1 \to, SL_1$		$L_1 \to SL_1$	$L_1 \to \epsilon$	$L_1 \to SL_1$

Problem 2

Show that the following grammar is LALR(1) but not SLR(1).

$$S \to Lp \mid qLr \mid sr \mid qsp$$
$$L \to s$$

Solution

The augmented grammar is

$$S' \to S$$
$$S \to Lp \mid qLr \mid sr \mid qsp$$
$$L \to s$$

The FIRST Set and FOLLOW Set of all the non-terminals are

Non Terminal	FIRST	FOLLOW
S	q s	$
S'	q s	$
L	s	p r

The canonical collection of sets of LR(0) items is given below,

$$
\begin{aligned}
I_0 = Closure(\{S' \to \cdot S\}) = \{ \\
S' &\to \cdot S \\
S &\to \cdot Lp \\
S &\to \cdot qLr \\
S &\to \cdot sr \\
S &\to \cdot qsp \\
L &\to \cdot s\}
\end{aligned}
$$

$$
\begin{aligned}
I_1 = \texttt{GOTO}(I_0, S) = Closure(\{S' \to S\cdot\}) = \{ \\
S' \to S\cdot\}
\end{aligned}
$$

$$
\begin{aligned}
I_2 = \texttt{GOTO}(I_0, L) = Closure(\{S \to L \cdot p\}) = \{ \\
S \to L \cdot p\}
\end{aligned}
$$

$$
\begin{aligned}
I_3 = \texttt{GOTO}(I_0, q) = Closure(\{S \to q \cdot Lr\ ,\ S \to q \cdot sp\}) = \{ \\
S &\to q \cdot Lr \\
S &\to q \cdot sp \\
L &\to \cdot s\}
\end{aligned}
$$

$$
\begin{aligned}
I_4 = \texttt{GOTO}(I_0, s) = Closure(\{S \to s \cdot r\ , L \to s\cdot\}) = \{ \\
S &\to s \cdot r \\
L &\to s\cdot\}
\end{aligned}
$$

$$
\begin{aligned}
I_5 = \texttt{GOTO}(I_2, p) = Closure(\{S \to Lp\cdot\}) = \{ \\
S \to Lp\cdot\}
\end{aligned}
$$

$$I_6 = \texttt{GOTO}(I_3, L) = Closure(\{S \rightarrow qL \cdot r\}) = \{$$
$$S \rightarrow qL \cdot r\}$$

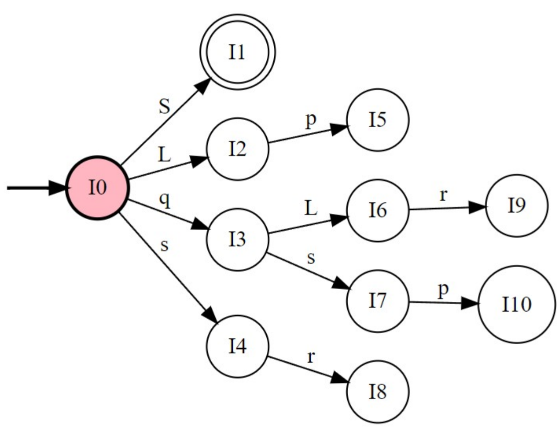
$$I_7 = \texttt{GOTO}(I_3, s) = Closure(\{S \rightarrow qs \cdot p\,, L \rightarrow s \cdot\}) = \{$$
$$S \rightarrow qs \cdot p$$
$$L \rightarrow s \cdot\}$$

$$I_8 = \texttt{GOTO}(I_4, r) = Closure(\{S \rightarrow sr \cdot\}) = \{$$
$$S \rightarrow sr \cdot\}$$

$$I_9 = \texttt{GOTO}(I_6, r) = Closure(\{S \rightarrow qLr \cdot\}) = \{$$
$$S \rightarrow qLr \cdot\}$$

$$I_{10} = \texttt{GOTO}(I_7, p) = Closure(\{S \rightarrow qsp \cdot\}) = \{$$
$$S \rightarrow qsp \cdot\}$$

Thus the LR(0) Automaton that we get based on the above sets is given below. Here the I1 state is the accepting state while the I0 state is the starting state of the automaton.



(1)

The SLR Parsing table is given below.

State No.	\$	s	p	q	r	S	L
0		I_4		I_3		I_1	I_2
1	Accept						
2			Shift: I_5				
3		Shift: I_7					Shift: I_6
4			Red.: $L \to s$		Shift: I_8 / Red.: $L \to s$		
5	Red.: $S \to Lp$						
6					Shift: I_9		
7			Shift: I_{10} / Red.: $L \to s$		Red.: $L \to s$		
8	Red.: $S \to sr$						
9	Red.: $S \to qLr$						
10	Red.: $S \to qsp$						

The LR(1) Canonical Collection of the given grammar is given below.

$$
\begin{aligned}
I_0 = Closure(\{[S' \to \cdot S, \$]\}) = \{ \\
[S' \to \cdot S, \$] \\
[S \to \cdot Lp, \$] \\
[S \to \cdot qLr, \$] \\
[S \to \cdot sr, \$] \\
[S \to \cdot qsp, \$] \\
[L \to \cdot s, p] \ \}
\end{aligned}
$$

$$
\begin{aligned}
I_1 = \texttt{GOTO}(I_0, S) = Closure(\{[S' \to S\cdot, \$]\}) = \{ \\
[S' \to S\cdot, \$] \ \}
\end{aligned}
$$

$$
\begin{aligned}
I_2 = \texttt{GOTO}(I_0, L) = Closure(\{[S \to L \cdot p, \$]\}) = \{ \\
[S \to L \cdot p, \$] \ \}
\end{aligned}
$$

$$I_3 = \texttt{GOTO}(I_0, q) = Closure(\{[S \to q \cdot Lr, \$], [S \to q \cdot sp, \$]\}) = \{$$
$$[S \to q \cdot Lr, \$]$$
$$[S \to q \cdot sp, \$]$$
$$[L \to \cdot s, r] \;\; \}$$

$$I_4 = \texttt{GOTO}(I_0, s) = Closure(\{[S \to s \cdot r, \$], [L \to s\cdot, p\$]\}) = \{$$
$$[S \to s \cdot r, \$]$$
$$[L \to s\cdot, p] \;\; \}$$

$$I_5 = \texttt{GOTO}(I_2, p) = Closure(\{[S \to Lp\cdot, \$]\}) = \{$$
$$[S \to Lp\cdot, \$] \;\; \}$$

$$I_6 = \texttt{GOTO}(I_3, L) = Closure(\{[S \to qL \cdot r, \$]\}) = \{$$
$$[S \to qL \cdot r, \$] \;\; \}$$

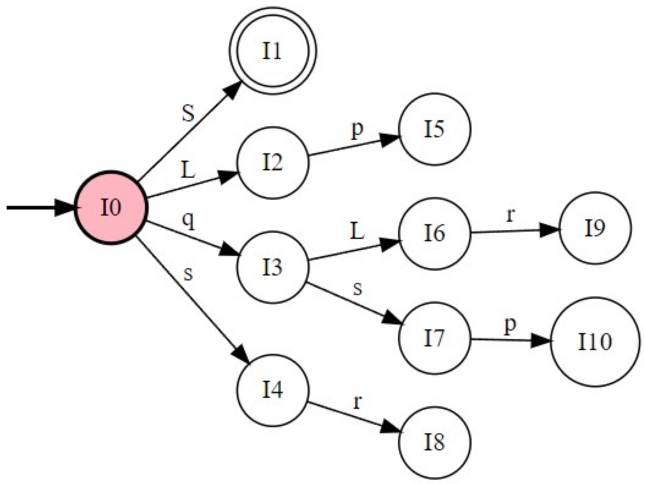
$$I_7 = \texttt{GOTO}(I_3, s) = Closure(\{[S \to qs \cdot p, \$], [L \to s\cdot, r]\}) = \{$$
$$[S \to qs \cdot p, \$]$$
$$[L \to s\cdot, r] \;\; \}$$

$$I_8 = \texttt{GOTO}(I_4, r) = Closure(\{[S \to sr\cdot, \$]\}) = \{$$
$$[S \to sr\cdot, \$] \;\; \}$$

$$I_9 = \texttt{GOTO}(I_6, r) = Closure(\{[S \to qLr\cdot, \$]\}) = \{$$
$$[S \to qLr\cdot, \$] \;\; \}$$

$$I_{10} = \texttt{GOTO}(I_7, p) = Closure(\{[S \to qsp\cdot, \$]\}) = \{$$
$$[S \to qsp\cdot, \$]\ \}$$

The LR(1) automaton corresponding to the above canonical collection of items is given below.



The LALR Parsing table is given below.

State	\$	s	p	q	r	S	L
I_0		Shift: I_4		Shift: I_3		Shift: I_1	Shift: I_2
I_1	Accept						
I_2			Shift: I_5				
I_3		Shift: I_7					Shift: I_6
I_4		Red.: $L \to s$			Shift: I_8		
I_5	$S \to Lp$						
I_6					Shift: I_9		
I_7			Shift: I_{10}		Red.: $L \to s$		
I_8	Red.: $S \to sr$						
I_9	Red.: $S \to qLr$						
I_{10}	Red.: $S \to qsp$						

Thus, we conclude the following points from the above tables and automatons

- For SLR Parsing Table, we can find shift-reduce conflicts for the state numbered 7 and 5. Thus the given grammar is not SLR.
- For LALR Parsing Table, we couldn't find any shift-reduce conflicts and thus concludes that the grammar is indeed LALR.

Problem 3

Construct an SLR parsing table for the following grammar. Show the canonical set of states and the transition diagram.

$$R \to R|R$$
$$R \to RR$$
$$R \to R*$$
$$R \to (R)$$
$$R \to a \mid b$$

Resolve the parsing action conflicts in such a way that regular expressions will be parsed normally. Include your disambiguation rules in the PDF file, and show the final parsing table.

Solution

We first create an augmented grammar with the start symbol R' such that $R' \to R$.
Firstly, the FIRST and FOLLOW Set of the grammar is given below,

Non Terminal	FIRST	FOLLOW
R	(a b	(a b) | * $
R'	(a b	$

State	Canonical Set
I_0	$\{ R' \to \cdot R,\ R \to \cdot R\|R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_1	$\{ R' \to R\cdot,\ R \to R\cdot\|R,\ R \to R\cdot R,\ R \to R\cdot *,\ R \to \cdot R\|R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_2	$\{ R \to (\cdot R),\ R \to \cdot R\|R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_3	$\{ R \to a\cdot \}$
I_4	$\{ R \to b\cdot \}$
I_5	$\{ R \to R\|\cdot R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_6	$\{ R \to RR\cdot,\ R \to R\cdot\|R,\ R \to R\cdot R,\ R \to R\cdot *,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot R\|R,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b\}$
I_7	$\{ R \to R*\cdot \}$
I_8	$\{ R \to (R\cdot),\ R \to R\cdot\|R,\ R \to R\cdot R,\ R \to R\cdot *,\ R \to \cdot R\|R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_9	$\{ R \to R\|R\cdot,\ R \to R\cdot\|R,\ R \to R\cdot R,\ R \to R\cdot *,\ R \to \cdot R\|R,\ R \to \cdot RR,\ R \to \cdot R*,\ R \to \cdot(R),\ R \to \cdot a,\ R \to \cdot b \}$
I_{10}	$\{ R \to (R)\cdot \}$

The LR(0) Canonical collection of items is given above.
The GOTO Functions only when new states gets created of the automata is given below.

$$\texttt{GOTO}(I_0, R) = I_1 \qquad \texttt{GOTO}(I_0, () = I_2 \qquad \texttt{GOTO}(I_0, a) = I_3$$
$$\texttt{GOTO}(I_0, b) = I_4 \qquad \texttt{GOTO}(I_1, |) = I_5 \qquad \texttt{GOTO}(I_1, R) = I_6$$
$$\texttt{GOTO}(I_1, *) = I_7 \qquad \texttt{GOTO}(I_2, R) = I_8 \qquad \texttt{GOTO}(I_5, R) = I_9$$
$$\texttt{GOTO}(I_8,)) = I_{10}$$

The SLR Parsing table for the given grammar is given below.

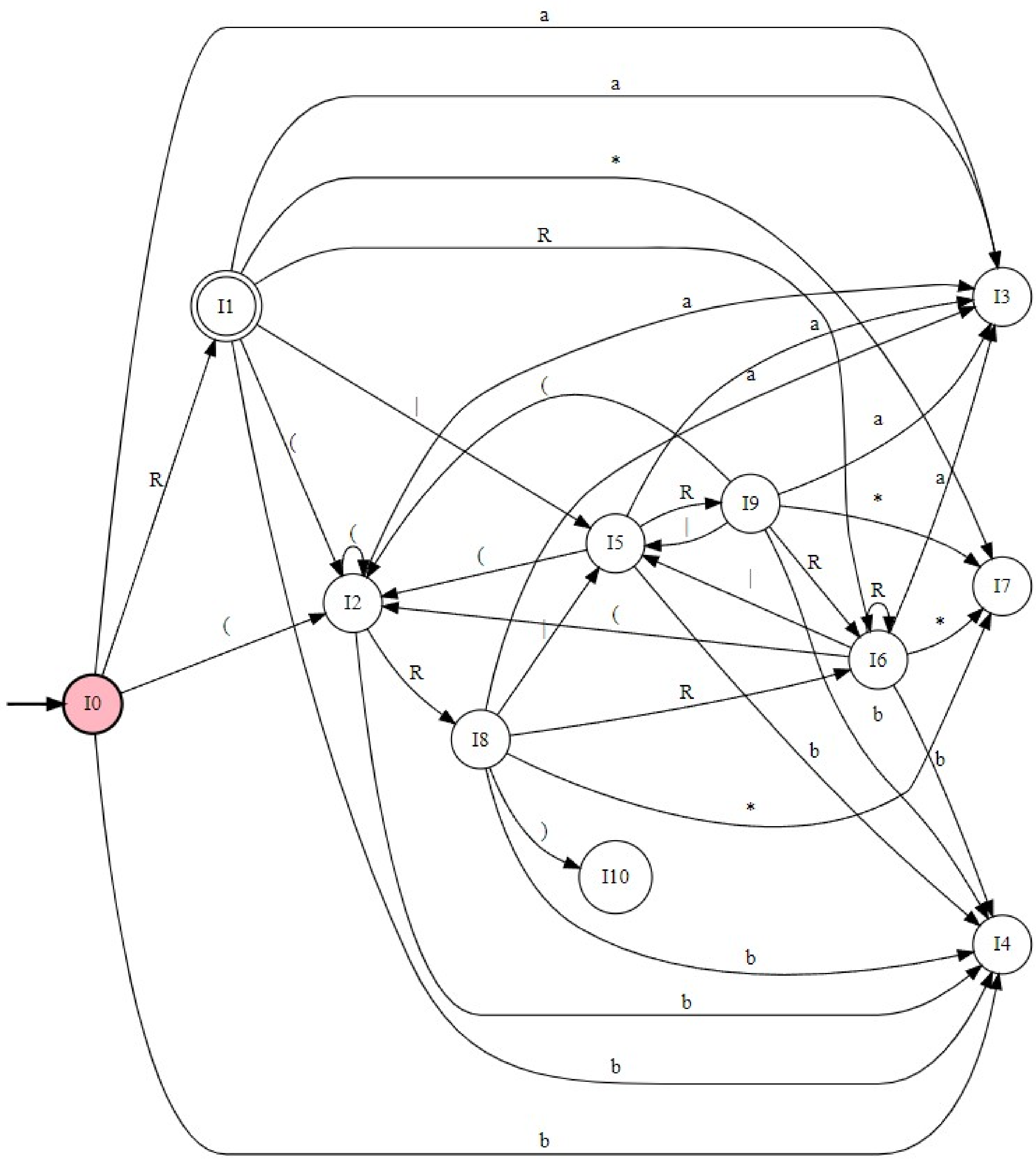
State	$	$\vert$	*	(	)	a	b	R
I_0				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_1
I_1	Accept	Shift: I_5	Shift: I_7	Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_6
I_2				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_8
I_3	$R \to a\cdot$	$R \to a\cdot$	$R \to a\cdot$	$R \to a\cdot$	$R \to a\cdot$	$R \to a\cdot$	$R \to a\cdot$	
I_4	$R \to b\cdot$	$R \to b\cdot$	$R \to b\cdot$	$R \to b\cdot$	$R \to b\cdot$	$R \to b\cdot$	$R \to b\cdot$	
I_5				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_9
I_6	$R \to RR\cdot$	Shift: I_5 $R \to RR\cdot$	Shift: I_7 $R \to RR\cdot$	Shift: I_2 $R \to RR\cdot$	$R \to RR\cdot$	Shift: I_3 $R \to RR\cdot$	Shift: I_4	Shift: I_6
I_7	$R \to R*\cdot$	$R \to R*\cdot$	$R \to R*\cdot$	$R \to R*\cdot$	$R \to R*\cdot$	$R \to R*\cdot$	$R \to R*\cdot$	
I_8		Shift: I_5	Shift: I_7	Shift: I_2	Shift: I_{10}	Shift: I_3	Shift: I_4	Shift: I_6
I_9	$R \to R\vert R\cdot$	Shift: I_5 $R \to R\vert R\cdot$	Shift: I_7 $R \to R\vert R\cdot$	Shift: I_2 $R \to R\vert R\cdot$	$R \to R\vert R\cdot$	Shift: I_3 $R \to R\vert R\cdot$	Shift: I_4 $R \to R\vert R\cdot$	Shift: I_6
I_{10}	$R \to (R)\cdot$	$R \to (R)\cdot$	$R \to (R)\cdot$	$R \to (R)\cdot$	$R \to (R)\cdot$	$R \to (R)\cdot$	$R \to (R)\cdot$	

We can clearly see that there are many shift-reduce conflicts for the state I_6 and I_9. Thus the grammar is not in SLR Grammar. But, we can remove these conflicts by including left associativity and operator precedence for the terminal symbols of the grammar. I assumed the following left operator precedence:

$$() > * > concatenate^{\dagger} > |$$

(†: the concatenate operation is useful for the production rule $R \to RR$)

Now, we are reducing the viable prefix instead of shifting the terminals because we assumed left associativity between these operators.



The automaton for the given SLR Grammar denoting start state I_0 and accepting state I_1.

The modified SLR Parsing table after the modification mentioned above is given below.

State	\$	|	*	(	)	a	b	R
I_0				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_1
I_1	Accept	Shift: I_5	Shift: I_7	Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_6
I_2				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_8
I_3	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	$R \rightarrow a\cdot$	
I_4	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	$R \rightarrow b\cdot$	
I_5				Shift: I_2		Shift: I_3	Shift: I_4	Shift: I_9
I_6	$R \rightarrow RR\cdot$	$R \rightarrow RR\cdot$	Shift: I_7	$R \rightarrow RR\cdot$	$R \rightarrow RR\cdot$	$R \rightarrow RR\cdot$	$R \rightarrow RR\cdot$	Shift: I_6
I_7	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	$R \rightarrow R*\cdot$	
I_8		Shift: I_5	Shift: I_7	Shift: I_2	Shift: I_{10}	Shift: I_3	Shift: I_4	Shift: I_6
I_9	$R \rightarrow R\|R\cdot$	$R \rightarrow R\|R\cdot$	Shift: I_7	Shift: I_2	$R \rightarrow R\|R\cdot$	Shift: I_3	Shift: I_4	Shift: I_6
I_{10}	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	$R \rightarrow (R)\cdot$	

This modified parsing table can be used in such a way that regular expressions can be parsed normally.

Problem 4

Solution

The Grammar and Lexer are present in the given zip file. There is also a README.md file that contains instructions on how to run the given files.