Compilers Assignment 2

Sarthak Kalankar

Problem 1

The given context-free grammar is as follows:

$$
\begin{aligned}
\text{Function} &\rightarrow \text{Type } \textbf{id}(\text{Arguments}) \\
\text{Type} &\rightarrow \textbf{id} \\
\text{Type} &\rightarrow \text{Type}\textbf{*} \\
\text{Arguments} &\rightarrow \text{ArgList} \\
\text{Arguments} &\rightarrow \epsilon \\
\text{ArgList} &\rightarrow \text{Type } \textbf{id}, \text{ArgList} \\
\text{ArgList} &\rightarrow \text{Type } \textbf{id}
\end{aligned}
$$

(i) Explanation

The given grammar is not LL(1) because of the ambiguity and left recursion in the productions.

(ii) Transformed Grammar

$$
\begin{aligned}
\text{Function} &\rightarrow \text{Type } \textbf{id} \text{ (Arguments)} \\
\text{Type} &\rightarrow \text{id Type'} \\
\text{Type'} &\rightarrow * \text{Type'} \\
\text{Type'} &\rightarrow \epsilon \\
\text{Arguments} &\rightarrow \text{ArgList} \\
\text{Arguments} &\rightarrow \epsilon \\
\text{ArgList} &\rightarrow \text{Type id ArgList'} \\
\text{ArgList'} &\rightarrow , \text{ArgList} \\
\text{ArgList'} &\rightarrow , \epsilon
\end{aligned}
$$

(iii) FIRST and FOLLOW Sets

$$\begin{aligned}
\text{FIRST(Function)} &= \{id\} \\
\text{FIRST(Type)} &= \{id\} \\
\text{FIRST(Type')} &= \{\epsilon, *\} \\
\text{FIRST(Arguments)} &= \{\epsilon, id\} \\
\text{FIRST(ArgList)} &= \{id\} \\
\text{FIRST(ArgList')} &= \{\epsilon, , \}
\end{aligned}$$

$$\begin{aligned}
\text{FOLLOW(Function)} &= \{\$\} \\
\text{FOLLOW(Type)} &= \{id\} \\
\text{FOLLOW(Type')} &= \{id\} \\
\text{FOLLOW(Arguments)} &= \{)\} \\
\text{FOLLOW(ArgList)} &= \{)\} \\
\text{FOLLOW(ArgList')} &= \{)\}
\end{aligned}$$

(iv) LL(1) Parsing Table

	id	(	)	*	,	$
Function	Function → Type **id** (Arguments)					
Type	Type → id Type'					
Type'	Type' → ϵ			Type' → $*$ Type'		
Arguments	Arguments → ArgList		Arguments → ϵ			
ArgList	ArgList → Type id ArgList'					
ArgList'			ArgList' →, ϵ		ArgList' →, ArgList	

Problem 2

0. $S' \to S$
1. $S \to LM$
2. $S \to Lp$
3. $S \to qLr$
4. $S \to sr$
5. $S \to qsp$
6. $L \to aMb$
7. $L \to s$
8. $L \to t$
9. $M \to t$

FIRST and FOLLOW sets:

$$FIRST(S) = \{a, s, q, t\}$$
$$FIRST(L) = \{a, s, t\}$$
$$FIRST(M) = \{t\}$$

$$FOLLOW(S) = \{\$\}$$
$$FOLLOW(L) = \{p, r, t\}$$
$$FOLLOW(M) = \{b, \$\}$$

Cannonical LR(0) Collection

$I_0 = Closure(\{[S' \to \cdot S]\})$
$= \{[S' \to \cdot S],\ [S \to \cdot LM],\ [S \to \cdot Lp],\ [S \to \cdot qLr],\ [S \to \cdot sr],\ [S \to \cdot qsp],\ [L \to \cdot aMb],\ [L \to \cdot s],\ [L \to \cdot t]\}$

$I_1 = GOTO(I_0, S) = \{[S' \to S\cdot]\}$
$I_2 = GOTO(I_0, L) = \{[S \to L \cdot M],\ [S \to L \cdot p],\ [M \to \cdot t]\}$
$I_3 = GOTO(I_0, q) = \{[S \to q \cdot Lr],\ [S \to q \cdot sp],\ [L \to \cdot aMb],\ [L \to \cdot s],\ [L \to \cdot t]\}$
$I_4 = GOTO(I_0, s) = \{[S \to s \cdot r],\ [L \to s\cdot]\}$
$I_5 = GOTO(I_0, a) = \{[L \to a \cdot Mb],\ [M \to \cdot t]\}$
$I_6 = GOTO(I_0, t) = \{[L \to t\cdot]\}$

$I_7 = GOTO(I_2, M) = \{[S \to LM\cdot]\}$
$I_8 = GOTO(I_2, p) = \{[S \to Lp\cdot]\}$
$I_9 = GOTO(I_2, t) = \{[M \to t\cdot]\}$

$I_{10} = GOTO(I_3, L) = \{[S \to qL \cdot r]\}$
$I_{11} = GOTO(I_3, s) = \{[S \to qs \cdot p], [L \to s\cdot]\}$

$I_{12} = GOTO(I_4, r) = \{[S \to sr\cdot]\}$

$I_{13} = GOTO(I_5, M) = \{[L \to aM \cdot b]\}$

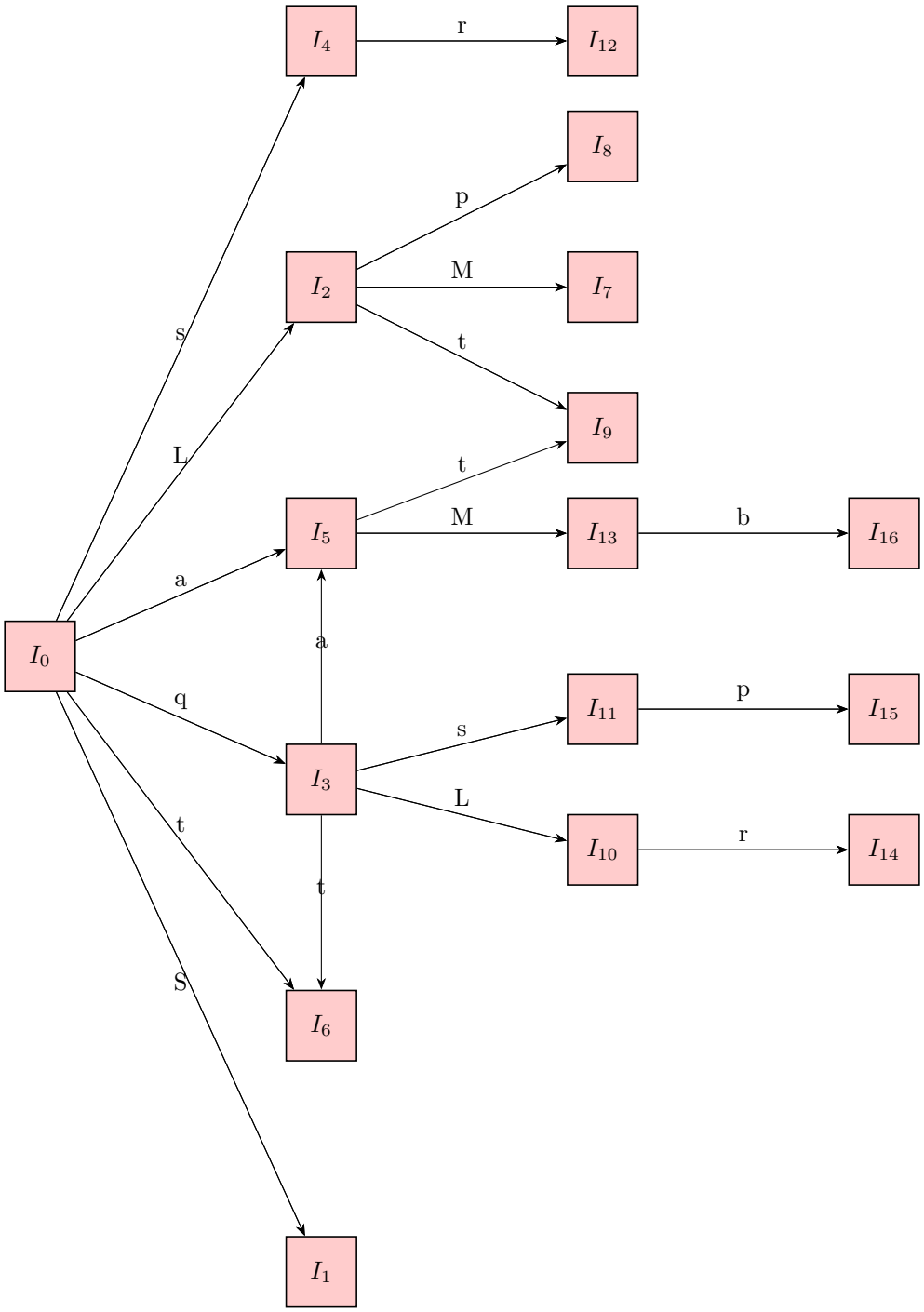
$I_{14} = GOTO(I_{10}, r) = \{[S \to qLr\cdot]\}$

$I_{15} = GOTO(I_{11}, p) = \{[S \to qsp\cdot]\}$

$I_{16} = GOTO(I_{13}, b) = \{[L \to aMb\cdot]\}$

$I_6 = GOTO(I_3, t)$
$I_5 = GOTO(I_3, a)$
$I_9 = GOTO(I_5, t)$

LR(0) Automaton



SLR(1) Table

STATE	*ACTION*								*GOTO*		
	a	**b**	**p**	**q**	**r**	**s**	**t**	**$**	**S**	**L**	**M**
0	s5			s3		s4	s6		1	2	
1								Accept			
2			s8				s9				7
3	s5					s11	s6			10	
4			r7		r7,s12		r7				
5							s9				13
6			r8		r8		r8				
7								r1			
8								r2			
9		r9						r9			
10					s14						
11			r7,s15		r7		r7				
12								r4			
13		s16									
14								r3			
15								r5			
16			r6		r6		r6				

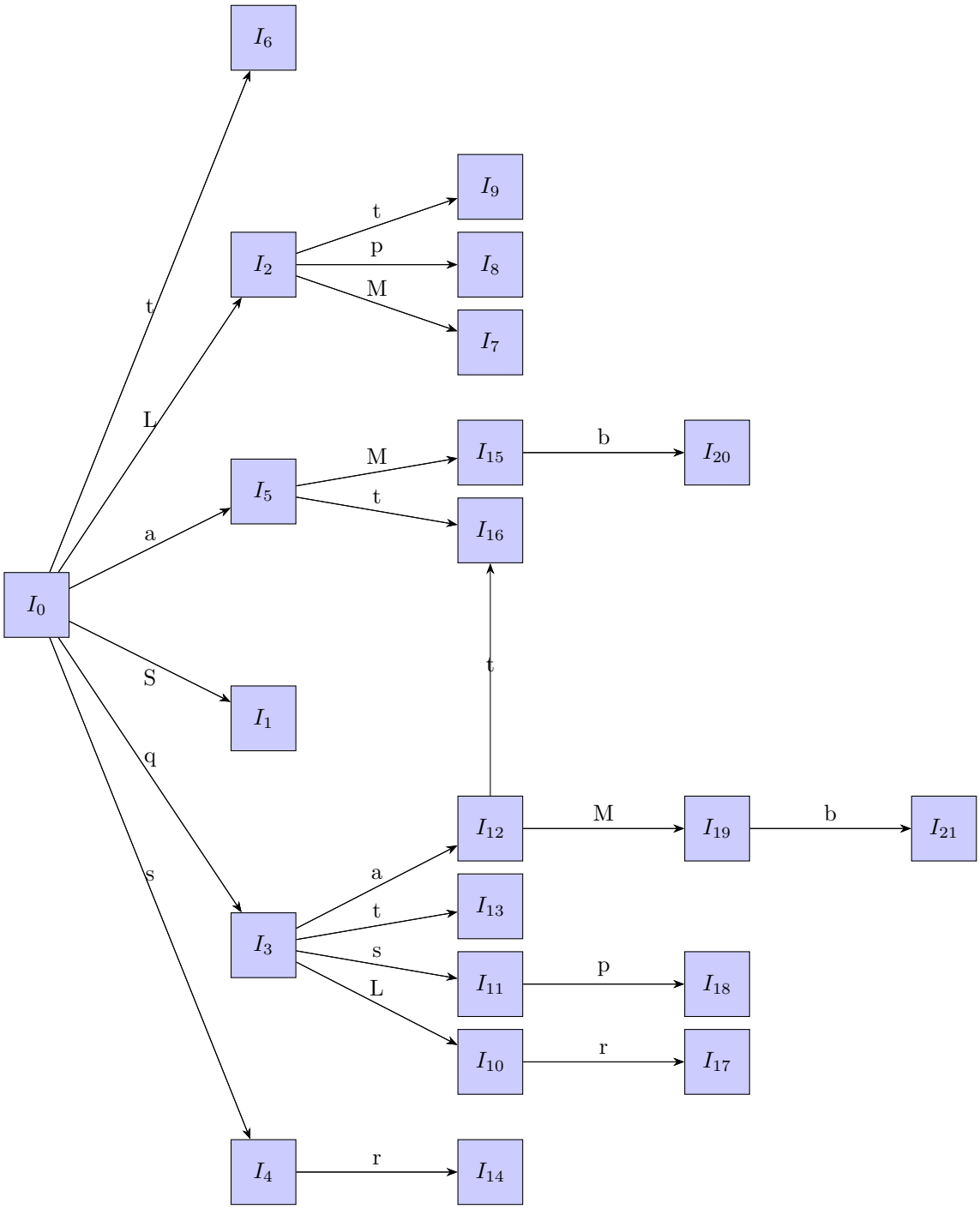
Table 1: SLR(1) Parsing Table

Hence, due to the multiply filled cells, the grammar is not SLR(1)

LR(1) Collection:-

$I_0 = Closure(\{[S' \to \cdot S, \$]\})$
$= \{[S' \to \cdot S, \$],\ [S \to \cdot Lp, \$],\ [S \to \cdot qLr, \$],\ [S \to \cdot sr, \$],\ [S \to \cdot qsp, \$],\ [L \to \cdot aMb, t/p],\ [L \to \cdot s, t/p],\ [L \to \cdot t, t/p]\}$
$I_1 = GOTO(I_0, S) = \{[S' \to S\cdot, \$]\}$
$I_2 = GOTO(I_0, L) = \{[S \to L \cdot M, \$],\ [S \to L \cdot p, \$],\ [M \to \cdot t, \$]\}$
$I_3 = GOTO(I_0, q) = \{[S \to q \cdot Lr, \$],\ [S \to q \cdot sp, \$],\ [L \to \cdot aMb, r],\ [L \to \cdot s, r],\ [L \to \cdot t, r]\}$
$I_4 = GOTO(I_0, s) = \{[S \to s \cdot r, \$],\ [L \to s\cdot, t/p]\}$
$I_5 = GOTO(I_0, a) = \{[L \to a \cdot Mb, t/p],\ [M \to \cdot t, b]\}$
$I_6 = GOTO(I_0, t) = \{[L \to t\cdot, t/p]\}$
$I_7 = GOTO(I_2, M) = \{[S \to LM\cdot, \$]\}$
$I_8 = GOTO(I_2, p) = \{[S \to Lp\cdot, \$]\}$
$I_9 = GOTO(I_2, t) = \{[M \to t\cdot, \$]\}$
$I_{10} = GOTO(I_3, L) = \{[S \to qL \cdot r, \$]\}$
$I_{11} = GOTO(I_3, s) = \{[S \to qs \cdot p, \$], [L \to s\cdot, r]\}$
$I_{12} = GOTO(I_3, a) = \{[L \to a \cdot Mb, r],\ [M \to \cdot t, b]\}$
$I_{13} = GOTO(I_3, t) = \{[L \to t\cdot, r]\}$
$I_{14} = GOTO(I_4, r) = \{[S \to sr\cdot, \$]\}$
$I_{15} = GOTO(I_5, M) = \{[L \to aM \cdot b, t/p]\}$
$I_{16} = GOTO(I_6, t) = \{[M \to t\cdot, b]\}$
$I_{17} = GOTO(I_{10}, r) = \{[S \to qLr\cdot, \$]\}$
$I_{18} = GOTO(I_{11}, p) = \{[S \to qsp\cdot, \$]\}$
$I_{19} = GOTO(I_{12}, M) = \{[L \to aM \cdot b, r]\}$
$I_{20} = GOTO(I_{15}, b) = \{[L \to aMb\cdot, t/p]\}$
$I_{21} = GOTO(I_{19}, b) = \{[L \to aMb\cdot, r]\}$
$I_{16} = GOTO(I_{12}, t)$

LR(1) Automaton:-



LR(1) Table

STATE	*ACTION*								*GOTO*		
	a	**b**	**p**	**q**	**r**	**s**	**t**	**$**	**S**	**L**	**M**
0	s5			s3		s4	s6		1	2	
1								Accept			
2			s8				s9				7
3	s12					s11	s13			10	
4			r7		s14		r7				
5							s16				15
6			r8				r8				
7								r1			
8								r2			
9								r9			
10					s17						
11			s18		r7						
12							s16				19
13					r8						
14								r4			
15		s20									
16		r9									
17								r3			
18								r5			
19		s21									
20			r6				r6				
21					r6						

Table 2: LR(1) Parsing Table

LALR(1) Table

STATE	*ACTION*								*GOTO*		
	a	**b**	**p**	**q**	**r**	**s**	**t**	**$**	**S**	**L**	**M**
0	s512			s3		s4	s613		1	2	
1								Accept			
2			s8				s9				7
3	s512					s11	s613			10	
4			r7		s14		r7				
512							s916				1519
613			r8		r8		r8				
7								r1			
8								r2			
916		r9						r9			
10					s17						
11			s18		r7						
14								r4			
1519		s2021									
17								r3			
18								r5			
2021			r6		r6						

Table 3: LALR(1) Parsing Table

As there are no multiply-filled cells, grammar is LALR(1).

Problem 3

Assumptions

- If the number of choices in a question is not 3 or 4, the question is **not considered** in the statistics of the quiz. Further, the program is terminated there only, keeping the stats only for the previously encountered valid questions.
- I have assumed that there are no stray characters in a valid tag. For ex- <choice random >

Bash Script for running the code

```
1 #!/bin/bash
2
3 bison  -d parser.y
4 flex -o lexer.c lexer.l
5 g++ parser.tab.c lexer.c -lfl
6 ./a.out < testfile.txt
```

Files Submitted

- lexer.l -> the lexer file
- parser.y -> the parser file
- script.sh -> script file

Process of running

- Instead of testfile.txt in line 6 of **script.sh**, put your test file.
- In **parser.y**, go to main(), in fopen(), write your testfile name.
- run **./script.sh** in the terminal