CS323 Asignment 3

Exercise 1

1. The string "a+a-a" is NOT a valid sentence in $L(G)$ since

$$a + a - a \Longleftarrow S + a - a \Longleftarrow S + S - a \Longleftarrow S + S - S$$

And there is no derivation can apply to the sentence.

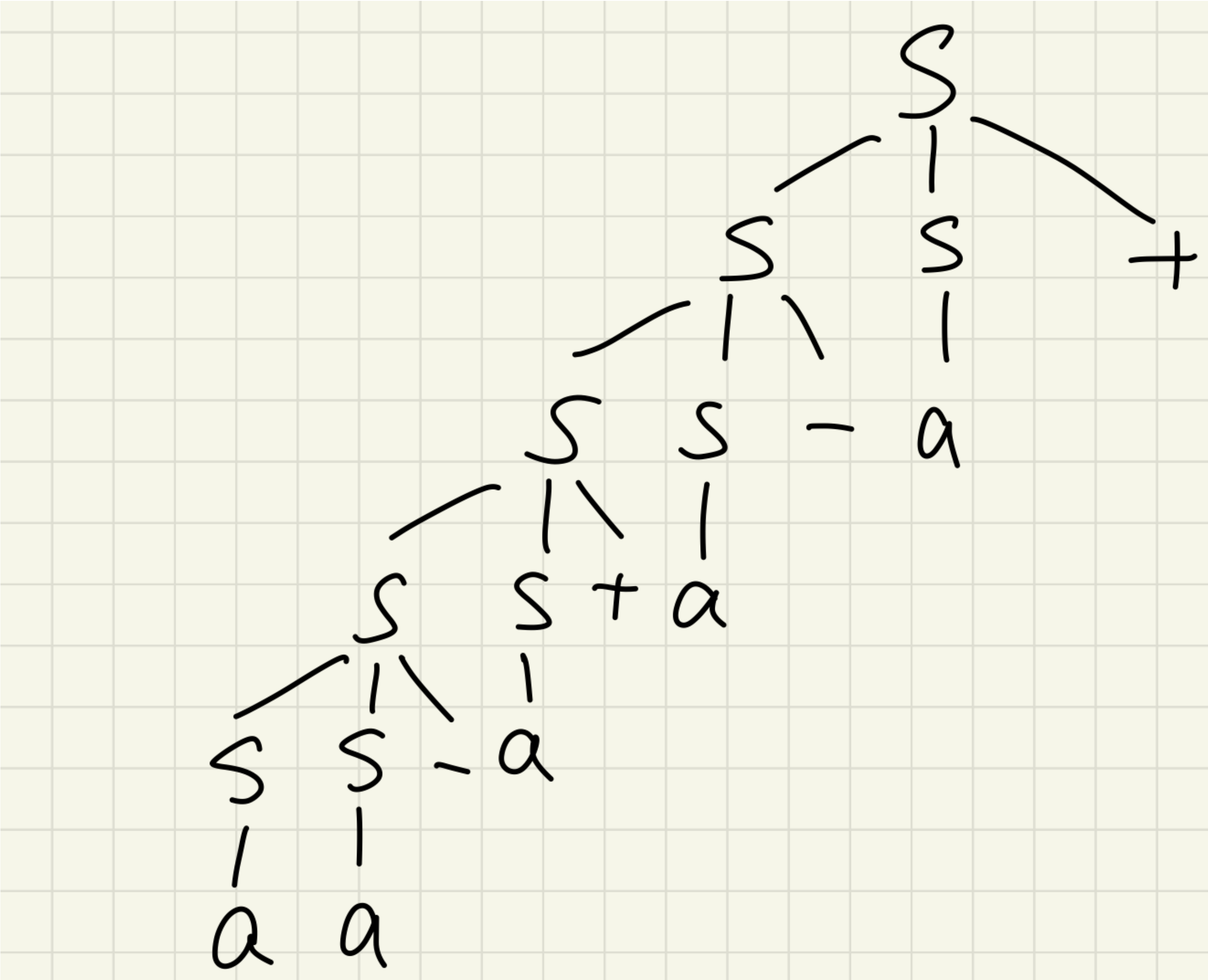
2. The leftmost derivation is as below:

$$\begin{aligned} S \Longrightarrow SS+ \Longrightarrow SS - S+ \Longrightarrow SS + S - S + \\ \Longrightarrow SS - S + S - S+ \Longrightarrow aS - S + S - S+ \Longrightarrow aa - S + S - S + \\ \Longrightarrow aa - a + S - S+ \Longrightarrow aa - a + a - S+ \Longrightarrow aa - a + a - a+ \end{aligned}$$

3. The rightmost derivation is as below:

$$\begin{aligned} S \Longrightarrow SS+ \Longrightarrow Sa+ \Longrightarrow SS - a + \\ \Longrightarrow Sa - a+ \Longrightarrow SS + a - a+ \Longrightarrow Sa + a - a + \\ \Longrightarrow SS - a + a - a+ \Longrightarrow Sa - a + a - a+ \Longrightarrow aa - a + a - a+ \end{aligned}$$

4. The parse tree is as below:



Exercise 2

1.

Since $S \to aB$, then $a \in FIRST(S)$. Since $B \to S * B|\epsilon$ and $S \to aB$, then $a \in FIRST(B), \epsilon \in FIRST(B)$. So

$$FIRST(S) = \{a\},$$
$$FIRST(B) = \{a, \epsilon\}$$

Add $\$$ into $FOLLOW(S)$ and $FOLLOW(B)$. Since $S \to aB$, then $FOLLOW(S) \subset FOLLOW(B)$ except ϵ. Since $B \to S * B|\epsilon$, then $* \in FOLLOW(S)$. So

$$FOLLOW(S) = \{\$, *\},$$
$$FOLLOW(B) = \{\$, *\}$$

For $S \rightarrow aB$, $FIRST(aB) = FIRST(a) = \{a\}$

For $B \rightarrow S * B$, $FIRST(S * B) = FIRST(S) = \{a\}$

For $B \rightarrow \epsilon$, $FIRST(\epsilon) = \{\epsilon\}$, then $FOLLOW(B) = \{\$, *\}$

The predictive parsing table is as below:

NON-TERMINAL	a	*	$
S	$S \rightarrow aB$		
B	$B \rightarrow S * B$	$B \rightarrow \epsilon$	$B \rightarrow \epsilon$

2. The grammar is LL(1) since there is no entries with multiple productions.
3. Yes.

Input: string $aaaa * **$, the parsing table M. The steps is as below.

MATCHED	STACK	INPUT	ACTION
	S$	aaaa***$	
	aB$	aaaa***$	output $S \rightarrow aB$
a	B$	aaa***$	match a
a	S*B$	aaa***$	output $B \rightarrow S * B$
a	aB*B$	aaa***$	output $S \rightarrow aB$
aa	B*B$	aa***$	match a
aa	S*B*B$	aa***$	output $B \rightarrow S * B$
aa	aB*B*B$	aa***$	output $S \rightarrow aB$
aaa	B*B*B$	a***$	match a
aaa	S*B*B*B$	a***$	output $B \rightarrow S * B$
aaa	aB*B*B*B$	a***$	output $S \rightarrow aB$
aaaa	B*B*B*B$	***$	match a
aaaa	*B*B*B$	***$	output $B \rightarrow \epsilon$

MATCHED	STACK	INPUT	ACTION
aaaa*	B*B*B$	**$	match *
aaaa*	*B*B$	**$	output $B \to \epsilon$
aaaa**	B*B$	*$	match *
aaaa**	*B$	*$	output $B \to \epsilon$
aaaa***	B$	$	match *
aaaa***	$	$	output $B \to \epsilon$

Exercise 3

Grammar G (code the derivations):

$$(1) S \to aB$$
$$(2) B \to S * B$$
$$(3) B \to \epsilon$$

Augmented grammar G':

$$S' \to S$$
$$S \to aB$$
$$B \to S * B | \epsilon$$

Items:

$$S' \to \cdot S, S' \to S \cdot$$
$$S \to \cdot aB, S \to a \cdot B, S \to aB \cdot$$
$$B \to \cdot S * B, B \to S \cdot * B, B \to S * \cdot B, B \to S * B \cdot$$
$$B \to \cdot$$

$FIRST$ and $FOLLOW$:

$$FIRST(S) = \{a\},$$
$$FIRST(B) = \{a, \epsilon\};$$
$$FOLLOW(S) = \{\$, *\},$$
$$FOLLOW(B) = \{\$, *\}$$

1.

(1) SLR

Calculation of canonical LR(0) collection:

- Initial state: $C = \{I_0\} = \{CLOSURE(\{[S' \to \cdot S]\})\} = \{[S' \to \cdot S], [S \to \cdot aB]\}$.
- Iteration for item set $I_0 = \{[S' \to \cdot S], [S \to \cdot aB]\}$:
  - for grammar symbol S, $GOTO(I_0, S) = CLOSURE(\{[S' \to S\cdot]\}) = \{[S' \to S\cdot]\}$, which is not in C and thus named I_1 and added into C.
  - for grammar symbol a, $GOTO(I_0, a) = CLOSURE(\{[S \to a \cdot B]\}) = \{[S \to a \cdot B], [B \to \cdot S * B], [B \to \cdot]\}$, which is not in C and thus named I_2 and added into C.
- Iteration for item set $I_1 = \{[S' \to S\cdot]\}$:
  - No $GOTO$
- Iteration for item set $I_2 = \{[S \to a \cdot B], [B \to \cdot S * B], [B \to \cdot]\}$
  - for grammar symbol B, $GOTO(I_2, B) = CLOSURE(\{[S \to aB\cdot]\}) = \{[S \to aB\cdot]\}$, which is not in C and thus named I_3 and added into C.
  - for grammar symbol S, $GOTO(I_2, S) = CLOSURE(\{[B \to S \cdot *B]\}) = \{[B \to S \cdot *B]\}$, which is not in C and thus named I_4 and added into C.
- Iteration for item set $I_3 = \{[S \to aB\cdot]\}$:
  - No $GOTO$
- Iteration for item set $I_4 = \{[B \to S \cdot *B]\}$
  - for grammar symbol $*$, $GOTO(I_4, *) = CLOSURE(\{[B \to S * \cdot B]\}) = \{[B \to S * \cdot B], [B \to \cdot S * B], [B \to \cdot]\}$, which is not in C and thus named I_5 and added into C.
- Iteration for item set $I_5 = \{[B \to S * \cdot B], [B \to \cdot S * B], [B \to \cdot]\}$
  - for grammar symbol B, $GOTO(I_5, B) = CLOSURE(\{[B \to S * B\cdot]\}) = \{[B \to S * B\cdot]\}$, which is not in C and thus named I_6 and added into C.
  - for grammar symbol S, $GOTO(I_5, S) = CLOSURE(\{[B \to S \cdot *B]\}) = \{[B \to S \cdot *B]\} = I_4$
- Iteration for item set $I_6 = \{[B \to S * B\cdot]\}$
  - No $GOTO$

After all, the canonical LR(0) collection is as below:

SET	ITEMS
I_0	$[S' \to \cdot S], [S \to \cdot aB]$
I_1	$[S' \to S\cdot]$
I_2	$[S \to a \cdot B], [B \to \cdot S * B], [B \to \cdot]$
I_3	$[S \to aB\cdot]$

SET	ITEMS
I_4	$[B \to S \cdot *B]$
I_5	$[B \to S * \cdot B], [B \to \cdot S * B], [B \to \cdot]$
I_6	$[B \to S * B \cdot]$

And the function $GOTO$ is defined as below:

$$\begin{gathered} GOTO(I_0, S) = I_1, GOTO(I_0, a) = I_2, \\ GOTO(I_2, B) = I_3, GOTO(I_2, S) = I_4, \\ GOTO(I_4, *) = I_5, \\ GOTO(I_5, B) = I_6, GOTO(I_5, S) = I_4 \end{gathered}$$

For $accept$,

- Since $[S' \to S \cdot] \in I_1$, $ACTION[1, \$] = acc$.

For $shift$,

- Since $[S \to \cdot aB] \in I_0$ and $GOTO(I_0, a) = I_2$, $ACTION[0, a] = s2$.
- Since $[B \to S \cdot *B] \in I_4$ and $GOTO(I_4, *) = I_5$, $ACTION[4, *] = s5$.

For $reduce$,

- Since $[B \to \cdot] \in I_2$ and $FOLLOW(B) = \{\$, *\}$, $ACTION[2, \$] = ACTION[2, *] = reduce\ B \to \epsilon\ (r3)$.
- Since $[S \to aB \cdot] \in I_3$ and $FOLLOW(S) = \{\$, *\}$, $ACTION[3, \$] = ACTION[3, *] = reduce\ S \to aB\ (r1)$.
- Since $[B \to \cdot] \in I_5$ and $FOLLOW(B) = \{\$, *\}$, $ACTION[5, \$] = ACTION[5, *] = reduce\ B \to \epsilon\ (r3)$.
- Since $[B \to S * B \cdot] \in I_6$ and $FOLLOW(B) = \{\$, *\}$, $ACTION[6, \$] = ACTION[6, *] = reduce\ B \to S * B\ (r2)$.

Then we can construct the SLR parsing table:

STATE	ACTION			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2		r3	r3	4	3
3		r1	r1		
4		s5			
5		r3	r3	4	6
6		r2	r2		

(2) CLR

Calculation of canonical LR(1) collection:

- Initial state: $C = \{I_0\} = \{CLOSURE(\{[S' \to \cdot S, \$]\})\} = \{[S' \to \cdot S, \$], [S \to \cdot aB, \$]\}$.
- Iteration for item set $I_0 = \{[S' \to \cdot S, \$], [S \to \cdot aB, \$]\}$:
  - for grammar symbol S, $GOTO(I_0, S) = CLOSURE(\{[S' \to S\cdot, \$]\}) = \{[S' \to S\cdot, \$]\}$, which is not in C and thus named I_1 and added into C.
  - for grammar symbol a, $GOTO(I_0, a) = CLOSURE(\{[S \to a \cdot B, \$]\}) = \{[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]\}$, which is not in C and thus named I_2 and added into C.
- Iteration for item set $I_1 = \{[S' \to S\cdot, \$]\}$:
  - No $GOTO$
- Iteration for item set $I_2 = \{[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]\}$
  - for grammar symbol B, $GOTO(I_2, B) = CLOSURE(\{[S \to aB\cdot, \$]\}) = \{[S \to aB\cdot, \$]\}$, which is not in C and thus named I_3 and added into C.
  - for grammar symbol S, $GOTO(I_2, S) = CLOSURE(\{[B \to S \cdot *B, \$]\}) = \{[B \to S \cdot *B, \$]\}$, which is not in C and thus named I_4 and added into C.
- Iteration for item set $I_3 = \{[S \to aB\cdot, \$]\}$:
  - No $GOTO$
- Iteration for item set $I_4 = \{[B \to S \cdot *B, \$]\}$
  - for grammar symbol $*$, $GOTO(I_4, *) = CLOSURE(\{[B \to S * \cdot B, \$]\}) = \{[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]\}$, which is not in C and thus named I_5 and added into C.
- Iteration for item set $I_5 = \{[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]\}$
  - for grammar symbol B, $GOTO(I_5, B) = CLOSURE(\{[B \to S * B\cdot, \$]\}) = \{[B \to S * B\cdot, \$]\}$, which is not in C and thus named I_6 and added into C.

    - for grammar symbol S, $GOTO(I_5, S) = CLOSURE(\{[B \to S \cdot *B, \$]\}) = \{[B \to S \cdot *B, \$]\} = I_4$
- Iteration for item set $I_6 = \{[B \to S * B\cdot, \$]\}$
    - No $GOTO$

After all, the canonical LR(1) collection is as below:

SET	ITEMS
I_0	$[S' \to \cdot S, \$], [S \to \cdot aB, \$]$
I_1	$[S' \to S\cdot, \$]$
I_2	$[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_3	$[S \to aB\cdot, \$]$
I_4	$[B \to S \cdot *B, \$]$
I_5	$[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_6	$[B \to S * B\cdot, \$]$

And the function $GOTO$ is defined as below:

$$GOTO(I_0, S) = I_1, GOTO(I_0, a) = I_2,\\ GOTO(I_2, B) = I_3, GOTO(I_2, S) = I_4,\\ GOTO(I_4, *) = I_5,\\ GOTO(I_5, B) = I_6, GOTO(I_5, S) = I_4$$

For $accept$,

- Since $[S' \to S\cdot, \$] \in I_1$, $ACTION[1, \$] = acc$.

For $shift$,

- Since $[S \to \cdot aB, \$] \in I_0$ and $GOTO(I_0, a) = I_2$, $ACTION[0, a] = s2$.
- Since $[B \to S \cdot *B, \$] \in I_4$ and $GOTO(I_4, *) = I_5$, $ACTION[4, *] = s5$.

For $reduce$,

- Since $[B \to \cdot, \$] \in I_2$, $ACTION[2, \$] = reduce\ B \to \epsilon\ (r3)$.
- Since $[S \to aB\cdot, \$] \in I_3$, $ACTION[3, \$] = reduce\ S \to aB\ (r1)$.
- Since $[B \to \cdot, \$] \in I_5$, $ACTION[5, \$] = reduce\ B \to \epsilon\ (r3)$.
- Since $[B \to S * B\cdot, \$] \in I_6$, $ACTION[6, \$] = reduce\ B \to S * B\ (r2)$.

Then we can construct the CLR parsing table:

STATE	ACTION			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2			r3	4	3
3			r1		
4		s5			
5			r3	4	6
6			r2		

(3) LALR

From the above, we can get the canonical LR(1) collection is as below:

SET	ITEMS
I_0	$[S' \to \cdot S, \$], [S \to \cdot aB, \$]$
I_1	$[S' \to S\cdot, \$]$
I_2	$[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_3	$[S \to aB\cdot, \$]$
I_4	$[B \to S \cdot *B, \$]$
I_5	$[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_6	$[B \to S * B\cdot, \$]$

And the function $GOTO$ is defined as below:

$$GOTO(I_0, S) = I_1, GOTO(I_0, a) = I_2,$$
$$GOTO(I_2, B) = I_3, GOTO(I_2, S) = I_4,$$
$$GOTO(I_4, *) = I_5,$$
$$GOTO(I_5, B) = I_6, GOTO(I_5, S) = I_4$$

Combining the canonical LR(1) collection, we can get the new LR(1) collection as (actually there is no need to combine)

SET	ITEMS
I_0	$[S' \to \cdot S, \$], [S \to \cdot aB, \$]$
I_1	$[S' \to S\cdot, \$]$
I_2	$[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_3	$[S \to aB\cdot, \$]$
I_4	$[B \to S \cdot *B, \$]$
I_5	$[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$]$
I_6	$[B \to S * B\cdot, \$]$

Then we can construct the LALR parsing table:

STATE	ACTION			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2			r3	4	3
3			r1		
4		s5			
5			r3	4	6
6			r2		

2. The grammar is SLR(1), LR(1) and LALR(1). All because it can construct the parsing table and there is no conflict.
3. No. It will enter the error state.

Stack	Symbol	Input	Action
0		aaaa***$	shift
0 2	a	aaa***$	shift
			error

Optional Exercise

1. Yes. The leftmost derivation is as below:

$$
\begin{gathered}
Phrase \implies Phrase\,Verb\,Phrase \implies Human\,Verb\,Phrase \\
\implies Tom\,Verb\,Phrase \implies Tom\ like\ Phrase \\
\implies Tom\ like\ Animal \implies Tom\ like\ dog
\end{gathered}
$$

Since the right part of the non-terminals' derivations are exclusive, i.e., for any given non-terminals, they must derive different strings, and for any given strings, they must be derived from different derivation paths. Therefore, the grammar is not anbiguous.

2. Grammar G:

$$S \to SS + | SS - | a$$

Since it is immediate left recursion, we can apply the method for eliminating immediate left recursion and get the result as below.

$$
\begin{gathered}
S \to aS' \\
S' \to S + S' | S - S' | \epsilon
\end{gathered}
$$

By left factoring, we have

$$
\begin{gathered}
S \to aS' \\
S' \to ST | \epsilon \\
T \to +S' | -S'
\end{gathered}
$$