CS323 Compilers

Homework #3

Site Fan
fanst2021@mail.sustech.edu.cn

Exercise 1 (Grammar Basics)

Consider the following context-free grammar G:

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

1. Is the string "$a + a - a$" a valid sentence in $L(G)$?
2. Give a leftmost derivation for the string $aa - a + a - a+$.
3. Give a rightmost derivation for the string $aa - a + a - a+$.
4. Give a parse tree for the string $aa - a + a - a+$.

Solution 1.

No. There is no derivation producing $a + a - a$.

According to the grammar, "+" and "-" appear at the tail of the production body, i.e., any sentence in $L(G)$ containing "+" or "-" ends with "+" or "-", but the given sentence ends with "a".

Therefore, this string is not a valid sentence in $L(G)$.

Solution 2.

$S \Rightarrow SS+ \Rightarrow SS - S+ \Rightarrow SS + S - S+ \Rightarrow SS - S + S - S+ \Rightarrow aS - S + S - S+$
$\Rightarrow aa - S + S - S+ \Rightarrow aa - a + S - S+ \Rightarrow aa - a + a - S+ \Rightarrow aa - a + a - a+$

Solution 3.

$S \Rightarrow SS+ \Rightarrow Sa+ \Rightarrow SS - a+ \Rightarrow Sa - a+ \Rightarrow SS + a - a+ \Rightarrow Sa + a - a+$
$\Rightarrow SS - a + a - a+ \Rightarrow Sa - a + a - a+ \Rightarrow aa - a + a - a+$

Solution 4.

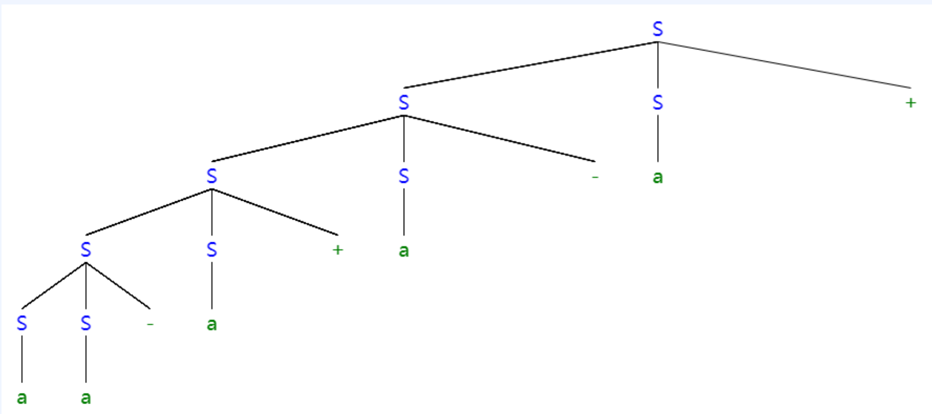

Figure 1: Parse Tree for string "aa-a+a-a+".

Exercise 2 (Top-Down Parsing)

Consider the following grammar G:

$$S \to aB$$
$$B \to S * B \mid \epsilon$$

1. Construct the predictive parsing table for G. Please put down the detailed steps, including the calculation of FIRST and FOLLOW sets.

2. Is the grammar LL(1)?
3. Can an LL(1) parser accept the input string $aaaa***\$$? If yes, please list the moves made by the parser; otherwise, state the reason. Before parsing, please resolve conflicts in the parsing table if any.

Solution 1.

Calculate FIRST sets:

1. For terminals, $\text{FIRST}(a) = \{a\}, \text{FIRST}(*) = \{*\}$.
2. From $S \to aB$, $a \in \text{FIRST}(S)$, so $\text{FIRST}(S) = \{a\}$.
3. From $B \to \epsilon$, $\epsilon \in \text{FIRST}(B)$.
4. From $B \to S * B$, $\text{FIRST}(S) \subseteq \text{FIRST}(B)$, so $\text{FIRST}(B) = \{a, \epsilon\}$

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

Calculate FOLLOW sets:

1. For begin symbol S, $\$ \in \text{FOLLOW}(S)$.
2. From $B \to S * B$, $* \in \text{FOLLOW}(S)$.
3. From $S \to aB$, $\text{FOLLOW}(S) \subseteq \text{FOLLOW}(B)$.

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

Then fill the parsing table using FIRST and FOLLOW:

Nonterminal	a	$*$	$
S	$S \to aB$		
B	$B \to S * B$	$B \to \epsilon$	$B \to \epsilon$

Solution 2.

For $B \to S * B \mid \epsilon$,

1. $\text{FIRST}(S * B) \cap \text{FIRST}(\epsilon) = \emptyset$.
2. $\text{FIRST}(\epsilon) = \epsilon$, and $\text{FIRST}(S * B) \cap \text{FOLLOW}(B) = \emptyset$.

There is a unique choice of production at each step by looking ahead, therefore G is LL(1).

Solution 3.

Yes. The moves made by an LL(1) parser are listed as follows:

Matched	Stack	Input	Action
	$S\$$	$aaaa***\$$	
	$aB\$$	$aaaa***\$$	Output $S \to aB$
a	$B\$$	$aaa***\$$	Match a
a	$S * B\$$	$aaa***\$$	Output $B \to S * B$
a	$aB * B\$$	$aaa***\$$	Output $S \to aB$
aa	$B * B\$$	$aa***\$$	Match a

aa	$S * B * B\$$	$aa * * * \$$	Output $B \to S * B$
aa	$aB * B * B\$$	$aa * * * \$$	Output $S \to aB$
aaa	$B * B * B\$$	$a * * * \$$	Match a
aa	$S * B * B * B\$$	$a * * * \$$	Output $B \to S * B$
aa	$aB * B * B * B\$$	$a * * * \$$	Output $S \to aB$
$aaaa$	$B * B * B * B\$$	$* * * \$$	Match a
$aaaa$	$* B * B * B\$$	$* * * \$$	Output $B \to \epsilon$
$aaaa *$	$B * B * B\$$	$* * \$$	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 (Bottom-Up Parsing)

1. Construct the shift-reduce parsing table for the above grammar G using each of the following algorithms: (1) SLR, (2) CLR, and (3) LALR. Please put down the detailed steps, including the calculation of item sets. For the calculation of closures, GOTO targets, and FIRST/FOLLOW sets, you may choose not to put down the details.
2. Is the grammar SLR(1)? Is the grammar LR(1)? Is the grammar LALR(1)?
3. Can an LALR(1) parser accept the input string $aaaa * * * \$$? If yes, please list the moves made by the parser; otherwise, state the reason. Before parsing, please resolve conflicts in the parsing table if any.

Solution 1: SLR

1. **Augmented Grammar**

$$
\begin{aligned}
S' &\to S && (1)\\
S &\to aB && (2)\\
B &\to S * B && (3)\\
B &\to \epsilon && (4)
\end{aligned}
$$

2. **FIRST and FOLLOW**

	S'	S	B
FIRST	$\{a\}$	$\{a\}$	$\{a, \epsilon\}$
FOLLOW	$\{\$, *\}$	$\{\$, *\}$	$\{\$, *\}$

3. **Calculate the CLOSURE of item sets and GOTO**

Initial item set I_0:

Initially, $[S' \to \cdot S]$ is in I_0.

$I_0 = \text{CLOSURE}\{[S' \to \cdot S]\} = \{[S' \to \cdot S], [S \to \cdot aB]\}$

GOTO for I_0:

$\text{GOTO}(I_0, S) = \text{CLOSURE}\{[S' \to S \cdot]\} = I_1$
$\text{GOTO}(I_0, a) = \text{CLOSURE}\{[S \to a \cdot B]\}$

$$= \{[S \to a \cdot B], [B \to \cdot S * B], [B \to \cdot], [S \to \cdot aB]\} \text{ GOTO}(I_0, a)$$
$$= I_2$$

GOTO for I_1:

$\text{GOTO}(I_1, \$) = \text{accept}$

GOTO for I_2:

$\text{GOTO}(I_2, S) = \text{CLOSURE}\{[B \to S \cdot * B]\} = \{[B \to S \cdot * B]\} = I_3$

$\text{GOTO}(I_2, B) = \text{CLOSURE}\{[S \to aB \cdot]\} = \{[S \to aB \cdot]\} = I_4$

$\text{GOTO}(I_2, a) = \text{CLOSURE}\{[S \to a \cdot B]\} = I_2$

GOTO for I_3:
$\text{GOTO}(I_3, *) = \text{CLOSURE}\{[B \to S * \cdot B]\}$

$$= \{[B \to S * \cdot B], [B \to \cdot S * B], [B \to \cdot], [S \to \cdot aB]\}$$
$$= I_5$$

GOTO for I_5:

$\text{GOTO}(I_5, S) = \text{CLOSURE}\{[B \to S \cdot * B]\} = I_3$

$\text{GOTO}(I_5, B) = \text{CLOSURE}\{[B \to S * B \cdot]\} = \{[B \to S * B \cdot]\} = I_6$

$\text{GOTO}(I_5, a) = \text{CLOSURE}\{[S \to a \cdot B]\} = I_2$

4. **Parsing Table**

State	Action			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2	s2	r4	r4	3	4
3		s5			
4		r2	r2		
5	s2	r4	r4	3	6
6		r3	r3		

Figure 2: SLR Parsing Table

Solution 1: CLR

Step 1 and 2 are the same.

3. **Calculate the CLOSURE of item sets and GOTO**

Initial item set I_0:

Initially, $[S' \to \cdot\, S, \$]$ is in I_0.

$I_0 = \text{CLOSURE}\{[S' \to \cdot\, S, \$]\} = \{[S' \to \cdot\, S, \$], [S \to \cdot\, aB, \$]\}$

GOTO for I_0:

$$
\begin{aligned}
\text{GOTO}(I_0, S) &= \text{CLOSURE}\{[S' \to S\, \cdot, \$]\} = I_1 \\
\text{GOTO}(I_0, a) &= \text{CLOSURE}\{[S \to a \cdot B, \$]\} \\
&= \{[S \to a \cdot B, \$], [B \to \cdot\, S * B, \$], [B \to \cdot, \$], [S \to \cdot\, aB, *]\} \\
&= I_2
\end{aligned}
$$

GOTO for I_1:

$\text{GOTO}(I_1, \$) = \text{accept}$

GOTO for I_2:

$\text{GOTO}(I_2, S) = \text{CLOSURE}\{[B \to S \cdot * B, \$]\} = \{[B \to S \cdot * B, \$]\} = I_3$

$$
\begin{aligned}
\text{GOTO}(I_2, B) &= \text{CLOSURE}\{[S \to aB\, \cdot, \$]\} = \{[S \to aB\, \cdot, \$]\} = I_4 \\
\text{GOTO}(I_2, a) &= \text{CLOSURE}\{[S \to a \cdot B, *]\} \\
&= \{[S \to a \cdot B, *], [B \to \cdot\, S * B, *], [B \to \cdot, *], [S \to \cdot\, aB, *]\} \\
&= I_7
\end{aligned}
$$

GOTO for I_3:

$$
\begin{aligned}
\text{GOTO}(I_3, *) &= \text{CLOSURE}\{[B \to S * \cdot B, \$]\} \\
&= \{[B \to S * \cdot B, \$], [B \to \cdot\, S * B, \$], [B \to \cdot, \$], [S \to \cdot\, aB, *]\} \\
&= I_5
\end{aligned}
$$

GOTO for I_5:

$\text{GOTO}(I_5, S) = \text{CLOSURE}\{[B \to S \cdot * B, \$]\} = I_3$

$\text{GOTO}(I_5, B) = \text{CLOSURE}\{[B \to S * B\, \cdot, \$]\} = \{[B \to S * B\, \cdot, \$]\} = I_6$

$\text{GOTO}(I_5, a) = \text{CLOSURE}\{[B \to S * B\, \cdot, *]\} = I_7$

GOTO for I_7:

$\text{GOTO}(I_7, S) = \text{CLOSURE}\{[B \to S \cdot * B, *]\} = \{[B \to S \cdot * B, *]\} = I_9$

$\text{GOTO}(I_7, B) = \text{CLOSURE}\{[B \to aB\, \cdot, *]\} = \{[B \to aB\, \cdot, *]\} = I_8$

$\text{GOTO}(I_7, a) = \text{CLOSURE}\{[B \to \cdot\, S * B, *]\} = I_7$

GOTO for I_9:

$$
\begin{aligned}
\text{GOTO}(I_9, *) &= \text{CLOSURE}\{[B \to S * \cdot B, *]\} \\
&= \{[B \to S * \cdot B, *], [B \to \cdot\, S * B, *], [B \to \cdot, *], [B \to \cdot\, aB, *]\} \\
&= I_{10}
\end{aligned}
$$

GOTO for I_{10}:

$\text{GOTO}(I_7, S) = \text{CLOSURE}\{[B \to \cdot\, S * B, *]\} = I_7$

$\text{GOTO}(I_7, B) = \text{CLOSURE}\{[B \to S \cdot * B, *]\} = I_9$

$\text{GOTO}(I_7, a) = \text{CLOSURE}\{[B \to S * B\,\cdot, *]\} = \{[B \to S * B\,\cdot, *]\} = I_{11}$

4. **Parsing Table**

State	Action			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2	s7		r4	3	4
3		s5			
4			r2		
5	s7		r4	3	6
6			r3		
7	s7	r4		9	8
8		r2			
9		s10			
10	s7	r2		9	11
11			r3		

Figure 3: CLR Parsing Table

Solution 1: LALR

Step 1 and 2 are the same.

3. **Calculate the CLOSURE of item sets and GOTO**

Initial item set I_0:

Initially, $[S' \to \cdot\, S, \$]$ is in I_0.

$I_0 = \text{CLOSURE}\{[S' \to \cdot\, S, \$]\} = \{[S' \to \cdot\, S, \$], [S \to \cdot\, aB, \$]\}$

GOTO for I_0:

$\text{GOTO}(I_0, S) = \text{CLOSURE}\{[S' \to S\,\cdot, \$]\} = I_1$

$$\begin{aligned}\text{GOTO}(I_0, a) &= \text{CLOSURE}\{[S \to a \cdot B, \$/*]\}\\ &= \{[S \to a \cdot B, \$/*], [B \to \cdot\, S * B, \$/*], [B \to \cdot, \$/*], [S \to \cdot\, aB, *]\}\\ &= I_2\end{aligned}$$

GOTO for I_1:

$\text{GOTO}(I_1, \$) = \text{accept}$

GOTO for I_2:

$\text{GOTO}(I_2, a) = \text{CLOSURE}\{[S \to a \cdot B, *]\} = I_2$

$\text{GOTO}(I_2, B) = \text{CLOSURE}\{[S \to aB\,\cdot, \$/*]\} = \{[S \to aB\,\cdot, \$/*]\} = I_3$

$\text{GOTO}(I_2, S) = \text{CLOSURE}\{[B \to S \cdot * B, \$/*]\} = \{[B \to S \cdot * B, \$/*]\} = I_4$

GOTO for I_4:

$$\begin{aligned}\text{GOTO}(I_4, *) &= \text{CLOSURE}\{[B \to S * \cdot B, \$/*]\} \\ &= \{[B \to S \cdot * B, \$/*], [B \to \cdot S * B, \$/*], [B \to \cdot, \$/*], [S \to \cdot aB, *]\} \\ &= I_5\end{aligned}$$

GOTO for I_5:

$\text{GOTO}(I_5, a) = \text{CLOSURE}\{[S \to a \cdot B, \$/*]\} = I_2$

$\text{GOTO}(I_5, S) = \text{CLOSURE}\{[B \to S \cdot * B, \$/*]\} = I_4$

$\text{GOTO}(I_5, B) = \text{CLOSURE}\{[B \to S * B \cdot, \$/*]\} = \{[B \to S * B \cdot, \$/*]\} = I_6$

4. **Parsing Table**

State	Action			GOTO	
	a	*	$	S	B
0	s2			1	
1			acc		
2	s2	r4	r4	4	3
3		r2	r2		
4		s5			
5	s2	r4	r4	4	6
6		r3	r3		

Figure 4: LALR Parsing Table

Solution 2.

Since there's no conflict, the grammar is SLR(1), LR(1) and LALR(1).

Solution 3.

The LALR(1) parser can accpect this string.

Stack	Symbols	Input	Action
0	$	aaaa***$	shift to 2
02	$a	aaa***$	shift to 2
022	$aa	aa***$	shift to 2
0222	$aaa	a***$	shift to 2
02222	$aaaa	***$	reduce by $B \to \epsilon$
022223	$aaaaB	***$	reduce by $S \to aB$
02224	$aaaS	***$	shift to 5
022245	$aaaS*	**$	reduce by $B \to \epsilon$
0222456	$aaaS*B	**$	reduce by $B \to S * B$
02223	$aaaB	**$	reduce by $S \to aB$

0224	$aaS	**$	shift to 5
02245	$aaS*	*$	reduce by $B \to \epsilon$
022456	$aaS*B	*$	reduce by $B \to S * B$
0223	$aaB	*$	reduce by $S \to aB$
024	$aS	*$	shift to 5
0245	$aS*	$	reduce by $B \to \epsilon$
02456	$aS*B	$	reduce by $B \to S * B$
023	$aB	$	reduce by $S \to aB$
01	$S	$	accept

Extra Exercise 1

Consider the following context-free grammar:

$$\text{Phrase} \to \text{Human} \mid \text{Animal} \mid \text{Phrase Verb Phrase}$$

$$\text{Verb} \to \text{like} \mid \text{hate}$$

$$\text{Human} \to \text{Tom} \mid \text{Jerry} \mid \text{Spike}$$

$$\text{Animal} \to \text{cat} \mid \text{mouse} \mid \text{dog}$$

The grammar can produce sentences such as "Tom like dog". Is the grammar ambiguous? Why?

Solution

Yes, it is ambiguous because for some sentence, there are multiple parse tree, here is an example:

Tom like Jerry like Spike

There are two different parse trees corresponding to the same sentence:

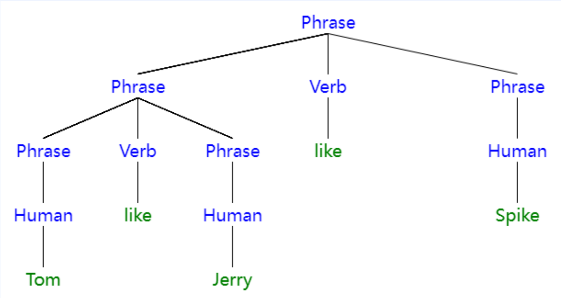

Figure 5: Possible Parse Tree 1

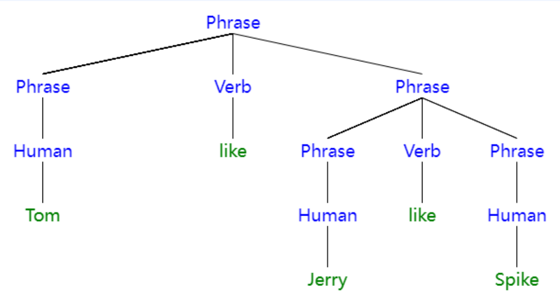

Figure 6: Possible Parse Tree 2

Therefore, this grammar is ambiguous.

Extra Exercise 2

For the grammar G in Exercise 1, give an equivalent grammar without immediate left recursions.

Solution

$$S \to aS'$$
$$S' \to S + S' | S - S' | \epsilon$$