CS323 Compiler Assignment 3

12110644 周思呈

Exercise 1 (Grammar Basics)

Consider the following context-free grammar G: S→SS+ |SS− |a

1. Is the string “a + a − a” a valid sentence in L(G)? [3 points]

No. L(G) can not end up with "-a".

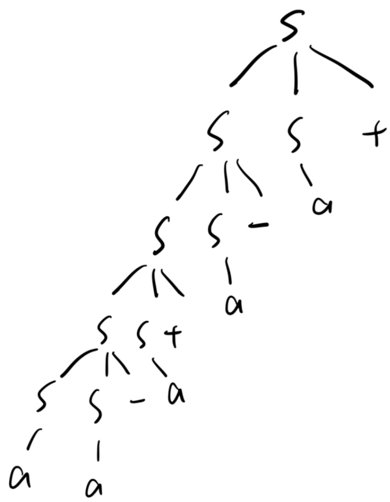
2. Give a leftmost derivation for the string aa − a + a − a+. [3 points]

$$S \underset{lm}{\Rightarrow} SS+ \underset{lm}{\Rightarrow} SS-S+ \underset{lm}{\Rightarrow} SS+S-S+ \underset{lm}{\Rightarrow} SS-S+S-S+ \underset{lm}{\Rightarrow} aa-a+a-a+ \tag{5}$$

3. Give a rightmost derivation for the string aa − a + a − a+. [3 points]

$$\begin{aligned} S &\underset{rm}{\Rightarrow} SS+ \underset{rm}{\Rightarrow} Sa+ \underset{rm}{\Rightarrow} SS-a+ \underset{rm}{\Rightarrow} Sa-a+ \underset{rm}{\Rightarrow} SS+a-a+ \\ &\underset{rm}{\Rightarrow} Sa+a-a+ \underset{rm}{\Rightarrow} SS-a+a-a+ \underset{rm}{\Rightarrow} aa-a+a-a+ \end{aligned} \tag{6}$$

4. Give a parse tree for the string aa−a+a−a+. [3points]



Exercise 2 (Top-Down Parsing)

Consider the following grammar G: S → aB, B→S∗B|ε

1. Construct the predictive parsing table for G. Please put down the detailed steps, including the calculation of FIRST and FOLLOW sets. [15 points]

1. Compute the FIRST sets of each nonterminal.

   S is a nonterminal, $S \to aB$.

   a is a terminal, then FIRST(a) = {a}.

   Because ϵ is not in FIRST(a), only add all non-ϵ symbols of FIRST(a) to FIRST(S). So FIRST(S) = {a}.

   B is a nonterminal and $B \to \epsilon$, then add ϵ to FIRST(B).

$B \to S * B$, ϵ is not in FIRST(S), only add all non-ϵ symbols of FIRST(S) to FIRST(B).

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

2. Compute the FOLLOW sets of each nonterminal.

   Add \$ in FOLLOW(S).

   $S \to aB$, then everything in FOLLOW(S) is in FOLLOW(B).

   $B \to S * B$, then everything in FIRST($*$) except ϵ is in FOLLOW(S), so $*$ is in FOLLOW(S).

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

3. Construct the parsing table.

   For production $S \to aB$, FIRST(aB) = $\{a\}$. a is a terminal, add $S \to aB$ to $M[S, a]$.

   For production $B \to S * B$, FIRST($S * B$) = $\{a\}$. a is a terminal, add $B \to S * B$ to $M[B, a]$.

   For production $B \to \epsilon$, for each terminal in FOLLOW(B), which is \$ and $*$, add $B \to \epsilon$ to $M[B, \$]$ and $M[B, *]$.

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

2. Is the grammar LL(1)? [3 points]

Yes. The parsing table of an LL(1) parser has no entries with multiple productions.

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. [8 points]**

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 $B \rightarrow \epsilon$
$aaaa*$	$B*B*B\$$	$**\$$	Match $*$
$aaaa*$	$*B*B\$$	$**\$$	Output $B \rightarrow \epsilon$
$aaaa**$	$B*B\$$	$*\$$	Match $*$
$aaaa**$	$*B\$$	$*\$$	Output $B \rightarrow \epsilon$
$aaaa***$	$B\$$	$\$$	Match $*$
$aaaa***$	$\$$	$\$$	Output $B \rightarrow \epsilon$

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. [45 points]

(1) SLR

Augmented grammar:

$$\begin{gathered} S' \rightarrow S(1) \\ S \rightarrow aB(2) \\ B \rightarrow S*B(3) \\ B \rightarrow \epsilon(4) \end{gathered} \tag{7}$$

STATE	CLOSURE
0	$\{[S' \to \cdot S], [S \to \cdot aB]\}$
1	$\{[S' \to S \cdot]\}$
2	$\{[S \to a \cdot B], [B \to \cdot S * B], [B \to \cdot], [S \to \cdot aB]\}$
3	$\{[S \to aB \cdot]\}$
4	$\{[B \to S \cdot * B]\}$
5	$\{[B \to S * \cdot B], [B \to \cdot S * B], [B \to \cdot], [S \to \cdot aB]\}$
6	$\{[B \to S * B \cdot]\}$

STATE	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		

(2) CLR

STATE	CLOSURE	Computation Step
0	$\{[S' \to \cdot S, \$], [S \to \cdot aB, \$]\}$	
1	$\{[S' \to S \cdot, \$]\}$	$GOTO(0, S)$
2	$\{[S \to a \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$], [S \to \cdot aB, *]\}$	$GOTO(0, a) = CLOSURE([S \to a \cdot B, \$])$
3	$\{[S \to aB \cdot, \$]\}$	$GOTO(2, B) = CLOSURE([S \to aB \cdot, \$])$
4	$\{[B \to S \cdot * B, \$]\}$	$GOTO(2, S) = CLOSURE([B \to S \cdot * B, \$])$
5	$\{[S \to a \cdot B, *], [B \to \cdot S * B, \$], [B \to \cdot, \$], [S \to \cdot aB, *]\}$	$GOTO(2, a) = CLOSURE([S \to a \cdot B, *])$
6	$\{[B \to S * \cdot B, \$], [B \to \cdot S * B, \$], [B \to \cdot, \$], [S \to \cdot aB, *]\}$	$GOTO(4, *) = CLOSURE([B \to S * \cdot B, \$])$
7	$\{[S \to aB \cdot, *]\}$	$GOTO(5, B) = CLOSURE([S \to aB \cdot, *])$
8	$\{[B \to S \cdot * B, *]\}$	$GOTO(5, S) = CLOSURE([B \to S \cdot * B, *])$
9	$\{[B \to S * B \cdot, \$]\}$	$GOTO(6, B) = CLOSURE([B \to S * B \cdot, \$])$
10	$\{[B \to S * \cdot B, *], [B \to \cdot S * B, *], [B \to \cdot, *], [S \to \cdot aB, *]\}$	$GOTO(8, *) = CLOSURE([B \to S * \cdot B, *])$
11	$\{[B \to S * B \cdot, *]\}$	$GOTO(10, B) = CLOSURE([B \to S * B \cdot, *])$

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

(3) LALR

STATE	a	*	$	S	B
0	s25			1	
1			acc		
25	s25	r4	r4	48	37
37		r2	r2		
48		s610			
610	s25	r4	r4	48	911
911		r3	r3		

2. Is the grammar SLR(1)? Is the grammar LR(1)? Is the grammar LALR(1)? [9 points]

There are no conflict in SLR(1), LR(1), and LALR(1) parsing table, so it is SLR(1), LR(1), and LALR(1) grammar.

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. [8 points]**

Yes, it can.

STACK	SYMBOL	INPUT	ACTION
$\$0$	$\$$	$aaaa***\$$	s25
$\$0\ 25$	$\$a$	$aaa***\$$	s25
$\$0\ 25\ 25$	$\$aa$	$aa***\$$	s25
$\$0\ 25\ 25\ 25$	$\$aaa$	$a***\$$	s25
$\$0\ 25\ 25\ 25\ 25$	$\$aaaa$	$***\$$	r4
$\$0\ 25\ 25\ 25\ 25\ 37$	$\$aaaaB$	$***\$$	r2
$\$0\ 25\ 25\ 25\ 48$	$\$aaaS$	$***\$$	s610
$\$0\ 25\ 25\ 25\ 48\ 610$	$\$aaaS*$	$**\$$	r4
$\$0\ 25\ 25\ 25\ 48\ 610\ 911$	$\$aaaS*B$	$**\$$	r3
$\$0\ 25\ 25\ 25\ 37$	$\$aaaB$	$**\$$	r2
$\$0\ 25\ 25\ 48$	$\$aaS$	$**\$$	s610
$\$0\ 25\ 25\ 48\ 610$	$\$aaS*$	$*\$$	r4
$\$0\ 25\ 25\ 48\ 610\ 911$	$\$aaS*B$	$*\$$	r3
$\$0\ 25\ 25\ 37$	$\$aaB$	$*\$$	r2
$\$0\ 25\ 48$	$\$aS$	$*\$$	s610
$\$0\ 25\ 48\ 610$	$\$aS*$	$	r4
$\$0\ 25\ 48\ 610\ 911$	$\$aS*B$	$	r3
$\$0\ 25\ 37$	$\$aB$	$	r2
$\$0\ 1$	$\$S$	$	acc

Optional Exercise (15 bonus points)

1. Consider the following context-free grammar:

Phrase → Human | Animal | Phrase Verb Phrase

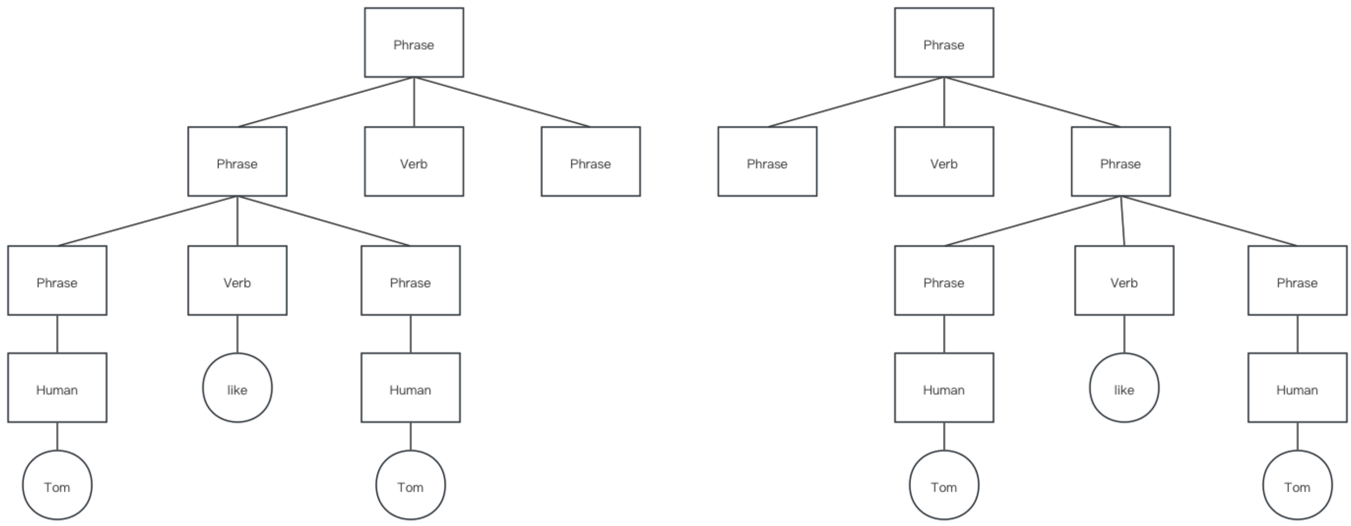
Verb → like | hate

Human → Tom | Jerry | Spike

Animal → cat | mouse | dog

The grammar can produce sentences such as “Tom like dog”. Is the grammar ambiguous? Why? [2 points for the yes/no answer and 8 points for the explanation]

This CFG is ambiguous. Given a grammar, if there are more than one parse tree for some sentence, it is ambiguous. For the same sentence "Tom like Tom like Tom", it has more than one parse tree.



2. For the grammar G in Required Exercise 1, give an equivalent grammar without immediate left recursions. [5 points]

Rewrite $S \to SS+|SS-|a$ to $A \to A\alpha_1|A\alpha_2|\beta$, we have $A = S, \alpha_1 = S+, \alpha_2 = S-, \beta = a$.

So the rewritten G is:

$$\begin{aligned} S &\to aS' \\ S' &\to S+S'|S-S'|\epsilon \end{aligned} \tag{8}$$