编译原理 - 作业(2)：语法分析 LL

截至时间：2021.4.12/周二 上课前（14:20）
提交方式：超算习堂（https://easyhpc.net/course/144）

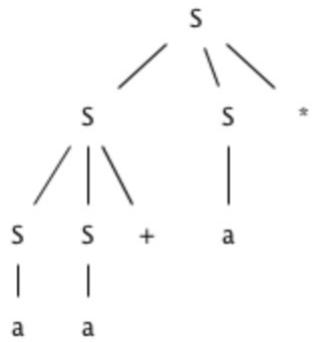
Q1: (p206, Exercise 4.2.1) Consider the context-free grammar:

$$S \rightarrow SS+ \mid SS* \mid a$$

and the string aa + a*.

a) Give the leftmost derivation for the string.
b) Give the rightmost derivation for the string.
c) Give a parse tree for the string.
d) Is the grammar ambiguous or unambiguous? Justify your answer.
e) Describe the language generated by this grammar.

a. S =lm=> SS* => SS+S* => aS+S* => aa+S* => aa+a*
b. S =rm=> SS* => Sa* => SS+a* => Sa+a* => aa+a*
c.



d. Unambiguous
e. The set of all postfix expressions consist of addition and multiplication

Q2: (p216, Exercise 4.3.1) The following is a grammar for regular expressions over symbols a and b only, using + in place of | for union, to avoid conflict with the use of vertical bar as a metasymbol in grammars:

$$rexpr \rightarrow rexpr + rterm \mid rterm$$
$$rterm \rightarrow rterm\ rfactor \mid rfactor$$
$$rfactor \rightarrow rfactor * \mid rprimary$$
$$rprimary \rightarrow \text{a} \mid \text{b}$$

a) Left factor this grammar.
b) Does left factoring make the grammar suitable for top-down parsing?
c) In addition to left factoring, eliminate left recursion from the original grammar.
d) Is the resulting grammar suitable for top-down parsing?

a. 无左公因子

b. 不适合
c. 消除左递归

rexpr -> rterm A
A -> + rterm A | ε
rterm -> rfactor B
B -> rfactor B | ε
rfactor -> rprimary C
C -> * C | ε
rprimary -> a | b

d.适合

Q3: Construct LL(1) parse table of the following grammar. Note: please list the detailed steps.

$E \rightarrow -E$
$E \rightarrow (E) \mid \mathrm{Var}\ T$
$T \rightarrow -E \mid \varepsilon$
$\mathrm{Var} \rightarrow \mathrm{id}\ V$
$V \rightarrow (E) \mid \varepsilon$

First:
FIRST(E)={-, (, id },FIRST(T)={-, ε },FIRST(Var)={id},FIRST(T)={ (, ε}
Follow:
FOLLOW(E)={# ,) },FOLLOW(T) ={# ,) },FOLLOW(Var) ={- , # ,) },FOLLOW(V) ={- , # ,) }

	-	Id	(	)	#
E	E → -E	E → Var T	E→ （E）		
T	T→ -E			T→ε	T→ε
Var		Var → id V			
V	V→ ε		V→ （E）	V→ ε	V→ε

Q4: Check whether the following G[S] grammar is an LL(1) grammar:

$E \rightarrow T\ E'$
$E' \rightarrow A\ T\ E' \mid \varepsilon$
$T \rightarrow F\ T'$
$T' \rightarrow M\ F\ T' \mid \varepsilon$
$F \rightarrow (E) \mid \mathrm{i}$
$A \rightarrow + \mid -$
$M \rightarrow * \mid /$

该文法 parse table 如下：

产生式	FIRST	FOLLOW	SELECT
E→TE′	{(,i}	{),#}	{(,i}
E′→ATE′	{+,−}	{),#}	{+,−}
E′→ε	{ε}		{),#}
T→FT′	{(,i}	{+,−,),#}	{(,i}
T′→MFT′	{*,/}	{+,−,),#}	{*,/}
T′→ε	{ε}		{+,−,),#}
F→(E)	{(}	{+,−,*,/,),#}	{(}
F→i	{i}		{i}
A→+	{+}	{(,i}	{+}
A→−	{−}		{−}
M→*	{*}	{(,i}	{*}
M→/	{/}		{/}

Because：

SELECT(E' -> ATE') ∩ SELECT(E' -> ε)= ∅

SELECT(T' -> MFT') ∩ SELECT(T' -> ε)= ∅

SELECT(F -> (E)) ∩ SELECT(F -> i)= ∅

SELECT(A -> +) ∩ SELECT(A -> -)= ∅

SELECT(M -> *) ∩ SELECT(M -> /)= ∅

Then the SELECT set intersects is empty, the grammar G[S] is an LL(1) grammar.