第5 章 自顶向下语法分析方法

**1**．对文法**G[S]**

**S→a|**∧**|(T)**

**T→T,S|S**

**(1)** 给出**(a,(a,a))**和**(((a,a),**∧**,(a)),a)**的最左推导。

**(2)** 对文法**G**，进行改写，然后对每个非终结符写出不带回溯的递归子程序。

**(3)** 经改写后的文法是否是**LL(1)**的**?**给出它的预测分析表。

**(4)** 给出输入串**(a,a)#**的分析过程，并说明该串是否为**G** 的句子。

答案：(1) 对(a,(a,a)的最左推导为：

S=>（T）=> (T,S)=> (S,S)=> (a,S)=> (a,(T))=> (a,(T,S))=> (a,(S,S))=> (a,(a,S))=> (a,(a,a))

对(((a,a),∧,(a)),a) 的最左推导为：

S => (T)=> (T,S)=> (S,S)=> ((T),S)=> ((T,S),S)=> ((T,S,S),S)=> ((S,S,S),S)=> (((T),S,S),S)

=> (((T,S),S,S),S)=> (((S,S),S,S),S)=> (((a,S),S,S),S)=> (((a,a),S,S),S)=> (((a,a),∧,S),S)

=> (((a,a),∧,(T)),S)=> (((a,a),∧,(S)),S) => (((a,a),∧,(a)),S)=> (((a,a),∧,(a)),a)

(2) 改写文法为：

S→**a|**∧**|(T)**

T→S N

N→,S N| ε

非终结符 FIRST 集 FOLLOW 集

S {a,∧,(} {#,,,)}

T {a,∧,(} {)}

N {,,ε} {)}

对左部为S 的产生式可知：

Select（S→a）∩Select（S→∧）∩Select（S→**(T)**）={a}∩{∧}∩{(}= φ

对左部为N 的产生式可知：

FIRST （,S N）={,}

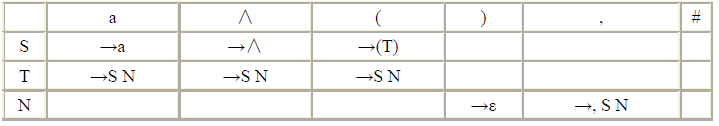
FIRST （ε）={ε}

FOLLOW （N）={)}

由于SELECT(N →, S N)∩SELECT(N →ε) ={,}∩ { )}= φ

所以文法是LL(1)的。

预测分析表:



也可由预测分析表中无多重入口判定文法是LL(1)的。

(3) 对输入串（a,a）#的分析过程为：



可见输入串（a,a）#是文法的句子。

2.对下面的文法G:

E->TE’

E’->+E|ε

T->FT’

T’->T|ε

F->PF’

F’->\*F’|ε

P->(E)|a|b|∧

1. 计算这个文法每个非终结符的FIRST和FOLLOW集。
2. 证明这个文法是LL(1)文法。
3. 构造它的预测分析表。

答案：

（1）非终结符 FIRST 集 FOLLOW 集

E {(,a,b,∧} {#,)}

E’ {+,ε} {#,)}

T {(,a,b,∧} {+,#,)}

T’ {(,a,b,∧, ε} {+,#,)}

F {(,a,b,∧} {(,a,b,∧, +,#,)}

F’ {\*,ε} {(,a,b,∧, +,#,)}

P {(,a,b,∧} {\*,(,a,b,∧, +,#,)}

(2) 要证明这个文法是LL(1)文法，只要证明同一左部的多个产生式的select无交集。

对左部为E’ 的产生式可知：

First(+E)={+} First(ε)={ ε}

Follow(E’)={ #,)}

Select(E’->+E) ∩SELECT(E’ ->ε)={+} ∩{#,)}= φ

对左部为T’ 的产生式可知：

First(T)= {(,a,b,∧} First(ε)={ ε}

Follow(T’)= { +,#,)}

Select(T’->T) ∩SELECT(E’ ->ε)={ (,a,b,∧} ∩{ +,#,)}= φ

对左部为F’ 的产生式可知：

First(\*F’)= {\*} First(ε)={ ε}

Follow(F’)= { (,a,b,∧, +,#,)}

Select(F’->\*F’) ∩SELECT(F’ ->ε)={ \*} ∩{ (,a,b,∧, +,#,)}= φ

所以文法是LL(1)的.

(3)预测分析表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | + | \* | ∧ | ( | ) | # |
| E | E->TE’ | E->TE’ |  |  | E->TE’ | E->TE’ |  |  |
| E' |  |  | E’->+E |  |  |  | E’->ε | E’->ε |
| T | T->FT’ | T->FT’ |  |  | T->FT’ | T->FT’ |  |  |
| T' | T’->T | T’->T | T’->ε |  | T’->T | T’->T | T’->ε | T’->ε |
| F | F->PF’ | F->PF’ |  |  | F->PF’ | F->PF’ |  |  |
| F' | F’->ε | F’->ε | F’->ε | F’->\*F’ | F’->ε | F’->ε | F’->ε | F’->ε |
| P | P->a | P->b |  |  | P->∧ | P->(E) |  |  |

3. 已知文法G[S]：

S→MH|a

H→LSo|ε

K→dML|ε

L→eHf

M→K|bLM

判断G 是否是LL(1)文法，如果是，构造LL(1)分析表。

答案：

非终结符 FIRST 集 FOLLOW 集

S {a,d,b,ε,e} {#,o}

M {d,ε,b} {e,#,o}

H {ε,e} {#,f,o}

L {e} {a,d,b,e,o,#}

K {d,ε} {e,#,o}

对相同左部的产生式可知：

SELECT(S→M H)∩SELECT(S→a) ={ d,b ,e，#,o }∩ { a }=φ

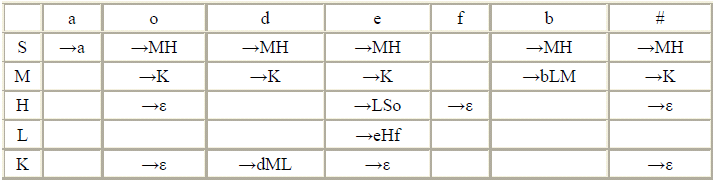
SELECT(H→L S o)∩SELECT(H→ε) ={ e }∩ { #,f,o }=φ

SELECT(K→d M L)∩SELECT(K→ε) ={ d }∩ { e,#,o }=φ

SELECT(M→K)∩SELECT(M→b L M) ={ d，e,#,o }∩ { b }=φ

所以文法是LL(1)的。

预测分析表：



由预测分析表中无多重入口也可判定文法是LL(1)的。

**7** .对于一个文法若消除了左递归，提取了左公共因子后是否一定为**LL(1)**文法**?**试对下面

文法进行改写，并对改写后的文法进行判断。

(2) A→aABe|a

B→Bb|d

(3) S→Aa|b

A→SB

B→ab

答案：

（2）A→aABe|a对提取左公共因子：

A→a X

X→A B e|ε

对B→Bb|d消除左递归后文法变为：

B→d Y

Y→b Y|ε

所以修改后的文法为：

A→a X

X→A B e|ε

B→d Y

Y→b Y|ε

非终结符 FIRST 集 FOLLOW 集

A {a} {#,d}

X {a,ε} {#,d}

B {d} {e}

Y {b,ε} {e}

对相同左部的产生式可知：

SELECT(X→A B e)∩SELECT(X→ε) ={ a }∩ {#,d }=φ

SELECT(Y→b Y)∩SELECT(Y→ε) ={ b }∩ { e }=φ

所以文法是LL(1)的。

（3）消除文法中的间接左递归：

用A 的产生式右部代替S 的产生式右部的A 得：

S→SBa|b

B→ab

消除左递归后文法变为：

S→b N

N→B a N|ε

B→a b

非终结符 FIRST 集 FOLLOW 集

S {b} {#}

B {a} {a}

N {ε,a} {#}

对相同左部的产生式可知：SELECT(N→B a N)∩SELECT(N→ε) ={ a }∩ {# }=φ

所以文法是LL(1)的。