

# 第四章作业 参考答案

- **1、给定文法G,写出G的递归下降分析程序**
- **$E \rightarrow -E$  [1] |  $(E)$  [2] |  $V ET$  [3]**
- **$ET \rightarrow -E$  [4] |  $\varepsilon$  [5]**
- **$V \rightarrow id VT$  [6]**
- **$VT \rightarrow (E)$  [7] |  $\varepsilon$  [8]**

解答：

**1,给每条产生式加一个编号；**

**2,计算每条产生式的Predict集**

**$\text{predict}([1]) = \text{first}(-E) = \{-\}$**

**$\text{predict}([2]) = \text{first}((E)) = \{( \}$**

- $E \rightarrow -E$  [1] |  $(E)$  [2] |  $V ET$  [3]
- $ET \rightarrow -E$  [4] |  $\varepsilon$  [5]
- $V \rightarrow id VT$  [6]
- $VT \rightarrow (E)$  [7] |  $\varepsilon$  [8]

$predict([3]) = first(V ET) = \{id\}$

$predict([4]) = first(- E) = \{-\}$

$predict([5]) = follow(ET) = \{\#, \})\}$

$predict([6]) = first(id VT) = \{id\}$

$predict([7]) = first((E)) = \{( \}$

$predict([8]) = follow(VT) = \{-, \#, \})\}$

**3,检查是否满足递归下降条件, 本题满足**

**4,对每个 $V_N$  写子程序。**

**$V_N = \{E, ET, V, VT\}$**

**•  $E \rightarrow -E$  [1] |  $(E)$  [2] |  $V ET$  [3]**

**predict([1]) = first(-E) = {-}**

**predict([2]) = first((E)) = {(}**

**predict([3]) = first(V ET) = {id}**

**E()**

**{ switch token of**

**“-” : match(-); E(); break;**

**“(” : match(()); E(); match());break;**

**“id” : V(); ET();break;**

**default: error();break; }**

- **$ET \rightarrow -E \text{ [4]} \mid \varepsilon \text{ [5]}$**   
**predict([4]) = first(- E) = {-}**  
**predict([5]) = follow(ET) = {#, )}**

**ET()**

```
{    swich token of  
    "-" : match(-); E(); break;  
    "") :  
    "#" : break;  
default: error();break; }
```

**$V \rightarrow id VT [6]$**

**$predict([6]) = first(id VT) = \{id\}$**

**$V()$**

**{    swich token of  
    “id” : match(id); VT(); break;  
default: error();break; }**

**$VT \rightarrow (E) [7] \mid \varepsilon [8]$**

**predict([7]) = first((E)) = {(**

**predict([8]) = follow(VT) = {-, #, )}**

**VT()**

**{**

**switch token of**

**“(” : match(); E();match());**

**break;**

**“-” :**

**“#” :**

**“)” :break;**

**default: error();break; }**

```
match (tokenType tk)  
{  
    if currentToken == tk  
    then readToken(currentToken)  
    else error();  
}
```

```
main()  
{  
    readToken(tk);  
    E();  
}
```

# LL(1)分析表

	(	)	-	id	#
E	2		1	3	
ET		5	4		5
V				6	
VT	7	8	8		8



# 作业中存在的问题

- 1.空产生式的问题：
  - (1)预测符不全或错误
  - (2)分支动作不对!

```
VT() { switch (token)
{ case (c) = match (c); E();
  match (); break;
  case (-) = skip; break;
  case (,) = skip; break;
  default = error ();
}
}
```

```
ET () { switch (token)
{ case (-) : match (-); E
case (,) * skip; break.
default = error ();
}
}
```

```
ET () { switch (token)
{ case - : match (-); E (); break
case  $\epsilon$  : match (-); E (); break;
default : error ();
}
}
```

# 作业中存在的问题

- 2.LL1分析表中缺少‘#’列或表项定义不全

PAGE:      DATE:

```
main() { read(); E(); }  
match (char w) { if (w == taken) error(); else read(); }
```

②]

	-	(	)	id	#
E	(1)	(2)		(3)	
ET	(4)		(5)		(5)
V				(6)	
VT	(8)	(7)	(8)		(8)

②]

	-	(	)	id	#
E	①	②		③	
ET	④		⑤		⑤
V			⑥	⑥	
VT	⑧	⑦	⑧		⑧