

## 《编译原理》课程实验报告

<b>实验名称</b>  <b>实验内容</b>	实验四 语法分析程序的构造二  对于给定文法 $G[S]$ :  $S \rightarrow aAcBe$ $A \rightarrow b   Ab$ $B \rightarrow d$ 1.用移进--规约给出对输入串 <code>abbcd#</code> 的分析过程。
<b>一、实验目的：</b>  进一步掌握自底向上的语法分析方法的同时，锻炼设计、实现、分析和维护编译程序等方面的初步能力，	
<b>二、主要数据结构：</b>  线性表和散列表	
<b>三、主要设计思想与算法：</b>  首先添加产生式 $Z \rightarrow \#S\#$ ( $S$ 为开始符号)，计算 FIRSTVT、LASTVT 集，然后生成优先分析表，最后根据优先分析表进行算符优先分析。  其中，求 FIRSTVT 集定义如下：  $FIRSTVT(P) = \{a \mid P \xrightarrow{+} a \dots \text{或} P \xrightarrow{+} Qa \dots, a \in V_T; P, Q \in V_N\}$ 求 LASTVT 集定义如下：  $LASTVT(P) = \{a \mid P \xrightarrow{+} \dots a \text{或} P \xrightarrow{+} \dots Qa, a \in V_T; P, Q \in V_N\}$ 算符优先表的构造算符如下：  FOR 每条产生式 $P \rightarrow X_1X_2 \dots X_n$ DO {FOR ( $i=1, i \leq n-1, i++$ ) {if $X_i$ 和 $X_{i+1}$ 均为终结符 then 置 $X_i = X_{i+1}$ If $i \leq n-2$ and $X_i$ 和 $X_{i+2}$ 均为终结符 and $X_{i+1}$ 为非终结符 then 置 $X_i = X_{i+1}$ If $X_i$ 为终结符而 $X_{i+1}$ 为非终结符 then For FIRSTVT( $X_{i+1}$ ) 中的每个 $a$ DO {置 $X_i < a$ } If $X_i$ 为非终结符而 $X_{i+1}$ 为终结符 then           }	

For LASTVT( $X_i$ ) 中的每个 a DO

{置  $a > X_i$  }

#### 四、实验结果及测试用例：

##### 测试用例：

文法：

$S \rightarrow aAcBe$

$A \rightarrow b \mid Ab$

$B \rightarrow d$

语句：abcde

##### 实验结果：

$S \rightarrow aAcBe$

$A \rightarrow b \mid Ab$

$B \rightarrow d$

优先矩阵：

VT	a	c	e	b	d	#
a		=		<		
c			=	<		
e						>
b				>		
d					>	
#	<					=

请输入要语法分析的句子:abcde

步骤	分析栈	关系	符号串	下步动作
1	#	<	abcde#	移进
2	#a	<	bcde#	移进
3	#ab	>	bcde#	归约 ( $A \rightarrow b$ )
3	#aA	<	bcde#	移进
4	#aAb	>	cde#	归约 ( $A \rightarrow Ab$ )
4	#aA	=	cde#	移进
5	#aAc	<	de#	移进
6	#aAcd	>	e#	归约 ( $B \rightarrow d$ )
6	#aAcB	=	e#	移进
7	#aAcBe	>	#	归约 ( $S \rightarrow aAcBe$ )
7	#S		#	成功

分析成功

#### 五、实验总结：

通过本实验，我进一步掌握了自底向上的语法分析方法，通过编写求解 FIRSTVT、LASTVT 集的程序使我对算符优先分析有了更深入的了解，在解决问题的同时提高了设计代码实现算法的能力。

实验初期由于没有检查文法，编写的是简单优先分析程序，而此文法并非简单优先文法，导致重头再来，但自己也加深了对简单优先分析的理解。实验的不足之

处在于分析过程中出错处理并不细致。