

# 自上而下预测分析

为以下文法编写自上而下预测分析器（LL 分析器）

<程序>::=<程序首部><分程序>.  
<程序首部>::=PROGRAM 标识符;  
<分程序>::=<常量说明部分><变量说明部分><过程说明部分><复合语句>  
<常量说明部分>::=CONST <常量定义><常量定义后缀>; |  $\epsilon$   
<常量定义>::=标识符=无符号整数  
<常量定义后缀>::=, <常量定义><常量定义后缀>|  $\epsilon$   
<变量说明部分>::=VAR <变量定义><变量定义后缀>|  $\epsilon$   
<变量定义>::=标识符<标识符后缀>: <类型>;  
<标识符后缀>::=, 标识符<标识符后缀>|  $\epsilon$   
<变量定义后缀>::=<变量定义><变量定义后缀>|  $\epsilon$   
<类型>::=INTEGER|REAL  
<过程说明部分>::=<过程首部><分程序>;<过程说明部分后缀>|  $\epsilon$   
<过程首部>::=PROCEDURE 标识符<参数部分>;  
<参数部分>::=(标识符: <类型>)  
<过程说明部分后缀>::=<过程首部><分程序>;<过程说明部分后缀>|  $\epsilon$   
<语句>::=<赋值或调用语句>|<条件语句>|<当型循环语句>|<读语句>  
|<写语句>|<复合语句>|  $\epsilon$   
<赋值或调用语句>::=标识符<后缀>  
<后缀>::=<表达式>|(<表达式>)|  $\epsilon$   
<条件语句>::=IF <条件> THEN <语句>  
<当型循环语句>::=WHILE <条件> DO <语句>  
<读语句>::=READ (标识符<标识符后缀>)  
<写语句>::=WRITE (<表达式><表达式后缀>)  
<表达式后缀>::=, <表达式><表达式后缀>|  $\epsilon$   
<复合语句>::=BEGIN <语句><语句后缀> END  
<语句后缀>::=;<语句><语句后缀>|  $\epsilon$   
<条件>::=<表达式><关系运算符><表达式>|ODD <表达式>  
<表达式>::=+<项><项后缀>|-<项><项后缀>|<项><项后缀>  
<项后缀>::=<加型运算符><项><项后缀>|  $\epsilon$   
<项>::=<因子><因子后缀>  
<因子后缀>::=<乘型运算符><因子><因子后缀>|  $\epsilon$   
<因子>::=标识符|无符号整数|(<表达式>)  
<整数后缀>::=.无符号整数|  $\epsilon$   
<加型运算符>::=+|-  
<乘型运算符>::=\*|/  
<关系运算符>::>=|<|<=>|>=

# Step 1：编写词法分析器

为词法单元建立内码：

| 内码 | 单词      | 内码 | 单词        | 内码 | 单词   | 内码 | 单词      |
|----|---------|----|-----------|----|------|----|---------|
| 1  | PROGRAM | 2  | CONST     | 3  | VAR  | 4  | INTEGER |
| 5  | LONG    | 6  | PROCEDURE | 7  | IF   | 8  | THEN    |
| 9  | WHILE   | 10 | DO        | 11 | READ | 12 | WRITE   |
| 13 | BEGIN   | 14 | END       | 15 | ODD  | 16 | +       |
| 17 | -       | 18 | *         | 19 | /    | 20 | =       |
| 21 | <>      | 22 | <         | 23 | <=   | 24 | >       |
| 25 | >=      | 26 | .         | 27 | ,    | 28 | ;       |
| 29 | :       | 30 | :=        | 31 | (    | 32 | )       |
| 33 | 无符号整数   | 34 | 标识符       | 35 | #    |    |         |

词法分析器代码详见附录 A

# Step 2：编写 LL 分析器

为非终结符建立内码：

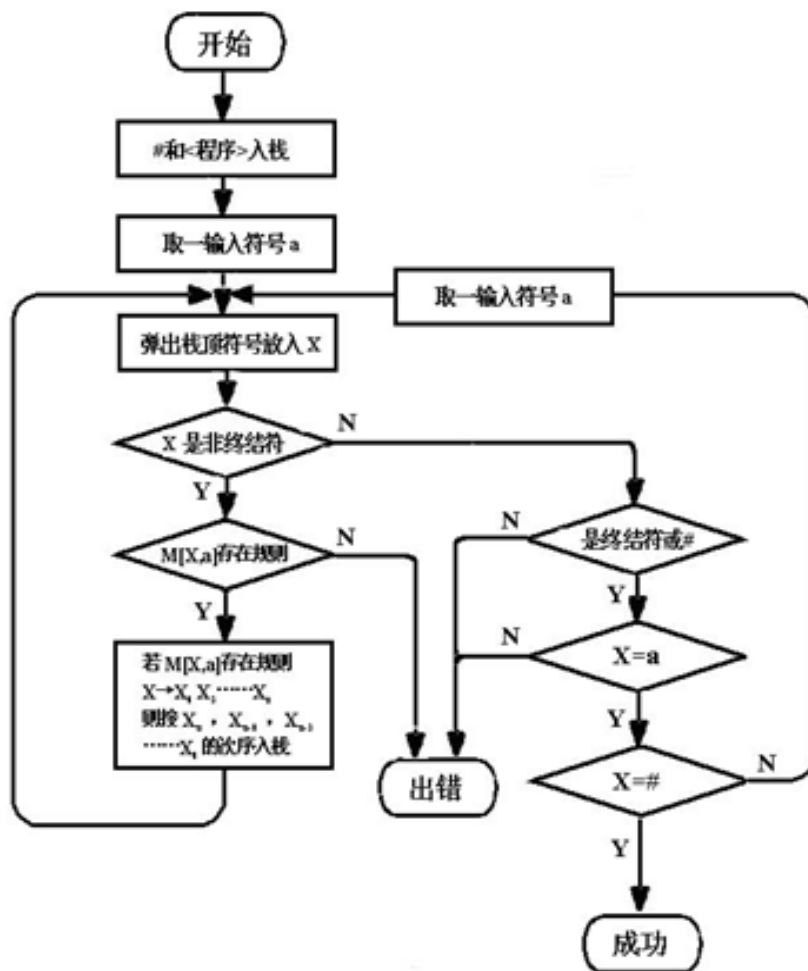
| 内码  | 非终结符       | 内码  | 非终结符     | 内码  | 非终结符      | 内码  | 非终结符     |
|-----|------------|-----|----------|-----|-----------|-----|----------|
| 128 | <程序>       | 129 | <程序首部>   | 130 | <分程序>     | 131 | <常量说明部分> |
| 132 | <常量定义>:    | 133 | <常量定义后缀> | 134 | <变量说明部分>  | 135 | <变量定义>   |
| 136 | <变量定义后缀>   | 137 | <类型>     | 138 | <过程说明部分>  | 139 | <过程首部>   |
| 140 | <过程说明部分后缀> | 141 | <语句>     | 142 | <赋值或调用语句> | 143 | <后缀>     |
| 144 | <条件语句>     | 145 | <当型循环语句> | 146 | <读语句>     | 147 | <标识符后缀>  |
| 148 | <写语句>      | 149 | <表达式后缀>  | 150 | <复合语句>    | 151 | <语句后缀>   |
| 152 | <条件>       | 153 | <表达式>    | 154 | <项后缀>     | 155 | <项>      |
| 156 | <因子后缀>     | 157 | <因子>     | 158 | <参数部分>    | 159 | <加型运算符>  |
| 160 | <乘型运算符>    | 161 | <关系运算符>  |     |           |     |          |

产生式规则的存储：

| 规则右部符号串编号                          | 内码表示               |
|------------------------------------|--------------------|
| (1) <程序首部><分程序>.                   | 129 130 26 0       |
| (2) PROGRAM 标识符;                   | 1 34 28 0          |
| (3) <常量说明部分><变量说明部分><过程说明部分><复合语句> | 131 134 138        |
| 150 0                              |                    |
| (4) CONST <常量定义><常量定义后缀>;          | 2 132 133 28 0     |
| (5) $\varepsilon$                  | 0                  |
| (6) 标识符=无符号整数                      | 34 20 33 0         |
| (7) , <常量定义><常量定义后缀>               | 27 132 133 0       |
| (8) VAR <变量定义><变量定义后缀>             | 3 135 136 0        |
| (9) 标识符<标识符后缀>: <类型>;              | 34 147 29 137      |
| 28 0                               |                    |
| (10) , 标识符<标识符后缀>                  | 27 34 147 0        |
| (11) <变量定义><变量定义后缀>                | 135 136 0          |
| (12) INTEGER                       | 4 0                |
| (13) LONG                          | 5 0                |
| (14) <过程首部><分程序>;<过程说明部分后缀>        | 139 130 28 140 0   |
| (15) PROCEDURE 标识符 <参数部分>;         | 6 34 158 28 0      |
| (16) <赋值或调用语句>                     | 142 0              |
| (17) <条件语句>                        | 144 0              |
| (18) <当型循环语句>                      | 145 0              |
| (19) <读语句>                         | 146 0              |
| (20) <写语句>                         | 148 0              |
| (21) <复合语句>                        | 150 0              |
| (22) 标识符<后缀>                       | 34 143 0           |
| (23) :=<表达式>                       | 30 153 10          |
| (24) IF <条件> THEN <语句>             | 7 152 8 141 0      |
| (25) WHILE <条件> DO <语句>            | 9 152 10 141 0     |
| (26) READ (标识符<标识符后缀>)             | 11 31 34 147 32 0  |
| (27) WRITE (<表达式><表达式后缀>)          | 12 31 153 149 32 0 |
| (28) , <表达式><表达式后缀>                | 27 153 149 0       |
| (29) BEGIN <语句><语句后缀> END          | 13 141 151 14 0    |
| (30) ;<语句><语句后缀>                   | 28 141 151 0       |
| (31) <表达式><关系运算符><表达式>             | 153 161 153 0      |
| (32) ODD <表达式>                     | 15 153 0           |
| (33) +<项><项后缀>                     | 16 155 154 0       |
| (34) -<项><项后缀>                     | 17 155 154 0       |
| (35) <项><项后缀>                      | 155 154 0          |
| (36) <加型运算符><项><项后缀>               | 159 155 154 0      |
| (37) <因子><因子后缀>                    | 157 156 0          |
| (38) <乘型运算符><因子><因子后缀>             | 160 157 156 0      |
| (39) 标识符                           | 34 0               |
| (40) 无符号整数                         | 33 0               |
| (41) (<表达式>)                       | 31 153 32 0        |

|      |            |                 |
|------|------------|-----------------|
| (42) | .无符号整数     | 26 33 0         |
| (43) | +          | 16 0            |
| (44) | -          | 17 0            |
| (45) | *          | 18 0            |
| (46) | /          | 19 0            |
| (47) | =          | 20 0            |
| (48) | <>         | 21 0            |
| (49) | <          | 22 0            |
| (50) | <=         | 23 0            |
| (51) | >          | 24 0            |
| (52) | >=         | 25 0            |
| (53) | (标识符 <类型>) | 31 34 29 137 32 |
|      | 0          |                 |

算法流程：



分析分析决策表如下程序：

```
int rule[][6]={0},{129,130,26,0},{1,34,28,0},{131,134,138,150,0},
{2,132,133,28,0},{0},{34,20,33,0},{27,132,133,0},{3,135,136,0},
```

```

{34,147,29,137,28,0},{27,34,147,0},{135,136,0},{4,0},{5,0},
{139,130,28,140,0},{6,34,158,28,0},{142,0},{144,0},{145,0},{146,0},
{148,0},{150,0},{34,143,0},{30,153,0},{7,152,8,141,0},{9,152,10,141,0},
{11,31,34,147,32,0},{12,31,153,149,32,0},{27,153,149,0},
{13,141,151,14,0},{28,141,151,0},{153,161,153,0},{15,153,0},
{16,155,154,0},{17,155,154,0},{155,154,0},{159,155,154,0},
{157,156,0},{160,157,156,0},{34,0},{33,0},{31,153,32,0},
{26,33,0},{16,0},{17,0},{18,0},{19,0},{20,0},{21,0},{22,0},
{23,0},{24,0},{25,0},{31,34,29,137,32,0}};
int table[34][35];
int pretable[][3]={128,1,1},{129,1,2},{130,2,3},{130,3,3},
{130,6,3},{130,34,3},{130,7,3},{130,9,3},{130,11,3},
{130,12,3},{130,13,3},{130,26,3},{130,28,3},{131,2,4},
{131,3,5},{131,6,5},{131,34,5},{131,7,5},{131,9,5},{131,11,5},
{131,12,5},{131,13,5},{131,26,5},{131,28,5},{132,34,6},
{133,27,7},{133,28,5},{134,3,8},{134,6,5},{134,34,5},{134,7,5},
{134,9,5},{134,11,5},{134,12,5},{134,13,5},{134,26,5},
{134,28,5},{135,34,9},{136,6,5},{136,13,5},{136,34,11},{136,28,5},{137,4,12},
{137,5,13},{138,6,14},{138,34,5},{138,7,5},{138,9,5},
{138,11,5},{138,12,5},{138,13,5},{138,26,5},{138,28,5},
{139,6,15},{140,6,14},{140,34,5},{140,7,5},{140,9,5},
{140,11,5},{140,12,5},{140,13,5},{140,26,5},{140,28,5},
{141,34,16},{141,7,17},{141,9,18},{141,11,19},{141,12,20},
{141,13,21},{141,14,5},{141,26,5},{141,28,5},{142,34,22},
{143,30,23},{143,31,41},{143,28,5},
{143,28,5},{143,14,5},{143,26,5},{143,14,5},
{144,7,24},{145,9,25},{146,11,26},{147,27,10},
{147,29,5},{147,32,5},{148,12,27},{149,27,28},{149,32,5},
{150,13,29},{150,26,5},{150,28,5},{151,28,30},{151,14,5},
{151,26,5},{151,28,30},{152,34,31},{152,16,31},{152,17,31},
{152,34,31},{152,33,31},{152,31,31},{152,15,32},{153,34,35},
{153,16,33},{153,17,34},{153,34,35},{153,33,35},{153,31,35},
{154,16,36},{154,17,36},{154,8,5},{154,10,5},{154,14,5},
{154,20,5},{154,21,5},{154,22,5},{154,23,5},{154,24,5},
{154,25,5},{154,26,5},{154,27,5},{154,28,5},{154,32,5},
{155,34,37},{155,33,37},{155,31,37},{156,18,38},{156,19,38},
{156,8,5},{156,10,5},{156,14,5},{156,16,5},{156,17,5},
{156,20,5},{156,21,5},
{156,22,5},{156,23,5},{156,24,5},{156,25,5},{156,26,5},
{156,27,5},{156,28,5},{156,32,5},{157,34,39},{157,33,40},
{157,31,41},{158,31,53},{158,28,5},
{159,16,43},{159,17,44},{160,18,45},
{160,19,46},{161,20,47},{161,21,48},{161,22,49},{161,23,50},
{161,24,51},{161,25,52},{0,0,0}};

```

```

/*预测分析决策表 table*/
while(pretable[i][0])
    {table[pretable[i][0]-128][pretable[i][1]]=pretable[i][2];
    i++;};

```

## 附录 A 词法分析程序源程序和注释

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <process.h> /*VC++使用该文件*/
#define lenth1 15
#define lenth2 17
struct {char name[21];int type;int addr;}
    indent[1000]; /*为标识符表*/
struct st
{char name[21];int code;} sym;
int lenth=0; /*为标识符表长*/
FILE *f1,*f2; /* f1 为输入文件指针,f2 输出文件指针*/
int line=0,row=0,val;
void getsym();
char getch();
void error(int);
main(int argc,char *argv[])
    { /*argc 表示参数的个数、argv[0]、argv[1]、argv[2]……
        表示运行程序时输入的字符串，包括执行文件名，调试时其参数设置，VC++6.0
        在菜单 Project 下的子菜单 Settings 中的 Debug 选项卡中 Program arguments 输入对话框
        中设置 Turbo C 2.0 在 Options 菜单下的子菜单 Options 中输入对话框中设置 */
        char ft[12],*fc;
        if((f1=fopen(argv[1],"r"))==NULL)
            {printf("cannot open the file\n");
            exit(0);
            /*如不能打开输入文件，显示出错信息退出*/
        }
        /*当参数个数小于等于 2 时，用户没有指出输出文件此时生成一个与输入
        文件名相同而后缀名不同的输出文件名；否则用户已输入输出文件名故直接使用*/
        if(argc<=2)
            {strcpy(ft,argv[1]);
            if((fc=strchr(ft,'.'))!=NULL)
                strcpy(fc,".mid");
            else
                strcat(ft,".mid");
            }
        else

```

```

strcpy(ft,argv[2]);
if((f2=fopen(ft,"w"))==NULL) /*打开输出文件*/
{printf("cannot open the file\n");
exit(0);
}
while(!feof(f1))
{getsym();
printf("%s %d\n",sym.name,sym.code);
fprintf(f2,"%s %d\n",sym.name,sym.code);
}
fclose(f1); fclose(f2);
}

void getsym()
{static char a[lenth1][10]={"program","const","var","integer","long","procedure","if",
"then","while","do","read","write","begin","end","odd"
},
d[lenth2][3]={"+","-","*","/","=","<","<=",">",">=",".",",",";",":",":=","(",")"},
str[21],ch=' ';
int i,n;
while(isspace(ch))
ch=getchr();
if(isalpha(ch))
{n=0;
while(isalpha(ch) || isalnum(ch))
{if(isalpha(ch))
ch=tolower(ch);
str[n++]=ch;
ch=getchr();
}
str[n]='\0';
for(i=0;i<lenth1;i++)
if(!strcmp(str,a[i])) break;
if(i<lenth1)
{strcpy(sym.name,a[i]);
sym.code=i+1;
}
else
{for(i=0;i<lenth;i++)
if(!strcmp(str,indent[i].name)) break;
if(i==lenth)
strcpy(indent[i].name,str);
strcpy(sym.name,indent[i].name);
sym.code=34;
}
}
}

```

```

    }
else if(isalnum(ch))
{
    val=0;n=0;
    while(isalnum(ch))
    {val=val*10+ch-'0';
    sym.name[n++]=ch;
    ch=getchr();
    }
    sym.name[n]='\0';
    sym.code=33;
}
else
{
    if(ch=='+' || ch=='-' || ch=='*' || ch=='/' || ch=='=' || ch=='.' || ch==','
        || ch==';' || ch=='(' || ch==')')
    {str[0]=ch;str[1]='\0';
    ch=getchr();
    for(i=0;i<lenth2;i++)
        if(!strcmp(str,d[i]))
        {strcpy(sym.name,str);
        sym.code=i+16;
        }
    }
else {n=0;
    if (ch=='>' || ch==':')
    {str[n++]=ch;
    if((ch=getchr())=='=')
    {str[n++]=ch;ch=getchr();
    }
    }
    else
    if(ch=='<')
    {str[n++]=ch;ch=getchr();
    if(ch=='=' || ch=='>')
    {str[n++]=ch;ch=getchr();
    }
    }
    else if(ch==-1)
    {strcpy(sym.name,"");
    sym.code=35;
    }
    else
        error(1);
}
str[n]='\0';

```



```

        for(i=0;i<lenth2;i++)
        if(!strcmp(str,d[i]))
        {strcpy(sym.name,str);
        sym.code=i+16;
        }
    }
}

char getch()
{char ch=fgetc(f1);
if(ch=='\n')
{row=1;
line++;
}
else
if(ch!=' '&&ch!='\t')
row++;
return(ch);
}

void error(int n)
{printf("There are %d-error\n");
exit(0);
}

```