2025/12/06 07:02 1/4 编译简介

# 递归下降与优先爬升

#### 文法规则

用"::="符号,表示"由……组成"。文法规则一般形如:

<句子> ::= <主语> <谓语>

<谓语>::= <动词> <宾语>

表示"句子由主语和谓语构成","谓语由动词和宾语构成"。这种构成是按顺序的。

文法规则中有一些特殊符号。

符号ε□表示空串,即长度为0的字符串,就是什么都没有。空串的概念类似于空集,但是空串是串,仍旧作为元素来看待,与空集概念有区别。

符号|:表示"或"。例如:

A ::= B | C

表示A由B构成,或者由C构成。这里的B和C称为文法项A的候选式。

符号{}:大括号里的内容,表示可以重复0次到重复任意有限次。不能重复无限次,因为字符串永远有限长。例如:

 $A ::= \{a\}$ 

表示A可以匹配ɛ[]a[]aaa[]aaa[]......[所有只由a构成的字符串。

文法G[S]的含义是,文法G的"入口点"(起始符号)是S□相当于语法树的根节点。

### 语法分析

语法分析是编译的一个环节,可以检查输入的字符串是否符合文法规则。

语法分析的思维模式总共分为两种:自顶向下的分析、自底向上的分析,它们按照遍历语法树的顺序来定义。

递归下降属于自顶向下的分析,优先爬升属于自底向上的分析。

### 递归下降

递归下降是一种语法分析的设计方法。

能够递归下降的文法,需要满足3个条件:

1.没有左递归。例如文法

 $A ::= a \mid Ab$ 

无法使用递归下降,必须改写为与之等价的

```
A ::= a\{b\}
```

才能使用递归下降。

2.候选式首符号不相交。例如文法:

```
A ::= ab \mid ac
```

无法使用递归下降。必须改写为与之等价的

A ::= aB

B := b l c

才能使用递归下降。

3.如果候选式可以为ε□则候选式的首符号与该语法项的后继符号也不相交。例如文法:

S ::= Aa

 $A ::= \varepsilon \mid ab$ 

无法使用递归下降。必须改写为与之等价的

S ::= aB

 $B ::= \varepsilon \mid ab$ 

才能使用递归下降。

递归下降的设计方法是:

对每一个左部的语法项,设计一个函数。

进入函数时,根据读入的首符号,来确定进入哪个候选式分支。

如果遇到ε□先预读一个符号,如果判断为语法项的后继符号,则进入该分支,于是退回一个符号并返回。

如果遇到大括号,通过while循环来实现循环0次与无数次的目的。在while的判断处要读入字符检查是否进入循环,因此在while结束后要退回一个字符。

示例:括号匹配的文法G[S]为

S ::= A

 $A ::= \varepsilon \mid '(' A ')' A \mid '[' A ']' A \mid '\{' A '\}' A$ 

于是设计的程序为:

```
int S() {
   int ans = A(); //括号匹配
   if (ans == 0) {
      return 0;
   }
   char c = getchar(); //匹配完应该读完
```

https://wiki.cvbbacm.com/ Printed on 2025/12/06 07:02

2025/12/06 07:02 3/4 编译简介

```
if (c != EOF) {
       return 0;
    }
    return 1;
}
int A() {
    char c = getchar();
    if (c == ')' || c == ']' || c == '}' || c == EOF) {
        ungetc(c, stdin);//与getchar相反,向读入中退回一个字符
        return 1;
    } else if (c == '(') {
        int temp = A();
        if (temp == 0)//调用匹配失败
            return 0;
        c = getchar();
        if (c != ')') {
           return 0;
        temp = A();
        if (temp == 0) {
            return 0;
        return 1;
    } else if (c == '[') {
        int temp = A();
        if (temp == 0) {
            return 0;
        c = getchar();
        if (c != ']') {
           return 0;
        temp = A();
        if (temp == 0) {
           return 0;
        return 1;
    } else if (c == '{') {
        int temp = A();
        if (temp == 0) {
            return 0;
        c = getchar();
        if (c != '}') {
            return 0;
        temp = A();
        if (temp == 0) {
```

Last update: 2020-2021:teams:namespace: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E9%80%92%E5%BD%92%E4%B8%8B%E9%99%8D%E4%B8%8E%E4%BC%98%E5%85%88%E7%88%AC%E5%8D%87&rev=1617863633 24:33

```
return 0;
}
return 1;
```

## 优先爬升

From: https://wiki.cvbbacm.com/ - CVBB ACM Team





Printed on 2025/12/06 07:02 https://wiki.cvbbacm.com/