自顶向下的分析算法通过在最左推导中描述出各个步骤来分析记号串输入。

之所以叫自顶向下，是因为分析树隐含的编号是一个前序编号，而且其顺序是由根到叶。

自顶向下的分析程序有两类：回溯分析程序和预测分析程序。

本章要学习的两类自顶向下分析算法是递归下降分析和LL(1)分析。

1、LL(1)的含义

LL(1)，第一个L指从左向右地处理输入，第二个L指为输入串生成一个最左推导的分析树，括号中的1指它仅适用输入中的1个符号来预测分析的方向。

递归下降分析程序和LL(1)分析一般地都要求计算先行集合，它们分别称作：First集合和Follow集合。

2、First集合

含义

可以正规地开始这个串的token。

如$First(A) = \{a, (\}$，则串A可以由a和(开始。

构造方法

若A是终结符，则$First(A) = \{A\}$。

设有产生式$A \to \alpha$，其中A是非终结符，$\alpha = X_1 X_2 \dots X_n$是终结符和非终结符的串，，$First(A) = First(\alpha)$。

1. $First(\alpha)$包括$First(X_1) - \{\epsilon\}$；
2. 对于每个$X_i, i = 2, \dots, n$，如果所有的$First(X_k), k = 1, \dots, i-1$都包括$\epsilon$，，则$First(\alpha)$包括$First(X_i) - \{\epsilon\}$；
3. 如果所有的$First(X_i), i = 1, \dots, n$都包括$\epsilon$，则$First(\alpha)$包括$\epsilon$。

需要将文法规则消除左递归和提取公因子，然后将选择展开，生成新的文法。

按照一定顺序，对于每一个产生式，运行上面的三个步骤，直至First集合不再发生变化，First集合即构造成功。

3、Follow集合

含义

可以正规地出现在非终结符A之后的token的集合，这样的token指出A可以恰当地在南非西中的这个点处消失。

构造方法

设A为非终结符，那么$Follow(A)$由终结符组成，此外可能还有$。

1. $Follow(S)$包括$，其中S为开始符号；
2. 若存在产生式$B \to \alpha A \gamma$，则$Follow(A)$包括$First(\gamma) - \{\epsilon\}$；
3. 若存在产生式$B \to \alpha A \gamma$且$First(\gamma)$包括ϵ，则$Follow(A)$包括$Follow(B)$。

递归下降分析方法

将文法规则A转化为识别A的一个函数（此外还有`match()`、`getToken()`）选择与选择分支语句对应。

如果使用的是递归下降程序，就应总是将文法翻译成EBNF（实际上，EBNF标识符是为了更紧密地映射递归下降分析程序的真实代码而设计的）。

问题

1. 将原来的BNF文法转变成EBNF文法可能有困难。
2. 在用公式表达一个用以区分两个或更多的文法规则选项的测试$A \to \alpha|\beta|\ldots$时，如果$\alpha$和$\beta$均以非终结符开始，那么就很难决定何时使用$A \to \alpha$，何时使用$A \to \beta$。

   这个问题要求计算α和β的First集合。
3. 在写ϵ产生式的代码$A \to \epsilon$时，需要了解什么token可以正规地出现在非终结符A之后，这是因为这样的token指出A可以恰当地在分析中的这个点处消失，这个集合被称为A的Follow集合。

LL(1)分析

- 不用将BNF转换为EBNF，但仍需处理文法规则
  - LL(1)分析中的重复和可选也存在着与在递归下降程序分析中遇到的类似问题
  - 我们利用EBNF表示法解决了递归下降程序中的这些问题
  - 但却不能在LL(1)分析中使用相同的方法，而是使用左递归消除和提取左因子
  - 左递归消除和提取左因子不能保证将一个文法变成LL(1)文法，就像EBNF不能保证在编写递归下降程序中可以解决所有的问题
- 使用显示栈而不是递归调用来完成分析

过程

需要画4列

步骤	分析栈	输入	动作
1	$S	...$	...
2	...	...	...
...	...	...	...
n	$	$	accept

1. 自顶向下的分析程序是从将开始符号放在栈中开始的；
2. 一系列步骤
   - 生成 generate

     选择$A \to \alpha$将栈顶的非终结符A替换为α（注意：α要倒序进栈）
   - 匹配 match

     将栈顶的token与输入中的下一个token匹配（要将匹配成功的两个token从栈中和输入中删除）
3. 栈和输入都空，即栈和输入中都只有$，此时接受（accept）

4、LL(1)分析表

定义

一个二维数组，索引为 非终结符 与 终结符和 $，这个表称为$M[N,T]$。

作用

为替换栈顶的A选择产生式

构造方法

假设分析表在开始时，所有项目为空。

根据以下规则在这个表中添加产生式：

对于每个产生式$A \to \alpha$，其中A是非终结符，重复以下两个步骤：

1. 对于$First(\alpha)$中的每个终结符a，都将$A \to \alpha$添加至$M[A,a]$中；
2. 若$First(\alpha)$包括ϵ，则对于$Follow(A)$中的每个元素a（记号或者是 $），都将$A \to \alpha$添加至$M[A,a]$。

5、LL(1)文法

定义

如果文法G相关的LL(1)分析表中的每个项目中至多有一个产生式，则该文法就是LL(1)文法。

判断定理

若满足以下两个条件，则BNF中的文法就是LL(1)文法。

1. 在每个产生式$A \to \alpha_1|\alpha_2|\dots|\alpha_n$，$First(\alpha_i) \cap First(\alpha_j) = \emptyset, 1 \le i,j \le n, i \ne j$；
2. 若非终结符A都有$First(A)$包含ϵ，那么$First(A) \cap Follow(A) = \emptyset$。

7、消除左递归

普遍的直接左递归

这种情况发生在有如下格式的产生式

$$A \to A\alpha_1|A\alpha_2|\dots|A\alpha_n|\beta_1|\beta_2|\dots|\beta_m \quad (1)$$

中，其中$\beta_1,\dots,\beta_m$均不以A开头。在这种情况下，其解法与简单情况类似，只需将选择相应地扩展：

$$\begin{aligned} &① \ A \to \beta_1 A'|\beta_2 A'|\dots|\beta_m A' \\ &② \ A' \to \alpha_1 A'|\alpha_2 A'|\dots|\alpha_n A'|\epsilon \end{aligned} \quad (2)$$

一般的左递归

8、提取左因子

当两个或更多个产生式选择共享一个通用前缀串时，需要提取左因子。

如$A \to \alpha\beta|\alpha\gamma$可重写为：

$$\begin{aligned} A &\to \alpha A' \\ A' &\to \beta|\gamma \end{aligned} \tag{3}$$

第四章英语

自顶向下

回溯分析程序

预测分析程序

递归下降分析

LL(1)分析

生成

LL(1)分析表

1、LL(1)的含义 L、L、1

2、First集合含义和构造方法、构造格式

3、Follow集合含义和构造方法、构造格式

4、LL(1)分析表的构造方法和结构

5、LL(1)文法判定定理

6、LL(1)分析过程(例)

7、消除左递归

8、提取左因子

格式相同、方法都是三步 遍历产生式

都是遍历产生式

过程：消

①消除左递归、提取左因子

②First集合

③Follow集合

④LL(1)分析表

⑤LL(1)分析