

# 编译原理

北方工业大学信息学院  
School of Information Science and Technology,  
North China University of Technology  
束劼  
[shujie@ncut.edu.cn](mailto:shujie@ncut.edu.cn)  
瀚学楼1122, 88801615

## 第四章 语法分析 -自上而下分析

## 第四章 语法分析-自上而下分析

- 本章目录
  - 4.1 语法分析器的功能
  - 4.2 自上而下分析面临的问题
  - 4.3 LL(1)分析法
  - 4.4 递归下降分析程序构造
  - 4.5 预测分析程序
  - 4.6 LL(1)分析中的错误处理

3

## 第四章 语法分析-自上而下分析

- 大纲要求
  - 掌握：LL(1)分析法的条件，消除左递归的算法，预测分析表的构造。
  - 理解：预测分析程序、递归下降分析程序的设计方法。
  - 了解：语法分析器的功能。

4

## 4.1 语法分析器的功能

### 4.1 语法分析器的功能

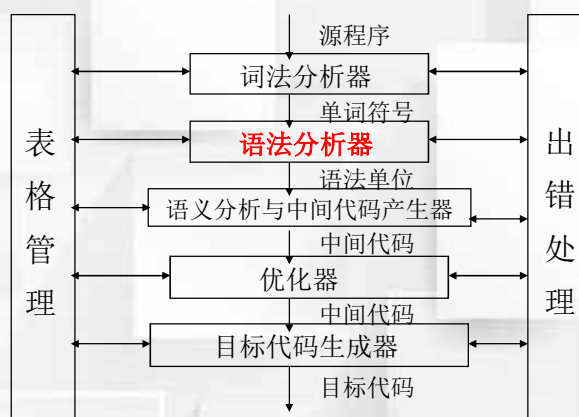
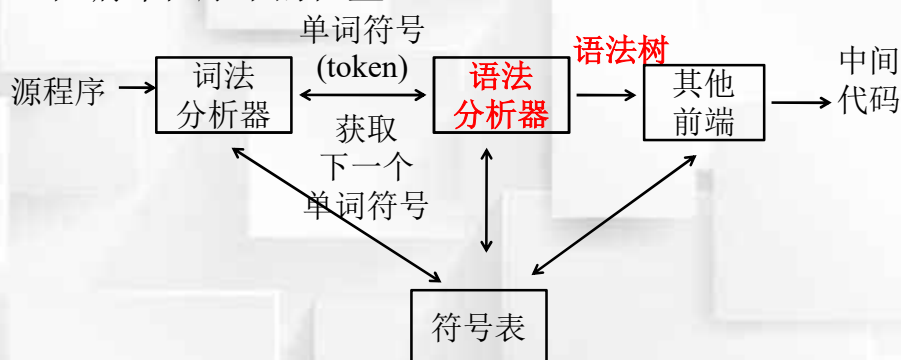


图1.1 编译程序总框

## 4.1 语法分析器的功能

### • 语法分析器(Parser)

在编译程序中的位置



7

## 4.1 语法分析器的功能

### • 语法分析器(Parser)

语言的语法结构是上下文无关文法(context-free)。

语法分析器的**本质**是按文法产生式，识别输入符号串是否为一个句子。建立一棵与输入符号串相匹配的语法分析树。

语法分析方法可以分三类：

1. 整体分析(universal), Cocke-Younger-Kasami algorithm 和 Earleys algorithm;

2. 自上而下分析法(top-down)

3. 自下而上分析法(bottom-up)

} 常用的两种方法

8

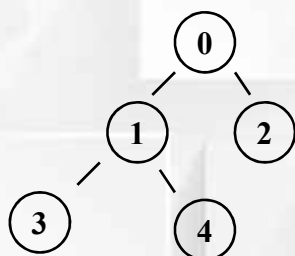
## 4.2 自上而下分析面临的问题

### 4.2 自上而下分析面临的问题

- 自上而下分析的方法

深度优先的树访问方式

从根结点开始，递归的访问各个结点的子结点。优先访问离根结点最远的未访问的结点，不一定要从左到右的访问，但可以设定先左后右的方式访问。



深度优先的访问顺序：  
0, 1, 3, 4, 2  
设定先左后右

## 4.2 自上而下分析面临的问题

### • 自上而下分析的方法

基本思想：可以看成是为输入字符串建立一棵语法分析树，从根结点开始，以深度优先的方式逐步生成语法分析树的结点。

例如：输入字符串为  $i + i * i$ , 文法

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid i$$

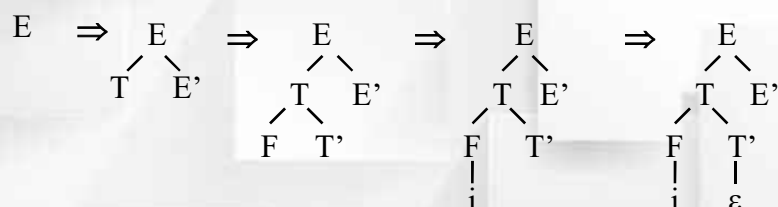
11

例如：输入字符串为  $i + i * i$ , 文法

$$E \rightarrow TE' \quad E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid i$$



12

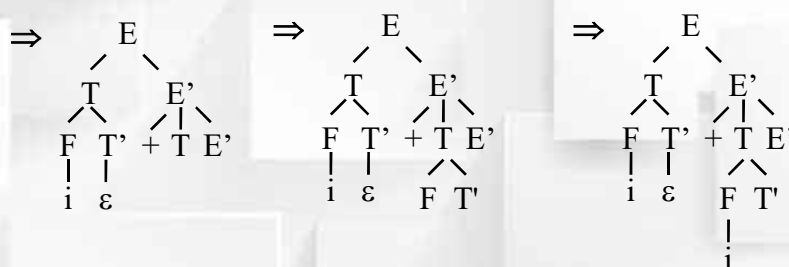
## 第四章 语法分析-自上而下分析 4.2 自上而下分析面临的问题

例如：输入字符串为  $i + i * i$ , 文法

$$E \rightarrow TE' \quad E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid i$$



13

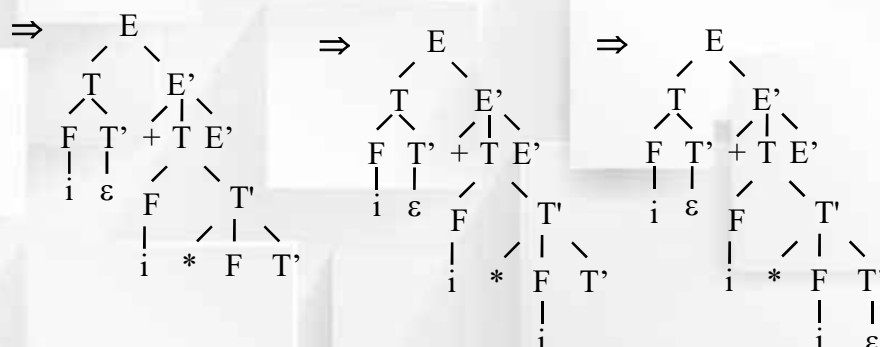
## 第四章 语法分析-自上而下分析 4.2 自上而下分析面临的问题

例如：输入字符串为  $i + i * i$ , 文法

$$E \rightarrow TE' \quad E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid i$$



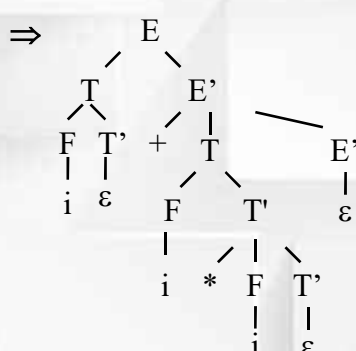
14

## 第四章 语法分析-自上而下分析 4.2 自上而下分析面临的问题

例如：输入字符串为  $i + i * i$ , 文法

$$E \rightarrow TE' \quad E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid i$$


• 自上而下分析的主旨：对任何输入串，试图用一切可能的办法，从文法开始符号(根结点)出发，自上而下地为输入串建立一棵语法树。或者说，为输入串寻找一个最左推导。

回溯，左递归

$$E' \rightarrow +TE'$$

$$T' \rightarrow *FT'$$

15

## 第四章 语法分析-自上而下分析 4.2 自上而下分析面临的问题

## 4.2 自上而下分析面临的问题

- 自上而下分析识别字符串的顺序

例 假定有文法  $G(S)$ :

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
①	$x*y$ ↑	$S$		根结 $S$ , 当前符 $x$
②	$x*y$ ↑	$\begin{array}{c} S \\ / \quad   \quad \backslash \\ x \quad A \quad y \end{array}$	$S \Rightarrow xAy$	用 $S \rightarrow xAy$ 展开 $S$ 欲用 $xAy$ 匹配输入串

16



## 4.2 自上而下分析面临的问题

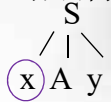

- 自上而下分析识别字符串的顺序

例 假定有文法G(S):

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
③	$x*y$ ↑			x得匹配, 移动IP
④	$x*y$ ↑		$S \Rightarrow xAy$ $\Rightarrow x**y$	试用 $A \rightarrow **$ 展开A

17

## 4.2 自上而下分析面临的问题


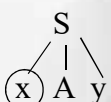
- 自上而下分析识别字符串的顺序

例 假定有文法G(S):

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
⑤	$x*y$ ↑			*得匹配, 移动IP, 但y得不到匹配
⑥	$x*y$ ↑		$S \Rightarrow xAy$	用 $A \rightarrow **$ 展开 失败, 回溯, 回到第③步

虚假匹配

18

4.2 自上而下分析面临的问题


• 自上而下分析识别字符串的顺序

例 假定有文法G(S):

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
⑦	$x*y$ ↑		$S \Rightarrow xAy$ $\Rightarrow x*y$	试用 $A \rightarrow *$ 展开 A

4.2 自上而下分析面临的问题

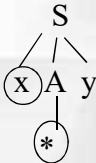
• 自上而下分析识别字符串的顺序

例 假定有文法G(S):

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
⑧	$x*y$ ↑			*得匹配, 移动IP

## 4.2 自上而下分析面临的问题

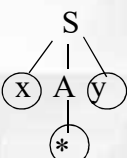
- 自上而下分析识别字符串的顺序

例 假定有文法G(S):

(1)  $S \rightarrow xAy$

(2)  $A \rightarrow **|*$

分析输入串  $x*y$

序号	指示器IP指向	语法树	最左推导	说明
⑨	$x*y$ ↑		$S \Rightarrow xAy$ $\Rightarrow x*y$	A完成匹配, y得匹配, 移动IP, 输入串匹配成功, 结束

21

## 4.2 自上而下分析面临的问题

- 自上而下分析面临的问题
- 问题一: 文法存在**左递归**, 将使自上而下的分析过程陷入无限循环。

$$P \xRightarrow{+} Pa$$

P无法匹配任何输入串, 回溯, 重新要求P进行新的匹配

22

## 4.2 自上而下分析面临的问题

- 自上而下分析面临的问题
- 问题二：**回溯**是一项复杂而费时的的工作，须废弃已做的许多工作，恢复到前面的某一情况，效率很低。

文法中非终结符A的产生式右部称为A的候选式，如果有多个候选式左端第一个符号相同，则语法分析程序无法根据当前输入符号选择产生式，只能试探。若不能匹配，则要**回溯**。

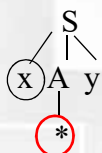
23

## 4.2 自上而下分析面临的问题

- 自上而下分析面临的问题
- 问题三：遇终结符匹配成功时，可能时暂时的成功。这就是**虚假匹配**。

输入串x\*y

x\*\*y  
↑



$S \Rightarrow xAy$   
 $\Rightarrow x*y$

试用 $A \rightarrow *$ 展开A

24

## 4.2 自上而下分析面临的问题

- 自上而下分析面临的问题
- 问题四：最终报告分析不成功时，难于知道输入串中出错的确切位置。**出错位置未知**。
- 问题五：带回溯的自上而下分析实际上采用了一种穷尽的试探法，**效率很低**，代价极高。这是一种理论上的方法，实践上价值不大。