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
瀚学楼1122，88801615

第五章 语法分析
-自下而上分析

第五章 语法分析-自下而上分析

- 本章目录

5.1自下而上分析基本问题
5.2算符优先分析
5.3 LR分析法
5.4语法分析器的自动产生工具YACC

第五章 语法分析-自下而上分析

- 大纲要求
- 掌握：归约，规范归约，算法优先分析法（算符优先文法、优先表的构造、算符优先分析算法、优先函数的构造）。
- 理解：符号栈的使用方法。
- 了解：自下而上语法分析的基本原理和工作方法，语法分析器的自动产生工具YACC的基本作用。

5.2.2 算符优先分析算法

5.2.2算符优先分析算法

- 算符优先分析算法概念

  可归约串，句型，短语，直接短语，句柄，规范归约。

  一个文法G的句型的素短语是指这样一个短语，它至少含有一个终结符，并且，除它自身之外不再含任何更小的素短语。

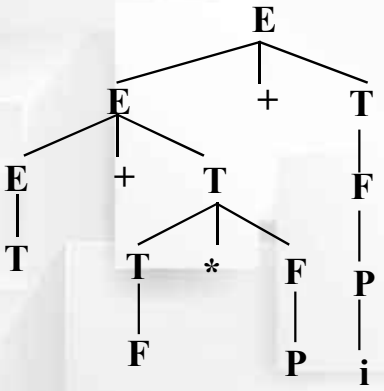
  最左素短语是指处于句型最左边的那个素短语。

5.2.2算符优先分析算法

- 算符优先分析算法概念
  - 考虑下面的文法G(E)：
    E→E+T | T
    T→T*F | F
    F→P↑F | P
    P→(E) | i

句型：T+F*P+i

求出短语、直接短语、句柄、素短语、最左素短语

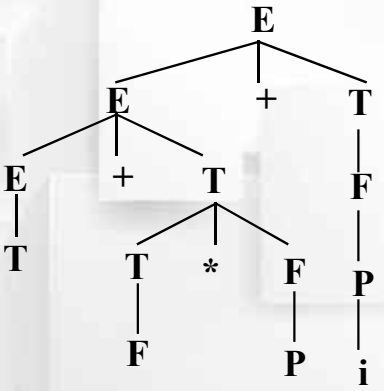


5.2.2算符优先分析算法

- 算符优先分析算法概念
  - 考虑下面的文法G(E)：
    E→E+T | T
    T→T*F | F
    F→P↑F | P
    P→(E) | i

句型：T+F*P+i

短语：T, F, P, i, F*P, T+(F*P), T+(F*P)+i
直接短语: T, F, P, i
句柄:T
素短语:i, F*P
最左素短语: F*P



5.2.2算符优先分析算法

- 算符优先分析算法概念
  算符优先文法句型(括在两个＃之间)的一般形式写成：

  $$\#N_1a_1N_2a_2\ldots N_na_nN_{n+1}\#$$

  其中，每个a_i都是终结符，N_i是可有可无的非终结符。

- 定理：一个算符优先文法G的任何句型的最左素短语是满足如下条件的最左子串 $N_ja_j\ldots N_ia_iN_{i+1}$，

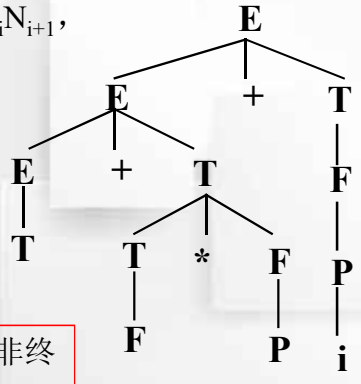
  $$a_{j-1} \lessdot a_j$$
  $$a_j \doteq a_{j+1},\ \ldots,\ a_{i-1} \doteq a_i$$
  $$a_i \gtrdot a_{i+1}$$

5.2.2算符优先分析算法

- 算符优先分析算法概念

  定理：一个算符优先文法G的任何句型的最左素短语是满足如下条件的最左子串 $N_ja_j\ldots N_ia_iN_{i+1}$，

  $a_{j-1} \lessdot a_j$
  a_{j-1}和a_j不在同一个子树
  $a_j \doteq a_{j+1},\ \ldots,\ a_{i-1} \doteq a_i$
  a_j到a_i在同一个子树
  $a_i \gtrdot a_{i+1}$
  a_i和a_{i+1}不在同一个子树



如何找到最左素短语可归约到的非终结符？

5.2.2算符优先分析算法

- 算符优先分析算法程序

```
k:=1;   S[k]:=‘#’;
REPEAT
 把下一个输入符号读进a中;
   IF S[k]∈VT THEN j:=k ELSE j:=k-1;
     WHILE S[j]⩾a DO
       BEGIN
         REPEAT
           Q:=S[j];
           IF S[j-1]∈VT THEN j:=j-1 ELSE j:=j-2
         UNTIL S[j]⩽Q;
        把S[j+1]...S[k]归约为某个N;
        k:=j+1;
        S[k]:=N
     END OF WHILE;
     IF S[j]⩽a OR S[j]=a THEN
       BEGIN k:=k+1;S[k]:=a END
     ELSE ERROR /*调用出错诊察程序*/
UNTIL a=‘#’
```

由于**非终结符**对归约没有影响，因此，非终结符可以不进符号栈S。

5.2.2算符优先分析算法

- 算符优先分析算法程序

使用一个符号栈S，用它寄存终结符和非终结符，k代表符号栈S的使用深度，a是输入符号栈。

```
k:=1;   S[k]:=‘#’;
REPEAT
 把下一个输入符号读进a中;
 算符优先分析算法;
UNTIL a=‘#’
```

在正确的情况下，算法工作完毕时，符号栈S应呈现：# N #。N是非终结符

5.2.2算符优先分析算法

- 算符优先分析算法程序

算符优先分析算法：
IF S[k]∈V_T THEN j:=k
ELSE j:=k-1;

DO WHILE S[j] ⩾ a

IF S[j] ⩽ a OR S[j] ⩵ a THEN
BEGIN
k:=k+1;
S[k]:=a
END
ELSE ERROR /*调用出错诊察程序*/

符号栈和输入符号同时为非终结符才符合WHILE循环条件，即**连续两个终结符**在输入中出现

在算法的工作过程中，若出现j减1后的值小于等于0时，则意味着输入串有错(ERROR)。

5.2.2算符优先分析算法

- 算符优先分析算法程序

算符优先分析算法：

WHILE S[j] ⩾a DO
BEGIN
REPEAT
Q:=S[j];
IF S[j-1]∈V_T THEN j:=j-1
ELSE j:=j-2
UNTIL S[j] ⩽ Q;
把S[j+1]...S[k]归约为某个N;
k:=j+1;
S[k]:=N
END OF WHILE;

由于**非终结符**对归约没有影响，因此，非终结符可以不进符号栈S。

在算符优先归约过程中，无法使用只有**单个非终结符**的产生式进行归约。因为没有定义**非终结符**的优先关系。

5.2.2算符优先分析算法

- 算符优先分析算法概念
- 只规定算符（终结符）之间的优先关系。在归约过程中只要找到**最左素短语**就归约，不必考虑归约到哪个非终结符，因此**不是规范归约**。
- 特点：速度快，**特别适合于表达式的分析**
- 通过**算符之间的优先关系**来确定最左素短语

5.2.2算符优先分析算法

- 算符优先分析算法程序

分析文法的句型T+T*F+i

例: 文法G[E]
E →E+T|T
T → T*F|F
F → (E)|i

从关系串中可以看到#低于左边第一个+，低于*。而*同时优先级高于右边的+，跟i的优先级无从判断(Do While)。不过右边+的优先级低于i。从左到右，最高优先级是*。所以第一步先归约*。

步骤	句型
1	#T⊥T*F⊥i#

5.2.2算符优先分析算法

- 算符优先分析算法程序

例: 文法G[E]
E →E+T|T
T → T*F|F
F → (E)|i

分析文法的句型T+T*F+i

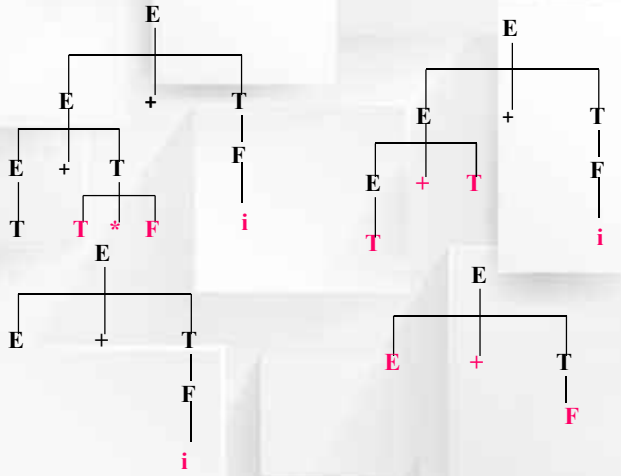
每次归约最左子串,确实是当前句型的**最左素短语**(语法树）

步骤	句型
1	#T⊥T*F⊥i#

17

5.2.2算符优先分析算法

- 算符优先分析算法程序



18

5.2.3 优先函数

5.2.3优先函数

- 优先函数

  把每个终结符θ与两个自然数f(θ)与g(θ)相对应，使得

  若$\theta_1 \lessdot \theta_2$，则$f(\theta_1) < g(\theta_2)$
  若$\theta_1 \doteq \theta_2$，则$f(\theta_1) = g(\theta_2)$
  若$\theta_1 \gtrdot \theta_2$，则$f(\theta_1) > g(\theta_2)$

  f称为入栈优先函数，g称为比较优先函数。

这两个函数是为了把终结符之间的优先关系，转换到一个特定数值，通过该数值再来判断归约顺序

20

5.2.3优先函数

- 优先函数

  优点:便于比较，节省空间，优先关系表占用存储量比较大；

  缺点:原来不存在优先关系的两个终结符，由于自然数相对应，变成可以比较的。要进行一些特殊的判断。

5.2.3优先函数

- 优先函数
  - 文法G(E)

    (1) E→E+T | T
    (2) T→T*F | F
    (3) F→P↑F | P
    (4) P→(E) | i

    的优先函数如下表

有许多优先关系表不存在优先函数

对应一个优先关系表的优先函数f和g不是唯一的，如果存在一对，就存在无穷对。

	+	*	↑	(	)	i	#
f	2	4	4	0	6	6	0
g	1	3	5	5	0	5	0

5.2.3优先函数

- 优先关系表构造函数
- 如果优先函数存在，则可以通过以下三个步骤从优先表构造优先函数:

(1) 对于每个终结符a（包括#），令其对应两个符号f_a和g_a，画一个以所有符号为结点的方向图。

如果a$\gtrdot$b，则从f_a画一条弧至g_b，如果a$\lessdot$b，则从g_b画一条弧至f_a 。

5.2.3优先函数

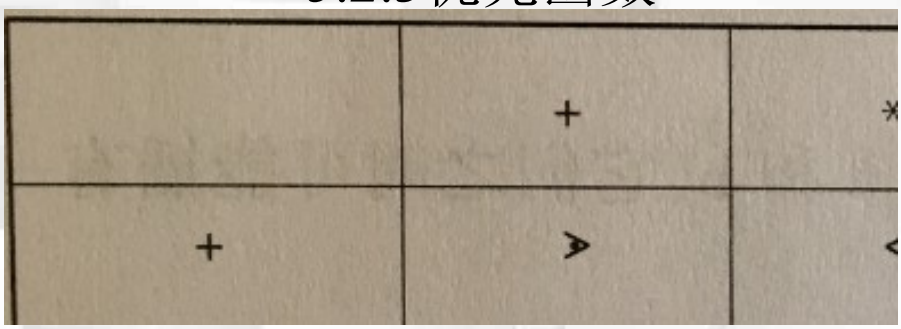
- 优先关系表构造函数

(2) 对每个结点都赋予一个数，此数等于从该结点出发所能到达的结点(包括出发点自身在内)的个数。赋给f_a的数作为f(a)，赋给g_b的数作为g(b)。

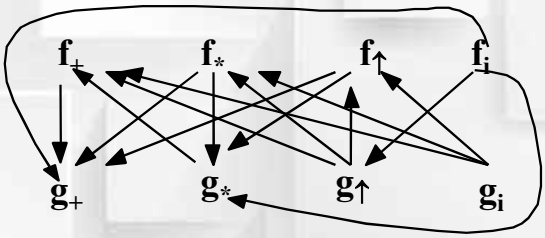
达到的结点如果还能到其他结点，也算进去

(3) 检查所构造出来的函数f和g是否与原来的优先关系表矛盾。若没有矛盾，则f和g就是要求的优先函数，若有矛盾，则不存在优先函数。

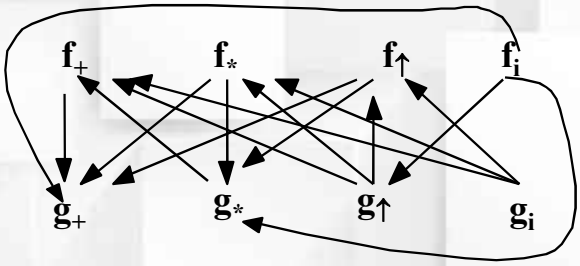
5.2.3优先函数



看第一列的
优先符号



5.2.3优先函数



	+	*	↑	i
f	2	4	4	7
g	1	3	6	6

5.2.3优先函数

- 优先函数特点

（1）优先函数值不唯一

（2） 优点：

节省内存空间

若文法有n个终结符，则优先关系表（矩阵）为n^2

而优先函数为2n

易于比较：算法上容易实现，数与数比，不必查优先关系表（矩阵）。

（3） 缺点：可能掩盖错误。

27

5.2.4 算符优先分析中的出错处理

5.2.4算符优先分析中的出错处理

- 错误类型

使用算符优先分析法时，可在两种情况下，发现语法错误：

① 若找到某一“句柄”（此处“句柄”指最左素短语），但不存在任一产生式的右部为此“句柄”。

② 若栈顶终结符号与下一输入符号之间不存在任何优先关系。

5.2.4算符优先分析中的出错处理

- 解决方案

① 若找到某一“句柄”（此处“句柄”指最左素短语），但不存在任一产生式的右部为此“句柄”。

解决方案：

a. 打印错误信息；

b. 删除句柄中的某个字符。

例如句柄为abc时，没有产生式包含这三个字符一起，但有aAcB, 则给出错误信息“非法b”，若产生式为abdc，则给出错误信息“缺少d”。

5.2.4算符优先分析中的出错处理

- 错误类型

② 若栈顶终结符号与下一输入符号之间不存在任何优先关系。

解决方案：

a. 改变、删除、插入；

在算符优先表中的空白处加入特定符号

5.2.4算符优先分析中的出错处理

- 错误解决方案

优先关系表(包括出错子程序)

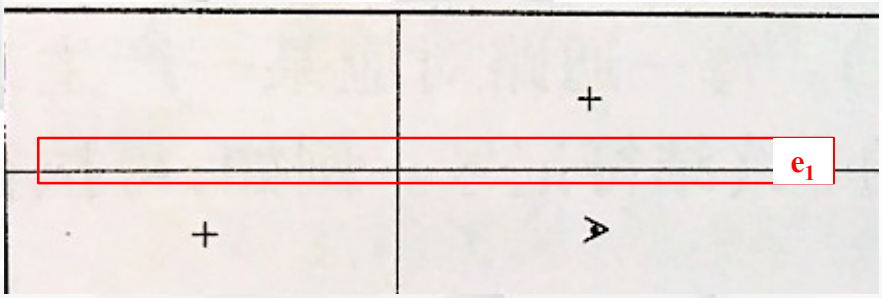
	+	
+	>	

特殊字符e_1, e_2, e_3, e_4

5.2.4算符优先分析中的出错处理

- 错误解决方案

优先关系表(包括出错子程序)



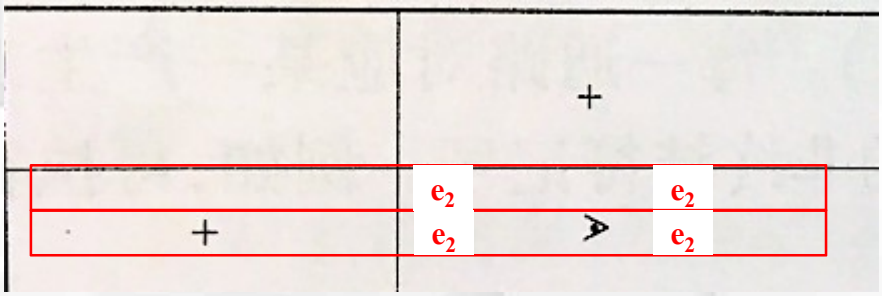
e_1 当表达式以左括号结尾时，调用此程序
将'('从栈顶移去；给出错误信息：非法左括号。

33

5.2.4算符优先分析中的出错处理

- 错误解决方案

优先关系表(包括出错子程序)



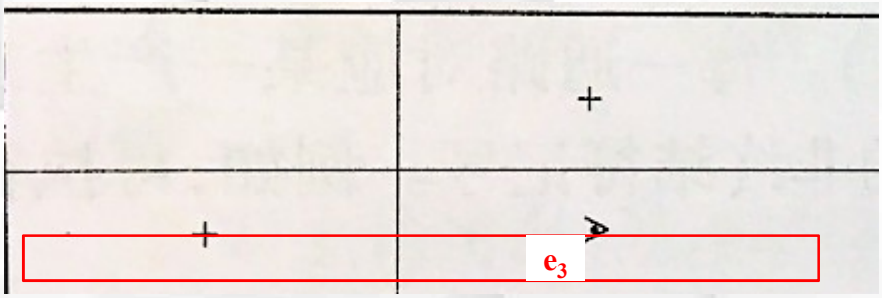
e_2 当i或）后跟（时，调用此程序
在输入端插入'+'；给出错误信息：缺少运算符。

34

5.2.4算符优先分析中的出错处理

- 错误解决方案

优先关系表(包括出错子程序)



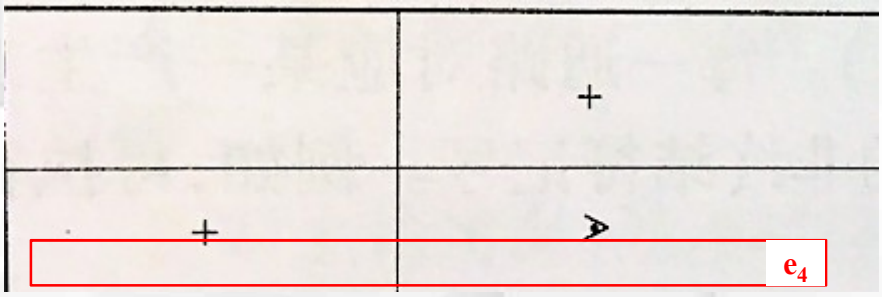
e_3当表达式以右括号开始时，调用此程序
从输入端删除')'；给出错误信息：非法右括号。

35

5.2.4算符优先分析中的出错处理

- 错误解决方案

优先关系表(包括出错子程序)



e_4若栈顶有非终结符E，则表达式分析完毕。
若为空，则在输入端插入i；给出错误信息：缺少表达式。

36

5.2.4算符优先分析中的出错处理

- 归约出现问题时的校正子程序
  - 文法G(E)

    (1) E→E+T | T
    (2) T→T*F | F
    (3) F→P↑F | P
    (4) P→(E) | i

    的优先函数如下表

1. 如+或*被规约，则检查其两端是否出现非终结符。否则，打印错误信息：“缺表达式”
2. 如i被规约，则检查其左端或右端是否有非终结符。如果有，则给出信息：“表达式无运算符连接”

5.2.4算符优先分析中的出错处理

- 归约出现问题时的校正子程序
  - 文法G(E)

    (1) E→E+T | T
    (2) T→T*F | F
    (3) F→P↑F | P
    (4) P→(E) | i

    的优先函数如下表

3. 如果( )被规约，则检查是否在括号间有一非终结符。如果没有，则给出信息：“括号无表达式”。

第五章 小结

第五章 小结

5.1自下而上分析基本问题

5.2算符优先分析

Coursework

5.1令文法G_1为

E→E+T｜T

T→T*F｜F

F→(E)｜i

证明E+T*F是它的一个句型，指出这个句型的所有短语，直接短语和句柄。

Coursework

5.2考虑下面的表格结构文法G_2：

S→a｜∧｜(T)

T→T，S｜S

（1）给出(a，(a，a))和(((a，a)，∧，(a))，a)的最左和最右推导。

（2）指出(((a，a)，∧，(a))，a)的规范归约及每一步的句柄。根据这个规范归约，给出“移进-归约”的过程，并给出它的语法树自下而上的构造过程。

Coursework

5.3（1）计算练习5.2文法G_2的FIRSTVT和LASTVT。

（2）计算G_2的优先关系。G_2是一个算符优先文法吗？

（3）计算G_2的优先函数。

（4）给出输入串(a，(a，a)) 的算符优先分析过程。