南京理工大学

编译原理

项欣光

计算机科学与工程学院

第5章　自底向上优先分析法

- 自底向上优先分析概述
- 简单优先分析
- 算符优先分析

- 自底向上分析方法，也称移进-归约分析法。
- 实现思想：
  - 对输入符号串自左向右进行扫描，并将输入符逐个移入一个后进先出栈中，边移入边分析，一旦栈顶符号串形成某个句型的句柄或可归约串时，就用该产生式的左部非终结符代替相应右部的文法符号串，这称为归约。
  - 重复这一过程，直到栈中只剩文法的开始符号时，则分析成功，也就确认输入串是文法的句子。

例1 文法G[S]：

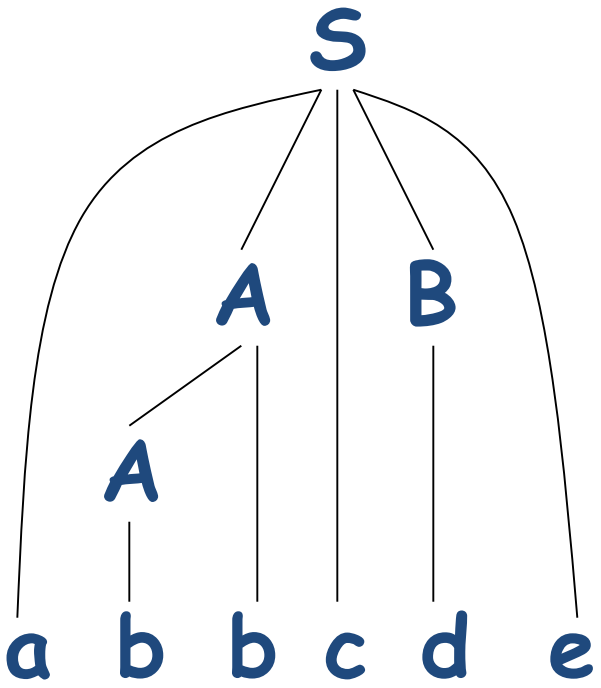
(1) S → aAcBe

(2) A → b

(3) A → Ab

(4) B → d

分析 abbcde



对输入串 #abbcde# 的移进-归约分析过程：

步骤	符号栈	输入符号串	动作
1）	#	abbcde#	移进
2）	#a	bbcde#	移进
3）	#ab	bcde#	归约(A→b)
4）	#aA	bcde#	移进
5）	#aAb	cde#	归约(A→Ab)
6）	#aA	cde#	移进
7）	#aAc	de#	移进
8）	#aAcd	e#	归约(B→d)
9）	#aAcB	e#	移进
10）	#aAcBe	#	归约(S→aAcBe)
11）	#S	#	接受

自底向上分析法

- 自底向上分析的策略：移进-归约分析。
  - 移进就是将一个终结符推进符号栈
  - 归约就是将0个或多个符号从栈中弹出，根据产生式将一个非终结符压入符号栈
- 移进-归约过程是规范推导（最右推导）的逆过程，所以它是规范归约。
- 何时移进？何时归约？
- 自底向上分析的关键：在分析过程中如何确定“句柄”。即：确定何时可以归约栈顶的符号串。

5.1 自底向上优先分析法概述

优先分析法分类：

- 简单优先分析法
  - 先按照一定原则，求出文法所有符号（V_T和V_N）之间的优先关系；再按照这种关系确定归约过程中的句柄。
  - 优点：规范归约，分析准确、规范
  - 缺点：分析效率低，实用价值不大
- 算符优先分析法
  - 先按照一定原则，求出文法所有V_T之间的优先关系；归约时，只要遇到可归约串就归约。（并不考虑归约到哪个非终结符）
  - 优点：分析速度快，特别适于表达式的分析
  - 缺点：不是规范归约。（采取适当措施克服）

5.2 简单优先分析法

- 主要思想：

  先按照一定原则，求出文法所有符号（V_T和V_N）之间的优先关系；再按照优先关系确定归约过程中的句柄。

- 实现步骤：

  1. 拓广文法 S’→#S#
  2. 构造优先关系表
  3. 判断是否为简单优先文法
  4. 根据优先关系表分析句子

下一节

优先关系

- $X \doteq Y$ ：X与Y优先关系相等

  文法G中存在产生式A→...XY...

  不等价于 $Y \doteq X$

- $X \lessdot Y$ ：X的优先性比Y小

  文法G中存在产生式A→...XB...，

  且B $\overset{+}{\Rightarrow}$ Y...

  不等价于 $Y \gtrdot X$

- $X \gtrdot Y$ ：X的优先性比Y大

  文法G中存在产生式A→...BD...，

  且B $\overset{+}{\Rightarrow}$ ...X，D $\overset{*}{\Rightarrow}$ Y...

  不等价于 $Y \lessdot X$

注：其中 $X, Y \in (V_T \cup V_N)$

构造优先关系表（优先关系矩阵）

例2 拓广后的文法G：

(0) S′ → #S#　//特殊：# $\doteq$ #，# $\lessdot$ 右相邻符号，左相邻符号 $\gtrdot$ #

(1) S → bAb

(2) A → (B

(3) A → a

(4) B → Aa)

求各符号之间的优先关系，并求该文法的优先关系矩阵。

解题步骤：

1）求各种优先关系

①求 $\doteq$ 关系

在产生式右部找相邻的符号V1V2，则 V1 $\doteq$ V2

②求 $\lessdot$ 关系

在产生式右部找V_TV_N形式，则 $V_T \lessdot$ a，其中$V_N \Rightarrow$a…

在产生式右部找$V_{N1}V_{N2}$形式，则 $V_{N1} \lessdot$ a，其中$V_{N2} \Rightarrow$ a…

③求 $\gtrdot$ 关系

在产生式右部找V_NV_T形式，则 b $\gtrdot$ V_T，其中$V_N \Rightarrow$…b

在产生式右部找$V_{N1}V_{N2}$形式，则 b $\gtrdot$ V_{N2}，其中$V_{N1} \Rightarrow$ …b

2）根据优先关系，构造优先关系矩阵

构造优先关系表（优先关系矩阵）

例2 拓广后的文法G：

(0) S' → #S#

(1) S → bAb

(2) A → (B

(3) A → a

(4) B → Aa)

1）求各种优先关系

① 求 $\doteq$ 关系

$\doteq$ #，b $\doteq$ A，A $\doteq$ b，($\doteq$ B，A $\doteq$ a，a $\doteq$)

② 求 $\lessdot$ 关系

$\lessdot$ S，# $\lessdot$ b，b $\lessdot$ a，b $\lessdot$ (，($\lessdot$ A，($\lessdot$ a，($\lessdot$ (

③ 求 $\gtrdot$ 关系

S $\gtrdot$ #，b $\gtrdot$ #，a $\gtrdot$ b，B $\gtrdot$ b，) $\gtrdot$ b，a $\gtrdot$ a，B $\gtrdot$ a，) $\gtrdot$ a

构造优先关系表（优先关系矩阵）

例2 拓广后的文法G：

(0) S' → #S#
(1) S → bAb
(2) A → (B
(3) A → a
(4) B → Aa)

优先关系：

#=#，b=A，A=b，(=B，A=a，a=)
#<S，#<b，b<a，b<(，(<A，(<a，(<(
S>#，b>#，a>b，B>b，)>b，a>a，B>a，)>a

2）根据优先关系，构造优先关系矩阵

	S	A	B	(	)	a	b	#
S								⋗
A						⋗	⋗	
B						⋗	⋗	
(		⋖	⋗	⋖		⋖		
)						⋗	⋗	
a					⋗	⋗	⋗	
b		⋗		⋖		⋖		⋗
#	⋖						⋖	⋗

返

简单优先文法的定义

满足以下条件的文法是简单优先文法

（1）在文法符号集V中，任意两个符号之间最多只有一种优先关系。

（2）在文法中，任意两个产生式没有相同的右部。

（3）不含空产生式。

采用简单优先分析时，必须是简单优先文法。

返回

简单优先分析法

构造相应优先关系矩阵，并将文法的产生式保存，设置符号栈S，算法步骤如下：

1. 将输入符号串$a_1a_2a_3...a_n\#$ 依次逐个存入符号栈S中，直到遇到栈顶符号a_i $\gtrdot$ 下一个待输入符号a_j 时为止。
2. 栈顶当前符号a_i为句柄尾，由此向左在栈中找句柄的头符号a_k，即找到$a_{k-1} \lessdot a_k$为止。
3. 找到句柄$a_k...a_i$，在文法的产生式中查找右部为$a_k...a_i$的产生式，若找到则用相应左部代替句柄，若找不到则为出错，这时可断定输入串不是该文法的句子。
4. 重复上述三步，直到归约完所有输入符号串为止。（此时栈中只剩文法的开始符号）

例2 文法G[S]：

(1) S → bAb

(2) A → (B

(3) A → a

(4) B → Aa)

分析输入串 #b(aa)b#

	S	A	B	(	)	a	b	#
S								⋗
A						⋗	⋗	
B						⋗	⋗	
(		⋖	⋗	⋖		⋖		
)						⋗	⋗	
a					⋗	⋗	⋗	
b		⋗		⋖		⋖		⋗
#	⋖						⋖	⋗

步骤	符号栈S	待输入符号串	动作
1）	#	b(aa)b#	#<b,移进
2）	#b	(aa)b#	b<(,移进
3）	#b(	aa)b#	(<a,移进
4）	#b(a	a)b#	a>a,归约A→a
5）	#b(A	a)b#	A=a,移进
6）	#b(Aa	)b#	a=),移进
7）	#b(Aa)	b#	)>b,归约B→Aa)
8）	#b(B	b#	B>b,归约A→(B
9）	#bA	b#	A=b,移进
10）	#bAb	#	b>#,归约S→bAb
11）	#S	#	接受

返回

5.3 算符优先分析法

- 主要思想：

  对文法按照一定规则，求出V_T之间的优先关系；再按照这种优先关系来确定可归约串。

- 实现步骤：

  1. 拓广文法 S'→#S#
  2. 构造算符优先关系表
  3. 判断是否为算符优先文法（OPG文法）
  4. 根据优先关系表分析句子

- 优先函数

  为节约存储空间，用“优先函数”代替“优先关系表”

直观法确定算符优先关系

例3 文法G[E]：

E→E+E | E-E

E→E*E | E/E

E→E↑E

E→(E)

E→i

算符优先关系表：

	+	-	*	/	↑	(	)	i	#
+	>	>	<	<	<	<	>	<	>
-	>	>	<	<	<	<	>	<	>
*	>	>	>	>	<	<	>	<	>
/	>	>	>	>	<	<	>	<	>
↑	>	>	>	>	<	<	>	<	>
(	<	<	<	<	<	<	=	<	
)	>	>	>	>	>		>		>
i	>	>	>	>	>		>		>
#	<	<	<	<	<	<		<	=

- i的优先级最高
- ↑优先级次于i，右结合
- *和/优先级次之，左结合
- +和-优先级最低，左结合
- 括号的优先级大于括号外的运算符，小于括号内的运算符
- 内括号的优先性大于外括号
- #的优先性低于与其相邻的算符

算符文法的定义

- 算符文法：上下文无关文法G中没有形如 A→...BC... 的产生式，其中B,C∈V_N，则称 G为算符文法（OG，Operator Grammer）。
- 性质1：在算符文法中任何句型都不包含两个相邻的非终结符。
- 性质2：如 Ab 或 bA 出现在算符文法的句型 α 中，其中 A∈V_N，b∈V_T，则 α 中任何含 b 的短语必含有 A 。（但含A的短语未必含b。）

算符优先关系的定义

设G[S]是一个不含ε产生式的算符文法G中

- $a \doteq b$ ： 当且仅当文法中含有形如A→...ab...或A →...aBb...的产生式
- $a \lessdot b$ ： 当且仅当文法中含有形如A→...aB...的产生式，且 $B \overset{+}{\Rightarrow} b...$ 或 $B \overset{+}{\Rightarrow} Cb...$
- $a \gtrdot b$ ：当且仅当文法中含有形如 A→...Bb...的产生式，且 $B \overset{+}{\Rightarrow} ...a$ 或 $B \overset{+}{\Rightarrow} ...aC$

算符优先文法的定义

- 设有一不含 ε 产生式的算符文法G，如果对任意两个终结符a和b之间至多只有 $\doteq$ 、$\lessdot$ 、$\gtrdot$ 三种关系的一种成立，则称 G是一个算符优先文法（OPG ，Operator Precedence Grammar）。
  - 不含空产生式
  - 任何产生式右部不包含两个相邻的非终结符
  - 任何两个终结符之间优先关系唯一
- 算符优先文法是无二义的。

返回

算符优先关系表的构造

定义法
关系图法

定义法 构造算符优先关系表

（1）定义 FirstVT 和 LastVT

FirstVT(B)={b|B $\overset{+}{\Rightarrow}$ b... 或 B $\overset{*}{\Rightarrow}$ Cb...}

LastVT(B)={a|B $\overset{+}{\Rightarrow}$ ...a 或 B $\overset{*}{\Rightarrow}$ ...aC}

（2）求优先关系

a $\doteq$ b	A→…ab 或 A→…aBb…
a $\lessdot$ FirstVT(B)	A→…aB…
LastVT(B) $\gtrdot$ c	A→…Bc…

（3）构造优先关系表

例4 文法G[E]

(0) E'→#E#

(1) E→E+T

(2) E→T

(3) T→T*F

(4) T→F

(5) F→P^F

(6) F→P

(7) P→(E)

(8) P→i

构造算符优先关系表。

构造优先分析表的步骤：

1）计算每个V_N的$FirstV_T$集和$LastV_T$集

2）求优先关系

- 求=关系
- 求<关系：找…aB…，a<FirstVT(B)
- 求>关系：找…Bc…，LastVT(B)>c

3）构造优先关系表

返回

例4 文法G[E]

(0) E'→#E#

(1) E→E+T

(2) E→T

(3) T→T*F

(4) T→F

(5) F→P^F

(6) F→P

(7) P→(E)

(8) P→i

构造算符优先关系表。

V_N的FirstVT集合和LastVT集合

» $FirstV_T(E')$= #

» $FirstV_T(E)$= +，*，↑，(，i

» $FirstV_T(T)$= *，↑，(，i

» $FirstV_T(F)$= ↑，(，i

» $FirstV_T(P)$= (，i

» $LastV_T(E')$= #

» $LastV_T(E)$= +，*，↑，i，)

» $LastV_T(T)$= *，↑，i，)

» $LastV_T(F)$= ↑，i，)

» $LastV_T(P)$= i，)

例4 文法G[E]

(0) E'→#E#
(1) E→E+T
(2) E→T
(3) T→T*F
(4) T→F
(5) F→P^F
(6) F→P
(7) P→(E)
(8) P→i

构造算符优先关系表。

求优先关系

求=关系：# = # (0)　　(=) (6)

求<关系　逐条扫描产生式，寻找形如：A→…aB…的产生式。

由于 #E 故　# < $FirstV_T$ (E)

由于 +T 故　+ < $FirstV_T$ (T)

由于 *F 故　* < $FirstV_T$ (F)

由于 ↑F 故　↑ < $FirstV_T$ (F)

由于 (E 故　(< $FirstV_T$ (E)

求>关系 逐条扫描产生式，寻找形如：A→…Bb…的产生式。

由于 E# 故　$LastV_T$ (E) > #

由于 E+ 故　$LastV_T$ (E) > +

- 由于 T* 故　$LastV_T$ (T) > *
- 由于 P↑故　$LastV_T$ (P) > ↑
- 由于 E) 故　$LastV_T$ (E) >)

算符优先分析

举例:利用例4的算符优先分析表分析句子 #i+i#

	+	*	^	(	)	i	#
+	>	<	<	<	>	<	>
*	>	>	<	<	>	<	>
^	>	>	<	<	>	<	>
(	<	<	<	<	=	<	
)	>	>	>		>		>
i	>	>	>		>		>
#	<	<	<	<		<	=

步骤	栈	优先关系	当前符号	待输入串	动作
1	#	⋖	i	+i#	移进
2	#i	⋗	+	i#	归约
3	#F	⋖	+	i#	移进
4	#F+	⋖	i	#	移进
5	#F+i	⋗	#		归约
6	#F+F	⋗	#		归约
7	#F	=	#		接受

算符优先分析

- 归约过程中，只考虑终结符之间的优先关系来确定句柄，而与非终结符无关。这样去掉了单个非终结符的归约，所以用算符优先分析法的规约过程不是规范归约。
- 为解决在算符优先分析过程中如何寻找句柄，引进最左素短语的概念

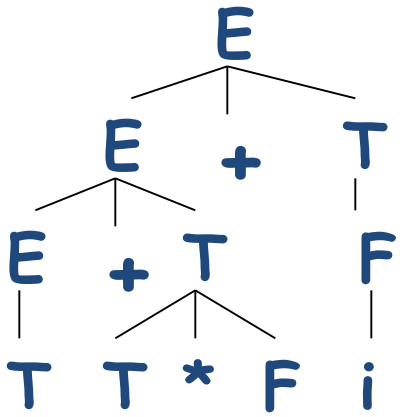
最左素短语

- 素短语
  设有文法G[S]，其句型的素短语是一个短语，它至少包含一个终结符，且除自身外不再包含其他素短语。
- 最左素短语：句型最左边的素短语。

文法G[E]：

(1) E→E+T
(2) E→T
(3) T→T*F
(4) T→F
(5) F→P↑F
(6) F→P
(6) P→(E)
(7) P→i

求句型 #T+T*F+i# 的素短语。



短语：

~~T+T*F+i~~
~~T+T*F~~
~~T~~
T*F
i

算符优先分析的局限性

- 简单优先分析：是规范归约，关键是寻找当前句型的句柄，符号栈顶一旦形成句柄就归约。
- 算符优先分析：不是规范归约，关键是寻找当前句型的最左素短语，符号栈顶一旦形成最左素短语就归约。

因此，算符优先分析可能出现“错误的句子得到正确的归约”；并且一般语言的文法很难满足算符优先文法的条件。

结论：算符优先分析法只适用于表达式的语法分析。

返回




优先函数

- 优点：优先函数比优先矩阵节省空间
  - 优先矩阵占用内存空间：$(n+1)^2$
  - 优先函数占用内存空间：$2(n+1)$
- 缺点：当发生错误时不能准确指出出错位置
- 优先函数的构造方法：
  - 由定义直接构造
  - 用关系图构造优先函数

优先函数的定义

- 定义两个函数f,g ，满足如下条件：
  - 当a $\doteq$ b，则令f(a)=g(b)
  - 当a $\lessdot$ b，则令f(a)<g(b)
  - 当a $\gtrdot$ b，则令f(a)>g(b)
- 由定义构造优先函数

  a) 对每个终结符a，令f(a)=g(a)=1

  b) 若a $\gtrdot$ b，而f(a)≤g(b)则令f(a)=g(b)+1

  c) 若a $\lessdot$ b，而f(a)≥g(b)则令g(b)=f(a)+1

  d) 若a $\doteq$ b，而f(a)≠g(b)则令

  min(f(a),g(b))=max(f(a),g(b))

  e) 重复b - d，直到过程收敛（重复过程中，若有一个值大于2n，表明该文法不存在算符优先函数）

求例4优先关系表
所对应的优先函数

	+	*	↑	(	)	i	#
+	>	<	<	<	>	<	>
*	>	>	<	<	>	<	>
↑	>	>	<	<	>	<	>
(	<	<	<	<	=	<	
)	>	>	>		>		>
i	>	>	>		>		
#	<	<	<	<		<	=

迭代次数		+	*	↑	(	)	i	#
0（初值）	f	1	1	1	1	1	1	1
	g	1	1	1	1	1	1	1
1	f	2	4	4	1	6	6	1
	g	2	3	5	5	1	5	1
2	f	3	5	5	1	7	7	1
	g	2	4	6	6	1	6	1
3	f	同第 2 次迭代结果						
	g							

优先函数

- 1、同一文法的优先关系矩阵对应的优先函数不唯一
- 2、有些优先矩阵不存在对应的优先函数
- 3、利用优先函数进行句子分析时，
  - 若a为栈顶符号，b为当前待输入符号
  - 当f(a)<=g(b)时，b移进
  - 当f(a)>g(b)时，对栈顶进行归约

南京理工大学

谢谢各位同学！