编译原理　第四章　语法分析(Syntax Analysis)

内容回顾

提出问题：其它上下文无关文法

4.1 自上而下的语法分析(TOP-DOWN PARSING)

4.2 自下而上的语法分析(BOTTOM-UP PARSING)

1、 一般分析方法　移进-归约（Shift-Reduce）　最左归约

2、 简单优先分析

简单优先文法　最左归约　句柄

3 、算符优先分析

算符优先文法　“最左归约”　最左素短语

4、 LR分析法

4.2.4 LR分析法 (LR PARSERS)

L R（K） LR(0)、SLR(1)、LR(1)……

- 向前看K个符号能确定用哪个产生式归约
- 产生一个最右推导(Rightmost)的逆过程
- 自左向右扫描输入符号(Left-to-right)

- 解决所有无二义性的上下文无关文法
- 严格的最左归约----句柄
- 语法分析程序自动生成器YACC(Yet Another Compiler-Compiler)

4.2.4 LR分析法 (LR PARSERS)

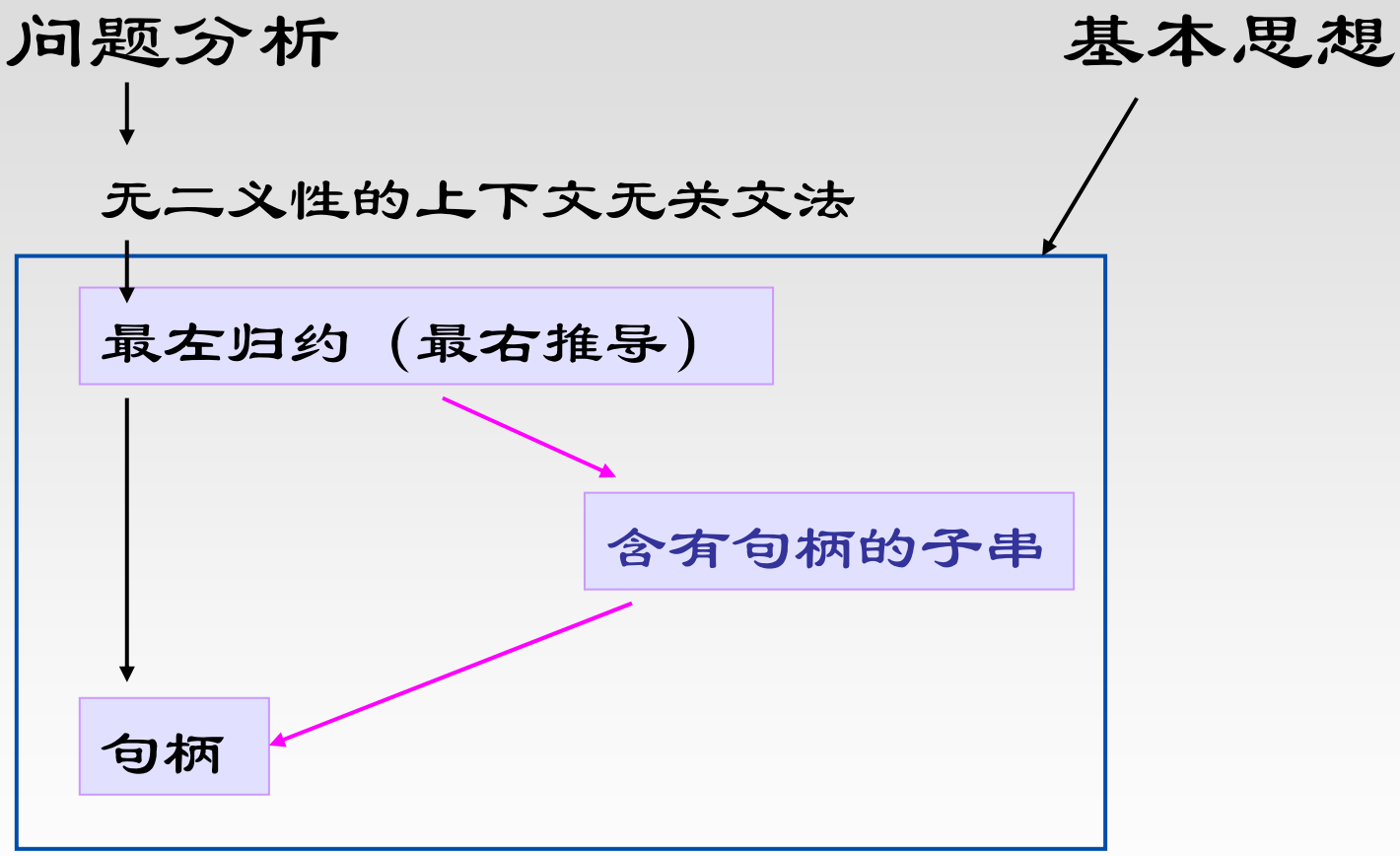
主要内容

- 基本概念
- LR(0) 分析法
- SLR(1) 分析法
- LR(1) 分析法
- LALR(1) 分析法

重点掌握

- LR(0)、SLR(1)、LR(1)等文法定义
- LR(0)和SLR(1) 分析器的构造

4.2.4 LR分析法 (LR PARSERS)



4.2.4 LR分析法 ❑基本概念

一、基本概念

❑可归前缀

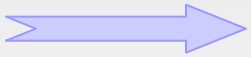
❑活前缀

例：G[S]:

S→aBEb

B→BaE｜c

E→e



S→aBEb[1]

B→BaE[2]

B→c[3]

E→e[4]

问题：将规则编上号,并将序号人为带入句型分析中，给出句子 acaeeb 的规范推导过程；

4.2.4 LR分析法 ❑基本概念

S→aBEb[1]
B→BaE[2]
B→c[3]
E→e[4]

S=>aBEb[1]
=>aBe[4]b[1]
=>aBaE[2]e[4]b[1]
=>aBae[4][2]e[4]b[1]
=>ac[3]ae[4][2]e[4][2]e[4]b[1]

规范推导

acaeeb

4.2.4 LR分析法

❑基本概念

S→aBEb[1]
B→BaE[2]
B→c[3]
E→e[4]

acaeeb

S=>aBEb[1]
=>aBe[4]b[1]
=>aBaE[2]e[4]b[1]
=>aBae[4][2]e[4]b[1]
=>ac[3]ae[4][2]e[4][2]e[4]b[1]

当前句型的句柄

aBEb[1]
aBe[4]
aBaE[2]
aBae[4]
ac[3]

可归前缀

4.2.4 LR分析法 ❑基本概念

S→aBEb[1]
B→BaE[2]
B→c[3]
E→e[4]

特点：

❑可归前缀的后半部分总是包含当前句型的句柄；

❑可归前缀含有用哪一个产生式进行归约的信息；

aBEb[1]
aBe[4]
aBaE[2]
aBae[4]
ac[3]

} 可归前缀

4.2.4 LR分析法 ❑基本概念

1、可归前缀

形式为 $\beta\ \omega$ [p]

其中: $\beta \in V^*$

p为规则序号,

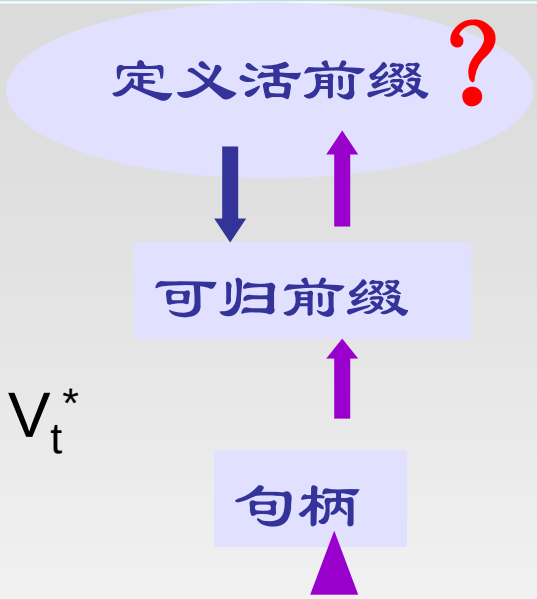
ω 为第p条规则右部, $B \rightarrow \omega$

可归前缀中应包含的信息

- ❑句柄（在最后）
- ❑用哪条产生式进行归约

4.2.4 LR分析法 ❑基本概念

2、活前缀(Viable Prefix)



对于最右推导过程

$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots\cdots \Rightarrow \alpha_m = x$

若 $\alpha_i = \phi Bt$ 且 $B \rightarrow \beta$，$\phi \in V^*$，$t \in V_t^*$

则存在最右推导 $\phi Bt \Rightarrow \phi \beta t$

令 $\phi \beta = u_1 u_2 \cdots\cdots u_r$ $u_i \in V$

则 $u_1 u_2 \cdots\cdots u_i$ $(1<=i<=r)$ 为句型 $\phi \beta t$ 的活前缀

- ❖最长的活前缀就是可归前缀
- ❖ ε 是句型 $\varphi \beta t$ 的活前缀

4.2.4 LR分析法 ❑基本概念

例：G[S]:E→E+T｜E-T｜T

T →i｜(E)

拓广文法：G′

S →E#

E→E+T｜E-T｜T

T →i｜(E)

句型：E-(i+i)#

有S=+>E-(T+i)#=>E-(i+i)#

活前缀：E　E-　E-(　E-(i　(可归前缀）

问题：用什么样的方法来识别活前缀？　FA　M

❖一个文法所有规范句型的活前缀（可归前缀），

能够为有限自动机所识别。

二、LR(0)分析法

在归约时不向前看任何一个符号就能确定用哪一个产生式

G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]

其识别可归前缀的有限自动机如图 DFA M

4.2.4 LR分析法　　❑LR(0)分析法

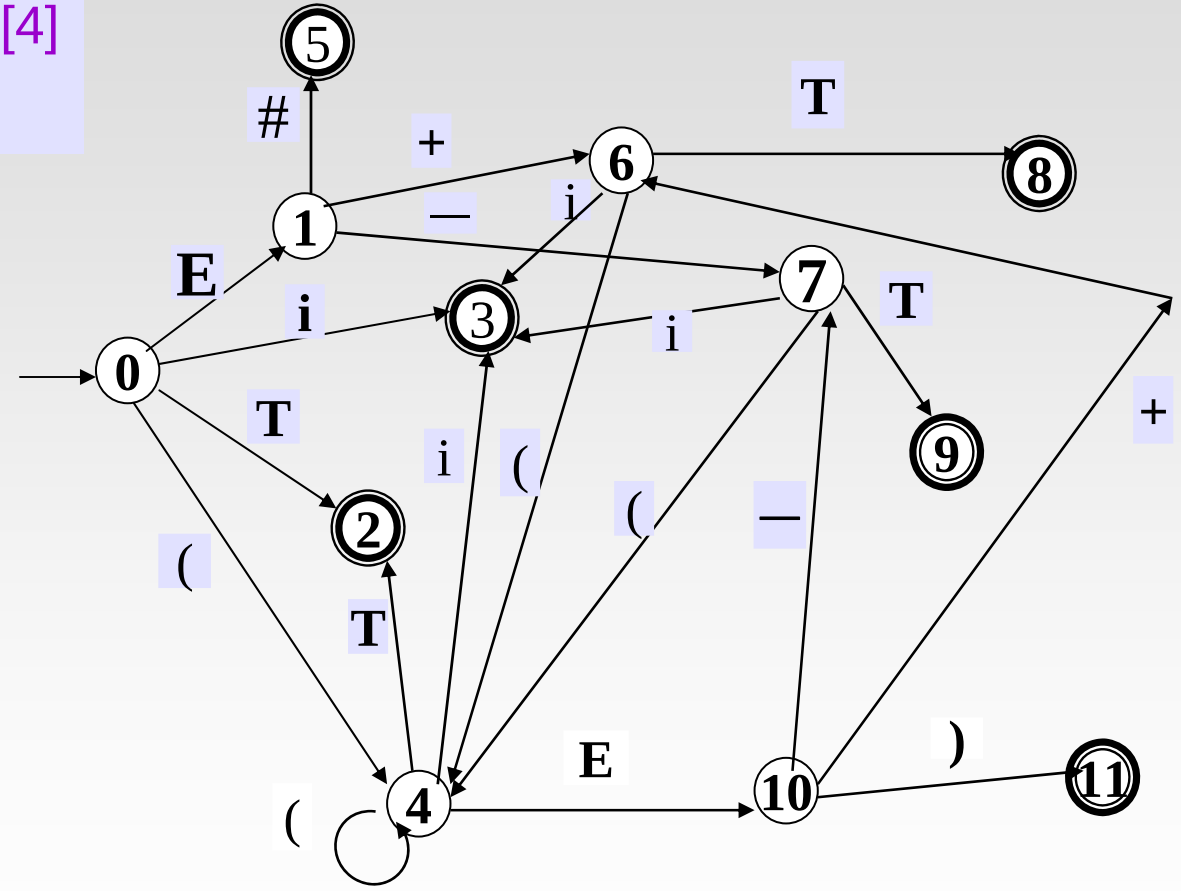
G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]

非终态:识别到活前缀

终态: 识别到可归前缀



识别活前缀和可归前缀的FA

4.2.4 LR分析法　　❑LR(0)分析法

1、LR(0) 分析器

❑总控程序

❑DFA M (状态转换矩阵，LR(0)分析表)

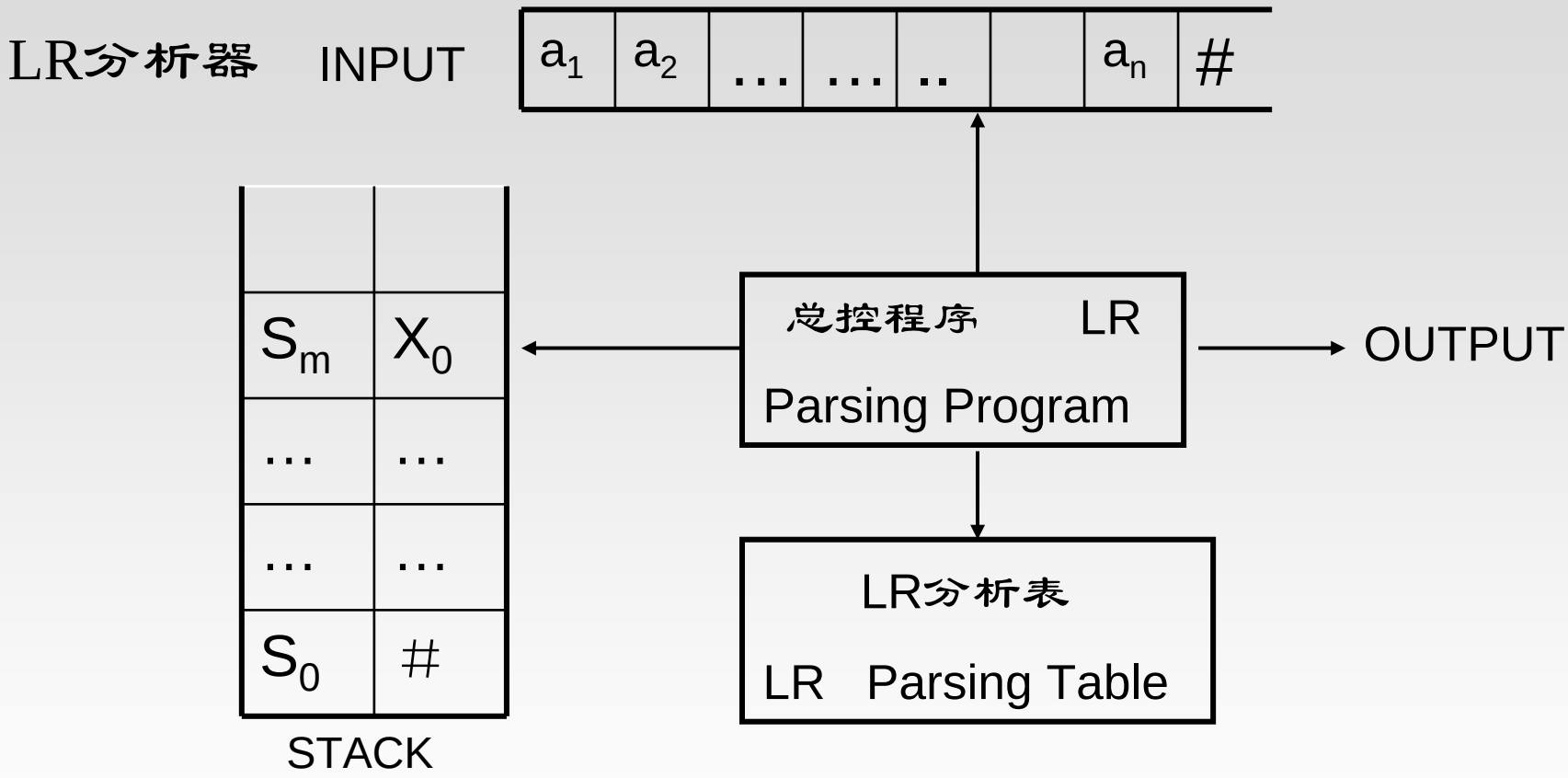
0状态：开始状态

读状态：非终态，识别到活前缀

归约状态：终态，识别到可归前缀

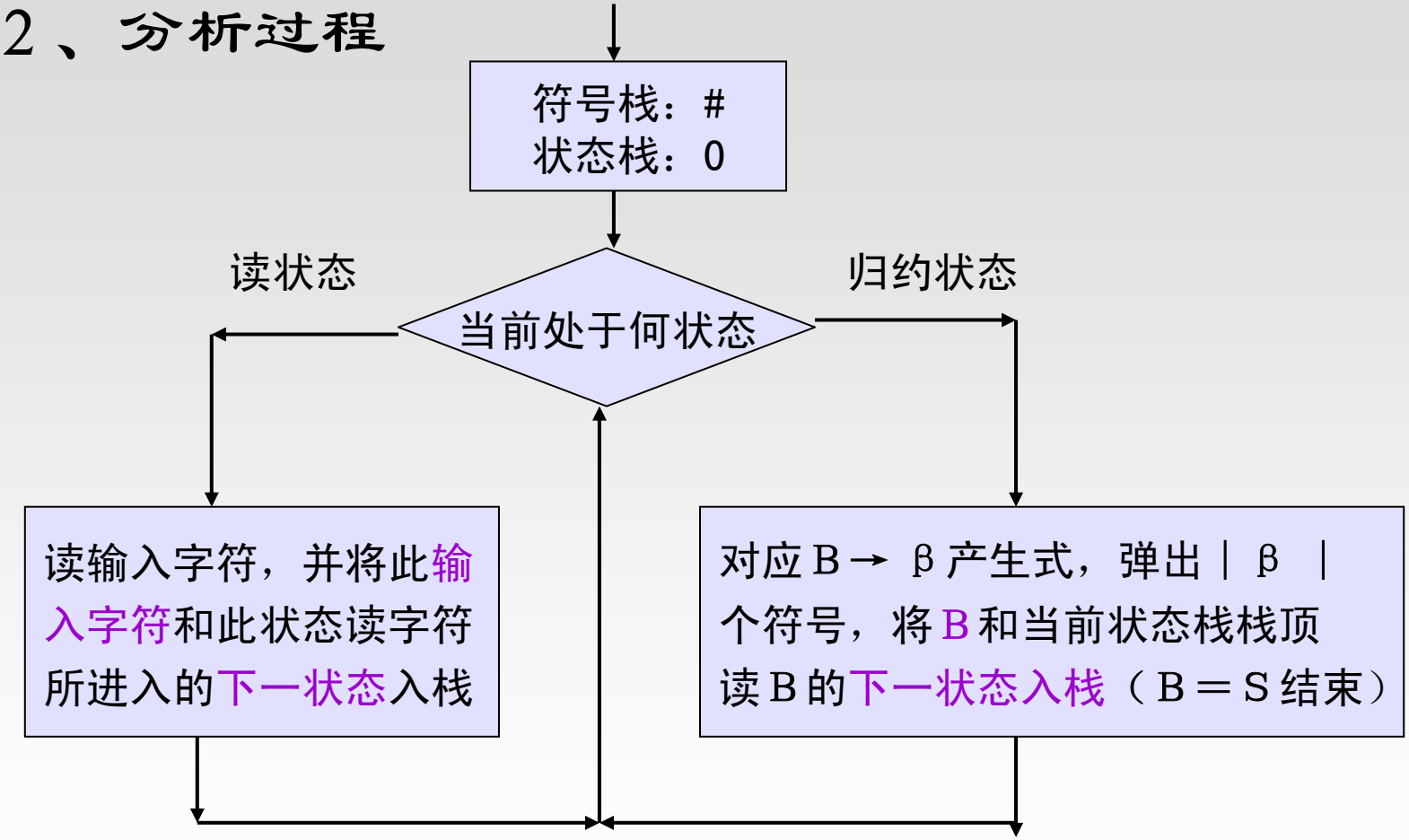
❑对偶栈
- 符号栈：放V中的字符
- 状态栈：扫描V上的字符后进入的状态(DFA)

4.2.4 LR分析法 ❑LR(0)分析法



4.2.4 LR分析法　　❑LR(0)分析法

2、分析过程



4.2.4 LR分析法 ❑LR(0)分析法

#		
0		

#	i	
0	3	

#	T	
0	2	

#	E	
0	1	

#	E	–	
0	1	7	

i

–

规约i 规则[4]

规约T 规则[5]

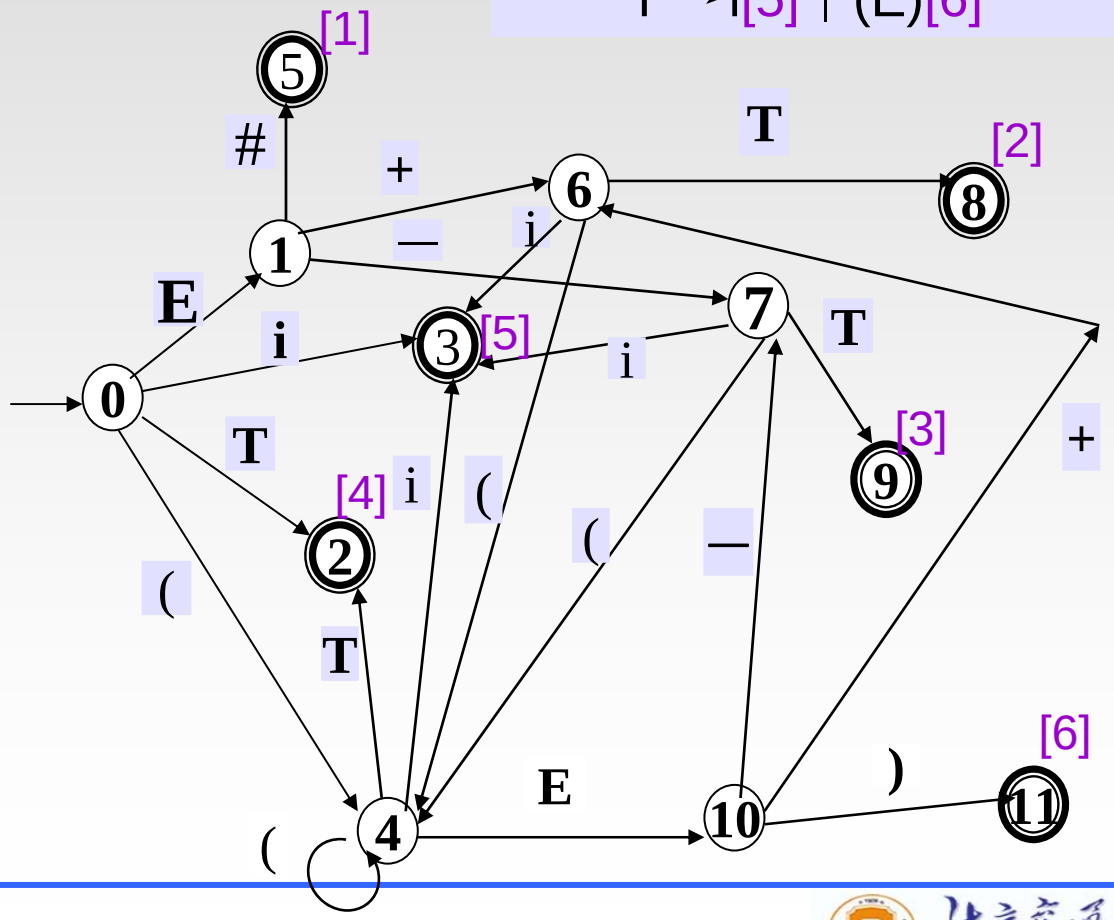
(

输入串i-(i+i)#

G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]



4.2.4 LR分析法　　❑LR(0)分析法

#	E	–	(	
0	1	7	4	

i

#	E	–	(	i	
0	1	7	4	3	

+

规约i

#	E	–	(	T	
0	1	7	4	2	

规约T

#	E	–	(	E	+
0	1	7	4	10	6

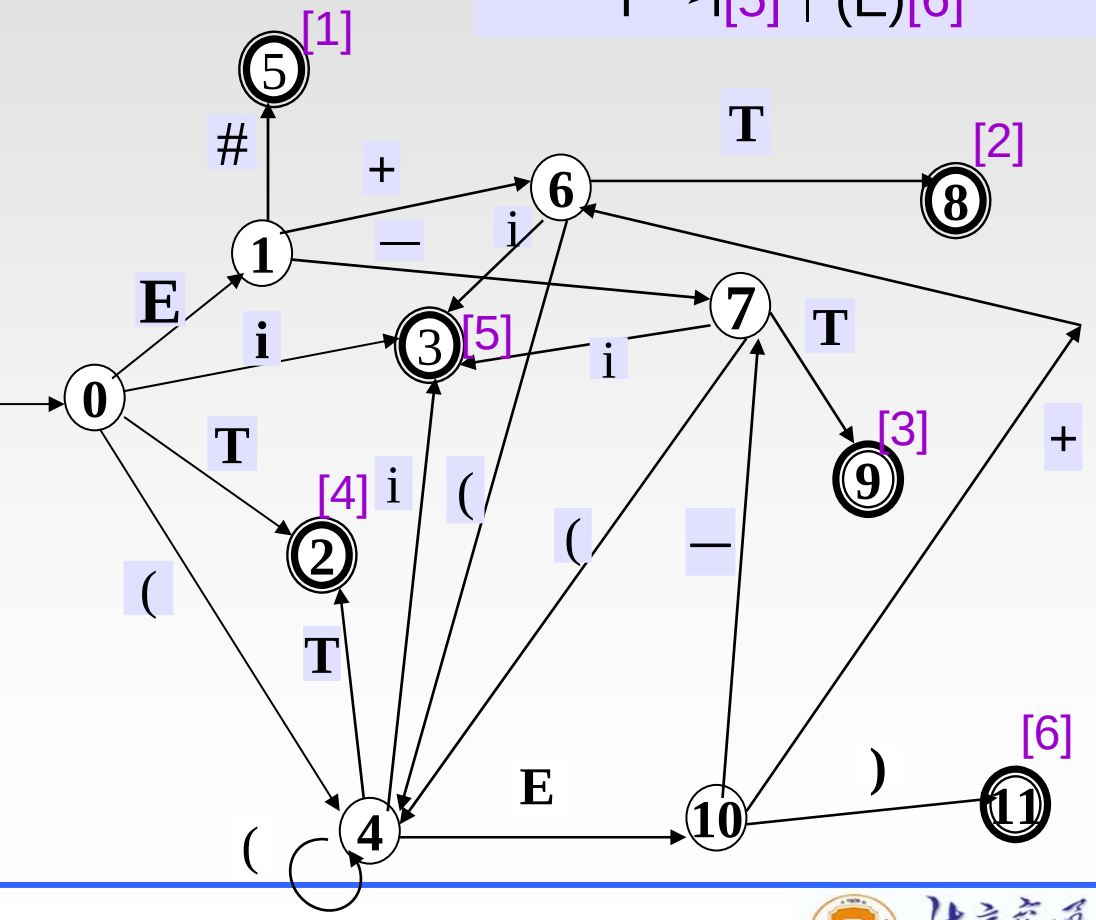
i

输入串i-(i+i)#

G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]



4.2.4 LR分析法　　❑LR(0)分析法

#	E	–	(	E	+	i	
0	1	7	4	**10**	6	3	

#	E	–	(	E	+	T	
0	1	7	4	**10**	6	8	

#	E	–	(	E	
0	1	7	4	10	

#	E	–	(	E	)	
0	1	7	4	**10**	11	

#

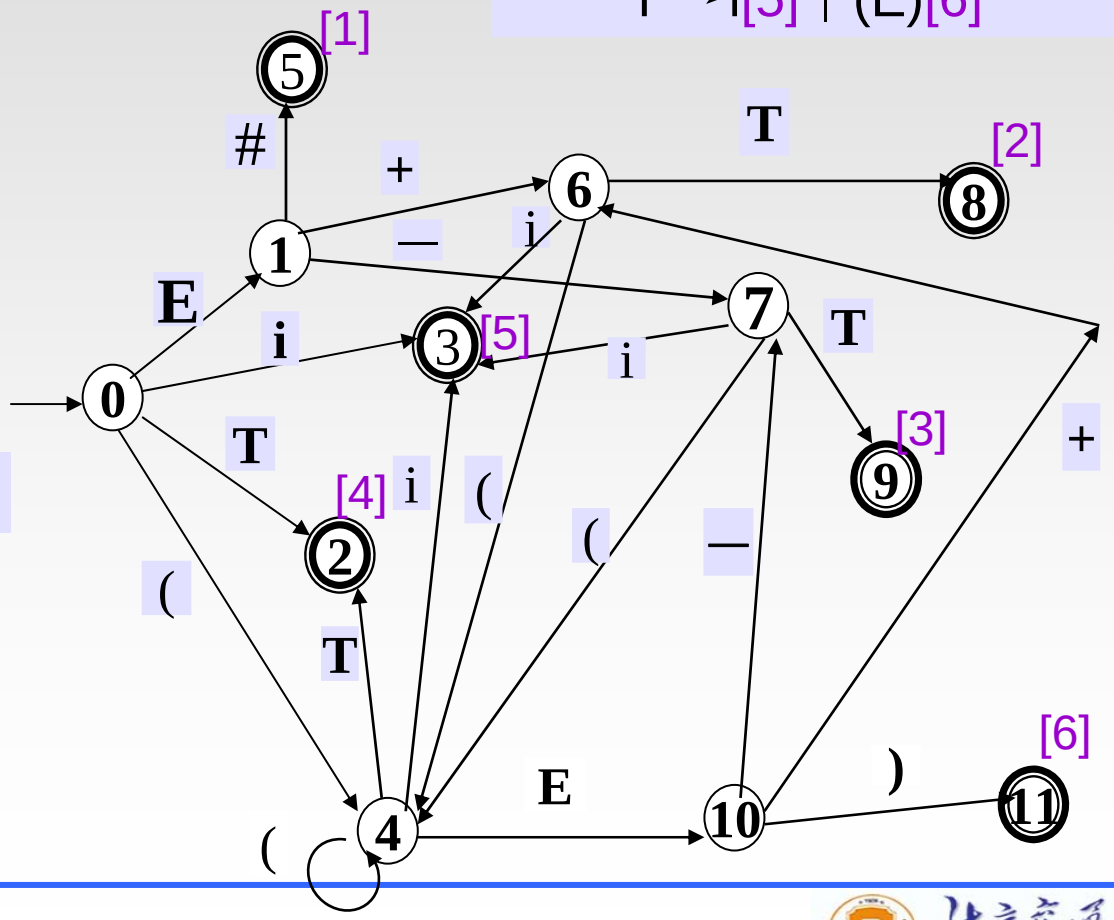
输入串i-(i+i)#

)

G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]



4.2.4 LR分析法　　❑LR(0)分析法

#	E	–	(	E	)	
0	1	7	4	**10**	11	

#	E	–	T	
0	1	7	9	

#	E	#	
0	1	5	

#	S		
0			

接受

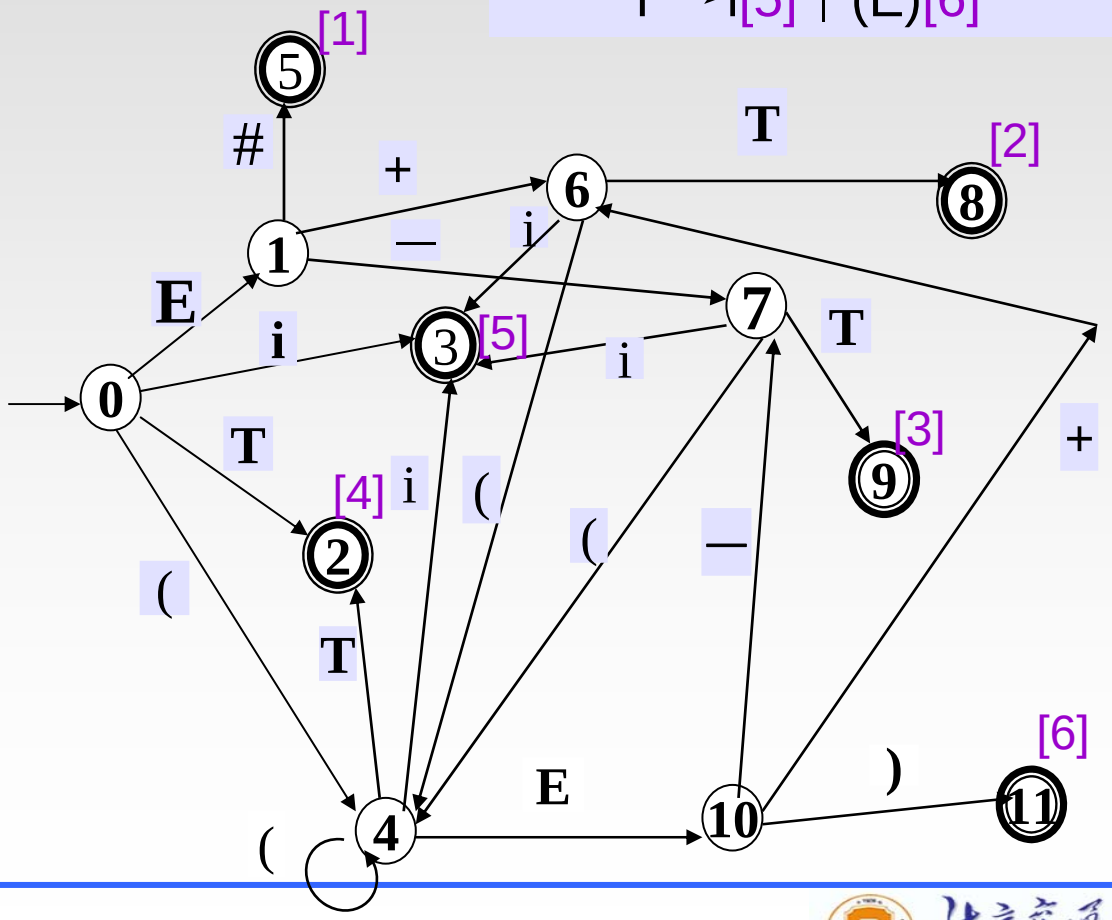
#

输入串i-(i+i)#

G[S]:S →E#[1]

E→E+T[2]｜E-T[3]｜T[4]

T →i[5]｜(E)[6]



4.2.4 LR分析法　　❑LR(0)分析法

讨论

❑算符优先分析与LR分析法比较

相同点：通过分析栈的栈顶项和当前输入符号找当前句型句柄的右端；

不同点：优先分析法为找句柄的头必须对栈进行搜索；

LR分析法只根据栈顶状态和当前输入符号就可判断；

❑问题：LR(0)分析器如何转换成LR(0)分析表？

LR(0)分析器如何构造？

4.2.4 LR分析法

总结

- 基本概念：活前缀、可归前缀
  - 规范句型的活前缀可以为FA M所识别
- LR(0)分析法
  - LR(0)分析器
  - LR(0)分析方法
- 进一步要解决的问题
  - 由LR(0)分析器构造LR(0)分析表
  - 构造LR(0)分析器

4.2.4 LR分析法　　❑LR(0)分析法

3、 LR(0)分析表:将DFA的信息放入一张表中

ACTION[S,a]函数:状态S面临输入符号a时应采取的动作

S_j:移进,把下一状态j和现输入符号a移入栈

R_j:归约,按第j产生式归约

acc:接受

空白:出错

GOTO[S,x]函数:状态S面临文法符号x($x \in V$)时下一状态.

文法G的LR(0)分析表

状态	ACTION						GOTO								
	i	+	—	(	)	#	S	E	T	i	+	—	(	)	#
0	S_3			S_4				1	2						
1		S_6	S_7			S_5									
2	r_4	r_4	r_4	r_4	r_4	r_4									
3	r_6	r_6	r_6	r_6	r_6	r_6									
4	S_3			S_4				10	2	3			4		
5						ac									
6	S_3			S_4					8	3			4		
7	S_3			S_4					9						
8	r_2	r_2	r_2	r_2	r_2	r_2									
9	r_3	r_3	r_3	r_3	r_3	r_3									
10		S_6	S_7		S_{11}						6	7		11	
11	r_5	r_5	r_5	r_5	r_5	r_5									

问题：

如何体现是LR(0)分析？

4、LR(0)分析器的构造 --DFA构造

1)项目:给定文法的一个项目是一个在右部符号串中标有一圆点的产生式,

形式: $A\rightarrow \alpha_1 \cdot \alpha_2$

$A\rightarrow \alpha_1 \alpha_2$ 为一个产生式

表示:已从输入串中看到了能由 α_1 推导出的符号串,希望进一步看到由 α_2 推导出的符号串.

例:$E \rightarrow E+T$ 项目: $E \rightarrow \cdot E+T$ $E \rightarrow E \cdot +T$

$E \rightarrow E+ \cdot T$ $E \rightarrow E+T \cdot$

4.2.4 LR分析法　　❑LR(0)分析法

- 归约项目:圆点在最后的项目

$E \rightarrow E+T \cdot$

- 接受项目:开始符号的归约项目

$S \rightarrow E\# \cdot$

- 移进项目:形如$A \rightarrow \alpha \cdot a \beta$项目 $a \in V_t$

$E \rightarrow E \cdot +T$

- 待约项目:形如$A \rightarrow \alpha \cdot B \beta$项目 $B \in V_n$

$E \rightarrow E + \cdot T$

4.2.4 LR分析法　　❑LR(0)分析法

2) 有效项目:

$A \rightarrow \alpha_1 \cdot \alpha_2$是项目,对于某一个活前缀 $\varphi\ \alpha_1$

存在$S \Rightarrow^* \varphi At \Rightarrow \varphi\ \alpha_1\ \alpha_2 t$　　$t \in V_t^*$

则称$A \rightarrow \alpha_1 \cdot \alpha_2$是活前缀$\varphi\ \alpha_1$的有效项目.

- 若归约项目$A \rightarrow \beta_1 \cdot$对活前缀$\alpha\ \beta_1$是有效的,应把$\beta_1$归约为A.
- 若待约或移入项目$A \rightarrow \beta_1 \cdot \beta_2$对活前缀$\alpha\ \beta_1$是有效的则句柄尚未形成,下一步动作是移进或待约.