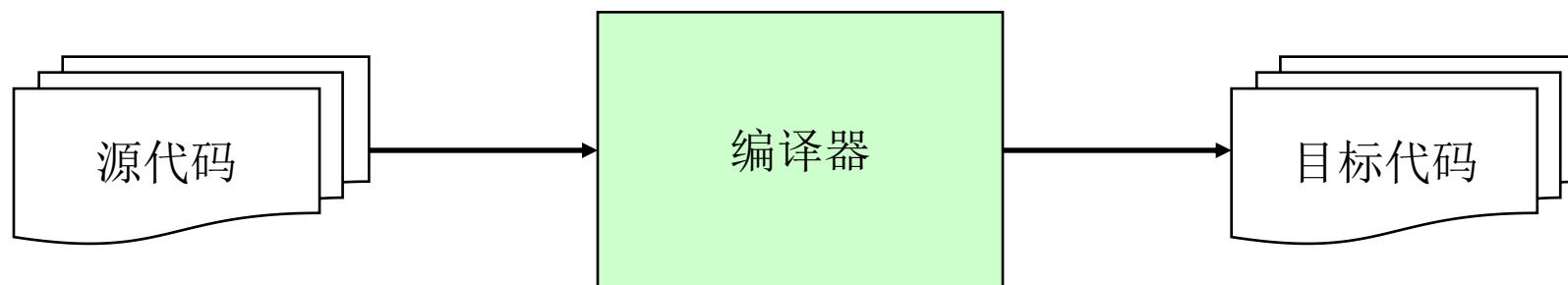
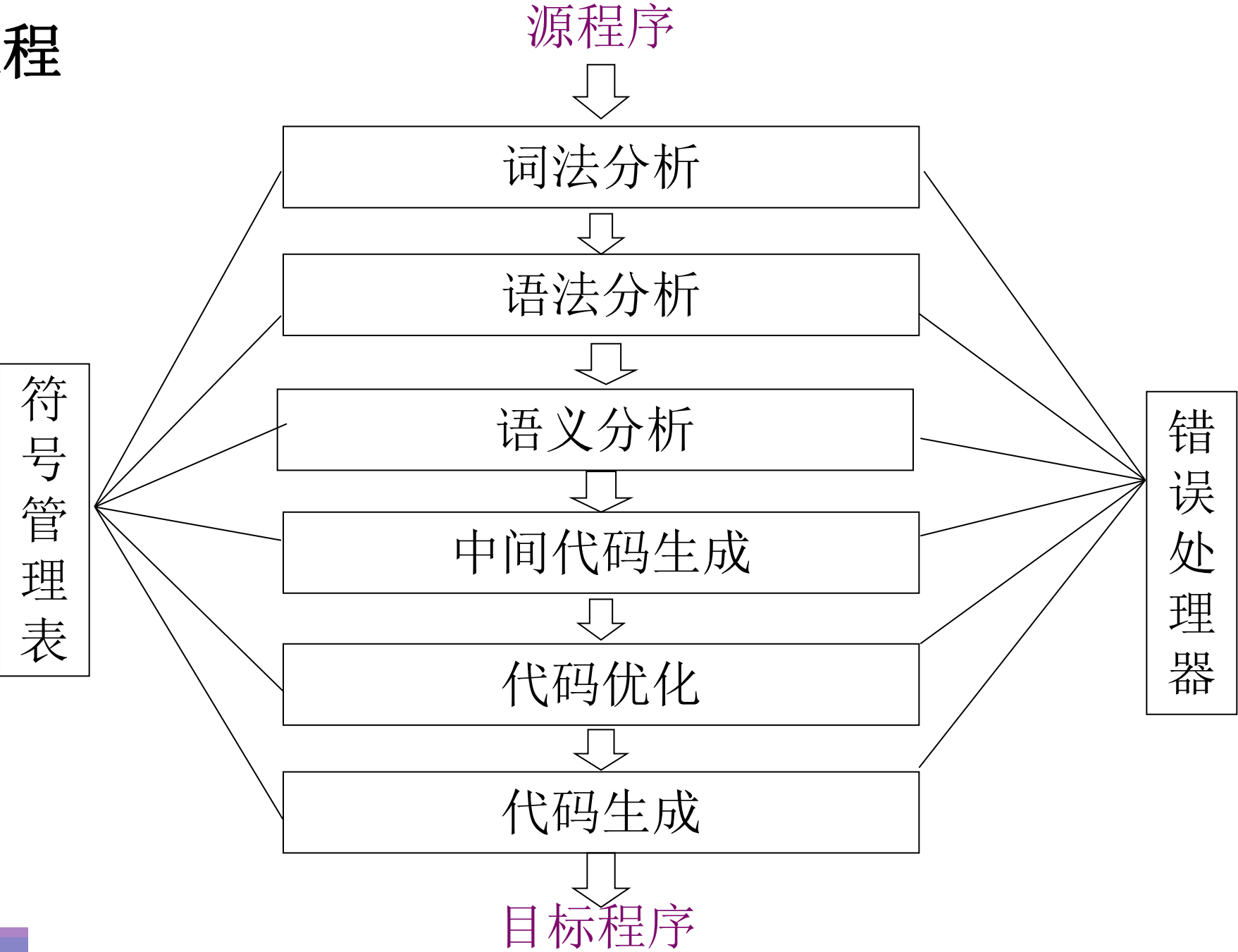


课程总结

编译器概述



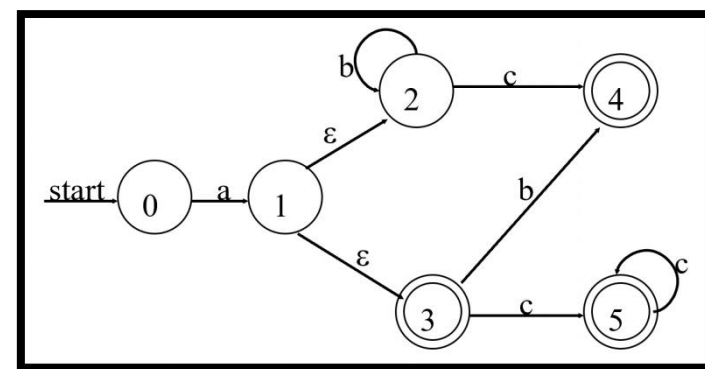
编译器的构造过程



词法分析

主要内容：

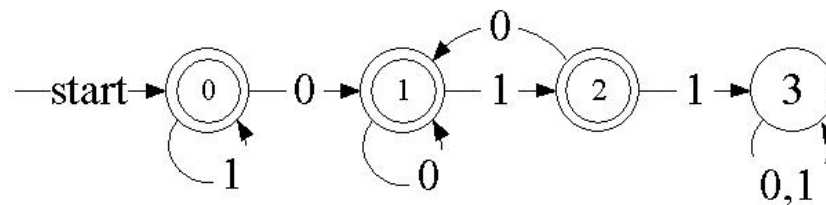
- 正则表达式：描述词素模式的重要表示方法
- 词法分析器的构造方法
 - 正则表达式 → 构造NFA（Thompson构造法）
 - NFA → 转换DFA（子集构造法）
 - 最小化DFA状态



语法分析

- 任务
- 上下文无关文法（描述语言的语法结构的形式规则）
- 语法分析器的类型
 1. 自顶向下分析器（递归下降分析，**LL(1)**）
 2. 自底向上分析器（LR(K)分析法：**LR(0)**，**SLR**，LR(1)，LALR)

设计CFG



- $L_1 = \{ wcw^R \mid w \in (a \mid b)^*, w^R \text{ 为 } w \text{ 的反转} \}$

$S \rightarrow aSa \mid bSb \mid c$

- $L_2 = \{ a^n b^m c^m d^n \mid n \geq 1 \text{ 且 } m \geq 1 \}$

$S \rightarrow aSd \mid aAd \quad A \rightarrow bAc \mid bc$

$L_2'' = \{ a^n b^n c^m d^m \mid n \geq 1 \text{ 且 } m \geq 1 \}$

$S \rightarrow AB \quad A \rightarrow aAb \mid ab \quad B \rightarrow cBd \mid cd$

- $L_3 = \{ a^n b^n \mid n \geq 0 \}$

$S \rightarrow aSb \mid ab$

$S \rightarrow \underline{a}A\underline{B}e$
 $A \rightarrow A\underline{b} \mid \underline{b}$
 $\underline{B} \rightarrow \underline{d}$

- $L_4 = \{ \text{不包含子串 } 011 \text{ 的 } 0/1 \text{ 串} \}$

$S \rightarrow 0A \mid 1S \mid \varepsilon$

$A \rightarrow 0A \mid 1B \mid \varepsilon$

$B \rightarrow 0A \mid \varepsilon$

- 二义性

- ε -moves

- 回路

- 左递归

- 左公因子

$S \Rightarrow_{rm} aABe \Rightarrow_{rm} aAde \Rightarrow_{rm} aAbde \Rightarrow_{rm} abbde$

LL(1)

输入：CFG G

输出：预测分析表M

Non-terminal	INPUT SYMBOL					
	a	b	e	i	t	\$
S						
S'						
E						

方法：

1. 对每个产生式 $A \rightarrow \alpha$ ，重复做2、3
2. 对所有的终结符 $a \in \text{FIRST}(\alpha)$ ，将 $A \rightarrow \alpha$ 加入 $M[A, a]$
3. 若 $\epsilon \in \text{FIRST}(\alpha)$ ：对所有终结符 $b \in \text{FOLLOW}(A)$ ，将 $A \rightarrow \alpha$ 加入 $M[A, b]$ ；
若 $\$ \in \text{FOLLOW}(A)$ ，将 $A \rightarrow \alpha$ 加入 $M[A, \$]$
4. 所有未定义的表项设置为错误

LL(1)

$S \rightarrow i E t S S' \mid a$	$\text{First}(S) = \{ i, a \}$	$\text{Follow}(S) = \{ e, \$ \}$
$S' \rightarrow eS \mid \epsilon$	$\text{First}(S') = \{ e, \epsilon \}$	$\text{Follow}(S') = \{ e, \$ \}$
$E \rightarrow b$	$\text{First}(E) = \{ b \}$	$\text{Follow}(E) = \{ t \}$

$S \rightarrow i E t S S'$

$S \rightarrow a$

$E \rightarrow b$

$\text{First}(i E t S S') = \{ i \}$

$\text{First}(a) = \{ a \}$

$\text{First}(b) = \{ b \}$

$S' \rightarrow eS$

$S' \rightarrow \epsilon$

$\text{First}(eS) = \{ e \}$

$\text{First}(\epsilon) = \{ \epsilon \}$

$\text{Follow}(S') = \{ e, \$ \}$

Non-terminal	INPUT SYMBOL					
	a	b	e	i	t	\$
S	<u>$S \rightarrow a$</u>			<u>$S \rightarrow iEtSS'$</u>		
S'			<u>$S' \rightarrow \epsilon$</u> <u>$S' \rightarrow eS$</u>			<u>$S' \rightarrow \epsilon$</u>
E		<u>$E \rightarrow b$</u>				

LL(1)

STACK	INPUT	OUTPUT
\$E	id + id * id\$	E → TE'
\$E'T	id + id * id\$	T → FT'
\$E'T'F	id + id * id\$	F → id
\$E'T'id	id + id * id\$	
\$E'T'	+ id * id\$	T' → ε
\$E'	+ id * id\$	E' → +TE'
\$E'T+	+ id * id\$	
\$E'T	id * id\$	T → FT'
\$E'T'F	id * id\$	F → id
\$E'T'id	id * id\$	
\$E'T'	* id\$	T' → *FT'
\$E'T'F*	* id\$	
\$E'T'F	id\$	F → id
\$E'T'id	id\$	
\$E'T'	\$	T' → ε
\$E'	\$	E' → ε
\$	\$	

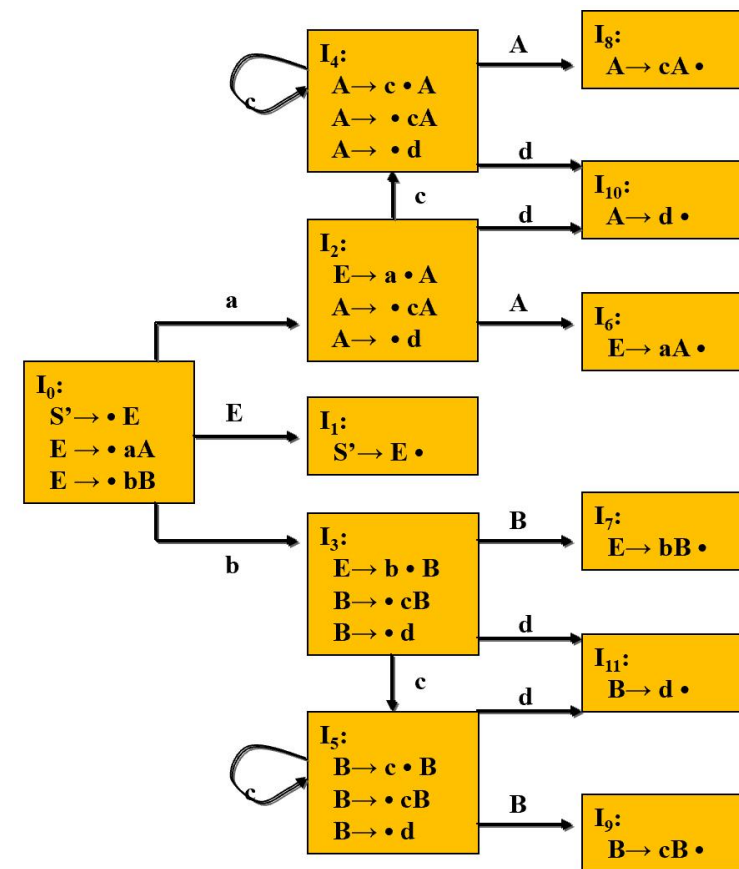
Non-terminal	INPUT SYMBOL					
	id	+	*	()	\$
E	E→TE'			E→TE'		
E'		E'→+TE'			E'→ε	E'→ε
T	T→FT'			T→FT'		
T'		T'→ε	T'→*FT'		T'→ε	T'→ε
F	F→id			F→(E)		

LR(0)

LR(0) 项目

$S' \rightarrow E \cdot$ 接受项目 $E \rightarrow aA \cdot$ 归约项目 $E \rightarrow \cdot aA$ 移进项目

- 0) $S' \rightarrow E$
- 1) $E \rightarrow aA$
- 2) $E \rightarrow bB$
- 3) $A \rightarrow cA$
- 4) $A \rightarrow d$
- 5) $B \rightarrow cB$
- 6) $B \rightarrow d$



SLR(1)

① 将文法 G 增广为 G' ，同时对每一产生式进行编号

- (0) $S' \rightarrow E$ (4) $T \rightarrow F$
(1) $E \rightarrow E + T$ (5) $F \rightarrow (E)$
(2) $E \rightarrow T$ (6) $F \rightarrow id$
(3) $T \rightarrow T * F$

② 对 G' 构造文法 LR (0) 项目集规范族如下：

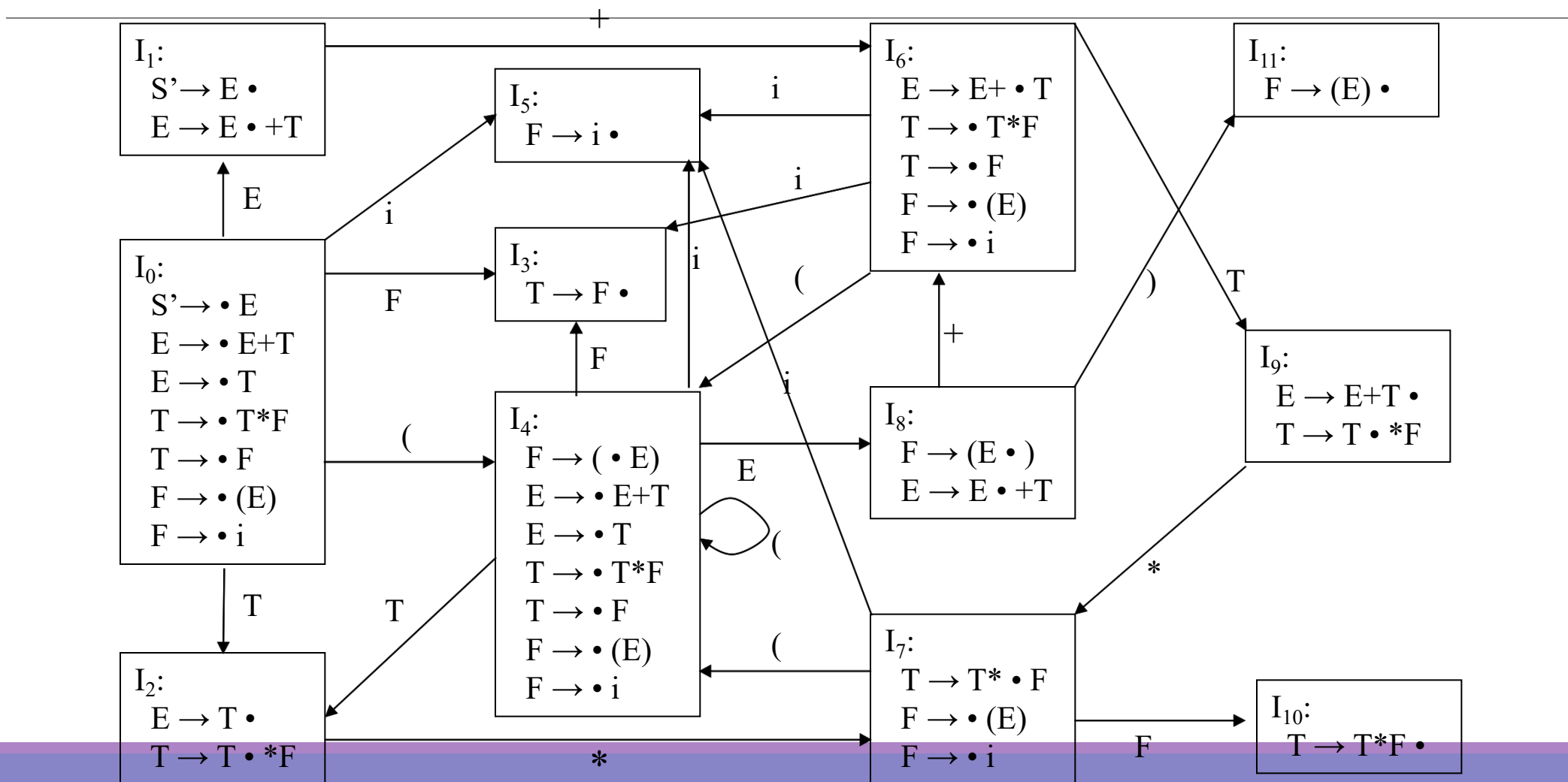
I_0 : $S' \rightarrow \cdot E$
 $E \rightarrow \cdot E + T$
 $E \rightarrow \cdot T$
 $T \rightarrow \cdot T * F$
 $T \rightarrow \cdot F$
 $F \rightarrow \cdot (E)$
 $F \rightarrow \cdot id$
 I_1 : $S' \rightarrow E \cdot$
 $E \rightarrow E \cdot + T$

I_2 : $E \rightarrow T \cdot$
 $T \rightarrow T \cdot * F$
 I_3 : $T \rightarrow F \cdot$
 I_4 : $F \rightarrow (\cdot E)$
 $E \rightarrow \cdot E + T$
 $E \rightarrow \cdot T$
 $T \rightarrow \cdot T * F$
 $T \rightarrow \cdot F$
 $F \rightarrow \cdot (E)$
 $F \rightarrow \cdot id$

I_5 : $F \rightarrow id \cdot$
 I_6 : $E \rightarrow E + \cdot T$
 $T \rightarrow \cdot T * F$
 $T \rightarrow \cdot F$
 $F \rightarrow \cdot (E)$
 $F \rightarrow \cdot id$
 I_7 : $T \rightarrow T * \cdot F$
 $F \rightarrow \cdot (E)$
 $F \rightarrow \cdot id$

I_8 : $F \rightarrow (E \cdot)$
 $E \rightarrow E \cdot + T$
 I_9 : $E \rightarrow E + T \cdot$
 $T \rightarrow T \cdot * F$
 I_{10} : $T \rightarrow T * F \cdot$
 I_{11} : $F \rightarrow (E) \cdot$

③ 取这些项目集作为各状态，并根据转换函数G0画出识别文法G' 的有穷自动机，



⑤ 构造SLR(1)分析表

状态	ACTION (动作)						GOTO (状态转换)		
	i	+	*	()	\$	E	T	F
0	S ₅			S ₄			1	2	3
1		S ₆				acc			
2		r ₂	S ₇		r ₂	r ₂			
3		r ₄	r ₄		r ₄	r ₄			
4	S ₅			S ₄			8	2	3
5		r ₆	r ₆		r ₆	r ₆			
6	S ₅			S ₄				9	3
7	S ₅			S ₄					10
8		S ₆			S ₁₁				
9		r ₁	S ₇		r ₁	r ₁			
10		r ₃	r ₃		r ₃	r ₃			
11		r ₅	r ₅		r ₅	r ₅			

输入串为id+id*id为例，分析过程：

步骤	状态栈	符号栈	输入串	分析动作	下一状态
1	0	\$	id+id*id\$	S ₅	5
2	05	\$ id	+id*id\$	r ₆	GOTO[0, F]=3
3	03	\$ F	+id*id\$	r ₄	GOTO[0, T]=2
4	02	\$ T	+id*id\$	r ₂	GOTO[0, E]=1
5	01	\$ E	+id*id\$	S ₆	6
6	016	\$ E+	id*id\$	S ₅	5
7	0165	\$ E+id	*id\$	r ₆	GOTO[6, F]=3
8	0163	\$ E+F	*id\$	r ₄	GOTO[6, T]=9
9	0169	\$ E+T	*id\$	S ₇	7
10	01657	\$ E+T*	id\$	S ₅	5
11	016575	\$ E+T*id	\$	r ₆	GOTO[7, F]=10
12	01657 <u>10</u>	\$ E+T*F	\$	r ₃	GOTO[6, T]=9
13	0169	\$ E+T	\$	r ₁	GOTO[0, E]=1
14	01	\$ E	\$	acc	

状态	ACTION (动作)						GOTO (状态转换)		
	id	+	*	()	\$	E	T	F
0	S ₅			S ₄			1	2	3
1		S ₆				acc			
2		r ₂	S ₇		r ₂	r ₂			
3		r ₄	r ₄		r ₄	r ₄			
4	S ₅			S ₄			8	2	3
5		r ₆	r ₆		r ₆	r ₆			
6	S ₅			S ₄				9	3
7	S ₅			S ₄					10
8		S ₆			S ₁₁				
9		r ₁	S ₇		r ₁	r ₁			
10		r ₃	r ₃		r ₃	r ₃			
11		r ₅	r ₅		r ₅	r ₅			

语法制导翻译

产生式	语义规则
$E \rightarrow E_1 + T$	$E.nptr = mknode("+", E_1.nptr, T.nptr)$
$E \rightarrow E_1 - T$	$E.nptr = mknode("-", E_1.nptr, T.nptr)$
$E \rightarrow T$	$E.nptr = T.nptr$
$T \rightarrow (E)$	$T.nptr = E.nptr$
$T \rightarrow id$	$T.nptr = mkleaf(id, id.lexval)$
$T \rightarrow num$	$T.nptr = mkleaf(num, num.val)$

- 语法制导定义(SDD)：每个文法产生式有一组基于文法符号的属性的语义规则。属性：**综合属性与继承属性**
- 语法制导的翻译方案(SDT)：是SDD的实现。在文法中嵌入语义动作，当归约时，执行语义动作
- S-属性的SDT， L-属性的SDT

功能 输出

```
L → E n
E → E1 + {print('+')} T
E → T
T → T1 * {print('*')} F
T → F
F → (E)
F → digit {print(digit.lexval)}
```

中间代码生成

```
int i;  
int a[10];  
int value;  
i = 0;  
while (i<10)  
{ value = 10; a[i] = value *i; i=i+1; }  
i=0;
```

```
100: i=0  
101: if i<10 goto 104  
102: goto 109  
103: value=10  
104: t1 = value * i  
105: t2 = i*4  
106: a[t2]=t1  
107: i=i+1  
108: goto 102  
109: i=0
```

主要内容:

- 任务
- 中间表示（语法树和三地址码）
- 类型和声明
- 表达式、控制语句、布尔表达式的SDT

运行时刻环境

主要内容：

- 存储管理：静态分配、栈分配、堆管理

先进后出 先进先出

- 过程运行（调用代码序列，返回代码序列）

代码生成与优化 寄存器选择信息 活跃信息

常量合并

活跃信息 下次引用信息

公共子表达式消除

复制传播

代码移动

死代码消除

归纳变量和强度消减

符号表 + 错误处理器

记录与每个标识符相关的各种属性信息

各阶段均会遇到错误
处理方式：报告错误，继续编译

考试题型

一、单项选择题 40

二、综合题 60

谢谢助教

谢谢大家！

A solid purple horizontal bar at the bottom of the slide.