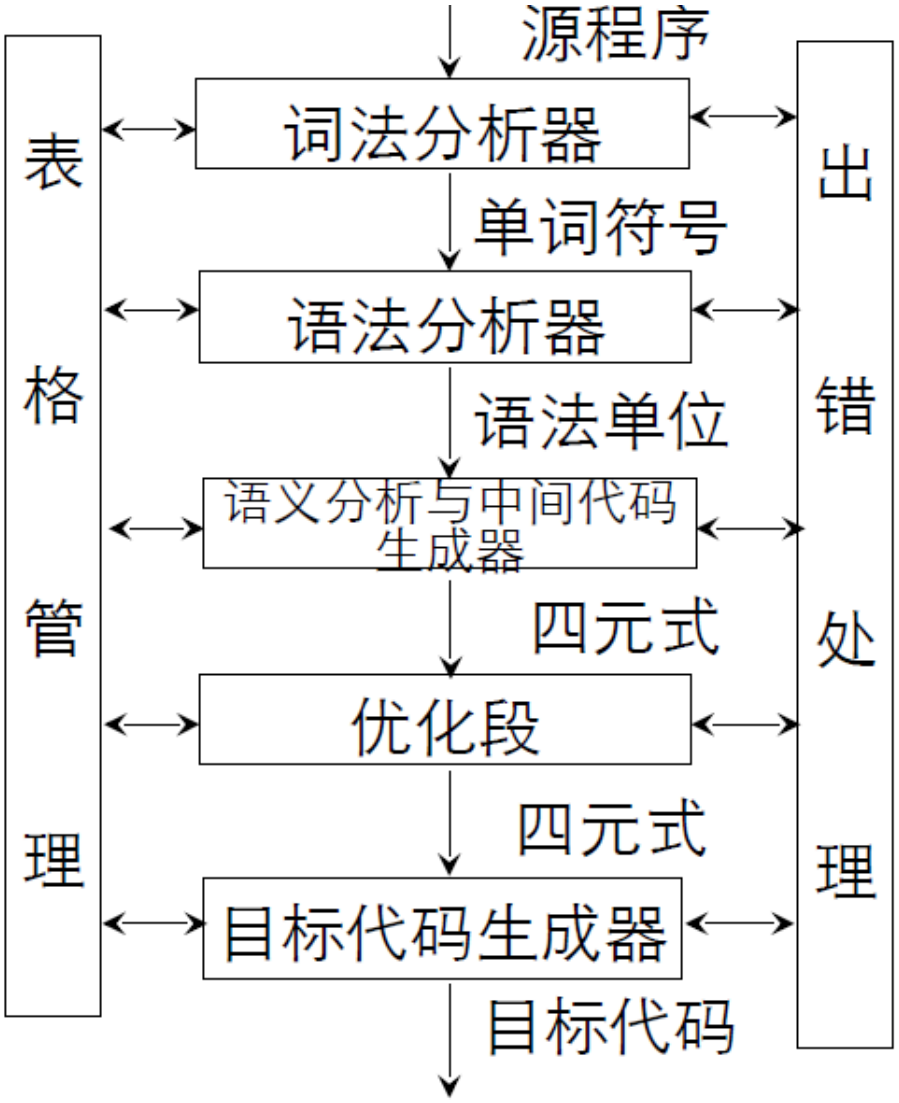
编译技术

习题课

编译程序的结构


源程序
词法分析器
单词符号
语法分析器
语法单位
语义分析与中间代码生成器
四元式
优化段
四元式
目标代码生成器
目标代码
表格管理
出错处理

核心内容

- 第一章 导论
  - 编译基本过程
  - 编译程序的结构
- 第二章 高级语言及其语法描述
  - 上下文无关文法
  - 语法树与二义性
- 第三章 词法分析
  - 正规式 ⇔ NFA
  - NFA ⇒ DFA
  - DFA化简算法
- 第四章 语法分析—自上而下分析
  - LL(1)分析法
  - 递归下降分析程序构造
  - 预测分析程序

核心内容

- 第五章 语法分析——自下而上分析I
  - 自下而上分析的基本问题
  - 算符优先分析算法
- 第六章 语法分析——自下而上分析II
  - LR分析器工作原理
  - LR(0)项目集规范族
  - LR(0)分析表的构造
  - SLR(1)分析
- 第七章 语法制导的语义计算
  - 基于属性文法的语义计算
  - 基于翻译模式的语义计算

核心内容

- 第八章 静态语义分析和中间代码生成
  - 符号表
  - 静态语义分析
  - 中间代码生成
- 第九章 运行时存储组织
  - 概述
  - 活动记录
  - 过程调用

第十章 代码优化和目标代码生成

- 基本块、流图和循环
- 数据流分析
- 代码优化技术
- 目标代码生成技术

特定语言文法

- 写一个文法，使其语言是奇数集，且每个奇数不以 0 开头

- G(S):

  $S \rightarrow O \mid A\ O$

  $O \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9$

  $N \rightarrow O \mid 2 \mid 4 \mid 6 \mid 8$

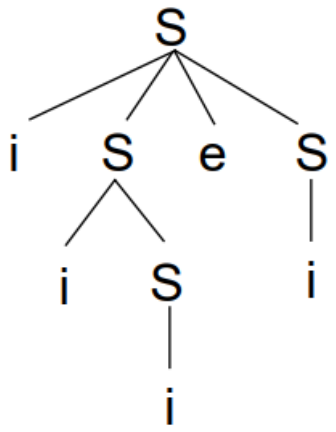
  $D \rightarrow 0 \mid N$

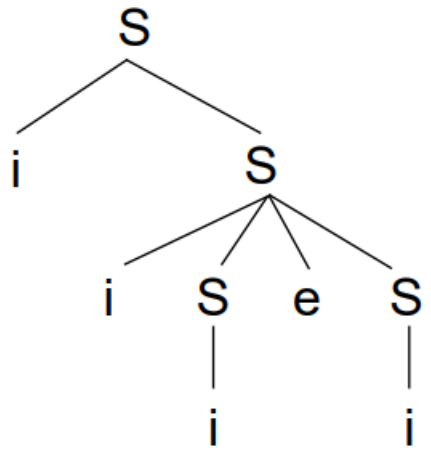
  $A \rightarrow A\ D \mid N$

文法二义性

- 定义：如果一个文法存在某个句子对应两颗不同的语法树，则说这个文法是二义的
- 证明下面的文法是二义的：

$$S \rightarrow iSeS \mid iS \mid i$$



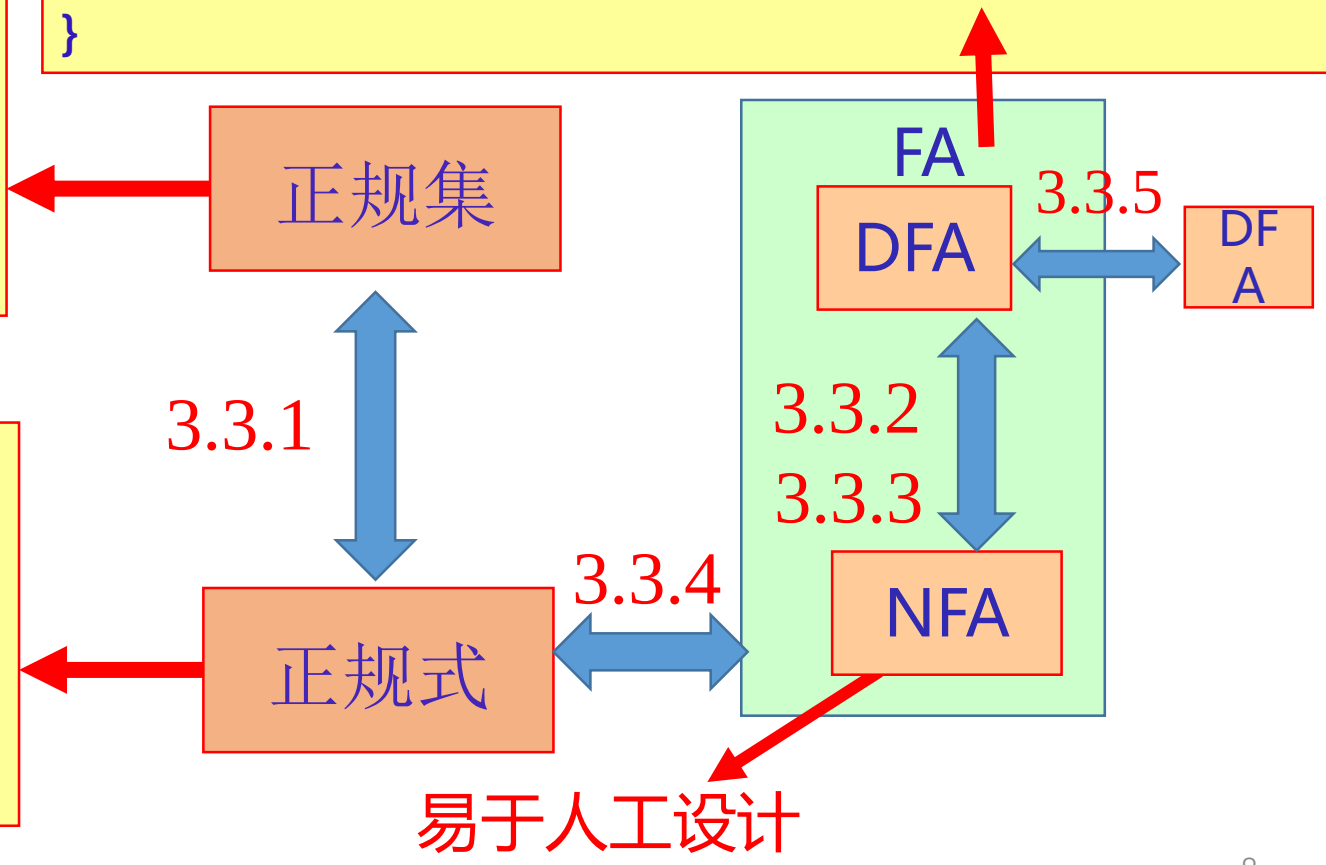


词法分析

```
curState = 初态
GetChar();
while( stateTrans[curState][ch] 有定义 ){
   // 存在后继状态，读入、拼接
   Concat();
   // 转换入下一状态，读入下一字符
   curState= stateTrans[curState][ch];
if cur_state 是终态 then 返回 strToken 中的单词
GetChar( );
}
```

DIM,IF, DO,STOP,END
number, name, age
125, 2169
…

DIM
IF
DO
STOP
END
letter(letter|digit)*
digit(digit)*

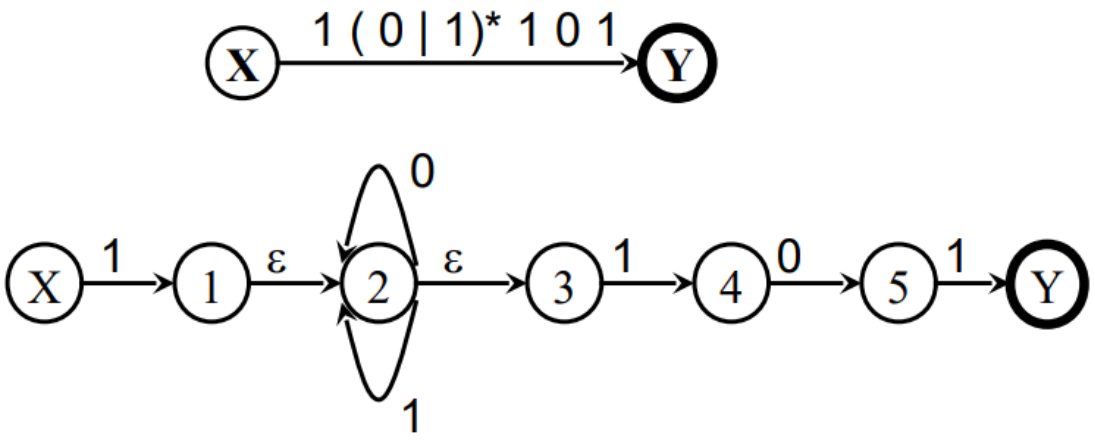


词法分析

- 构造下列正规式相应的DFA

$$1(0|1)^*101$$

思路：正规式⇒ NFA ⇒DFA



- 确定化过程： ε -closure(I)， $I_a=\varepsilon$-closure(J)
- 最小化：状态集划分

自上而下语法分析

- 考虑下面文法 $G_1(S)$ ：

$$S \rightarrow a \mid \Lambda \mid (T)$$

$$T \rightarrow T, S \mid S$$

(1) 消去G_1的左递归。

(2) 经改写后的文法是否是 LL(1)的？给出它的预测分析表。

- 思路
  - 消除左递归
  - 提取左公共因子: $T \rightarrow ST', \cdots$
  - 计算非终结符的 FIRST 、 FOLLOW和SELECT 集合
  - 检查 LL(1) 条件
  - 构造预测分析表

	a	^	(	)	,	#
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow S\ T'$	$T \rightarrow S\ T'$	$T \rightarrow S\ T'$			
T'				$T' \rightarrow \varepsilon$	$T' \rightarrow , S\ T'$	

自下而上分析

- 短语、直接短语和句柄
- 令文法G_1为：

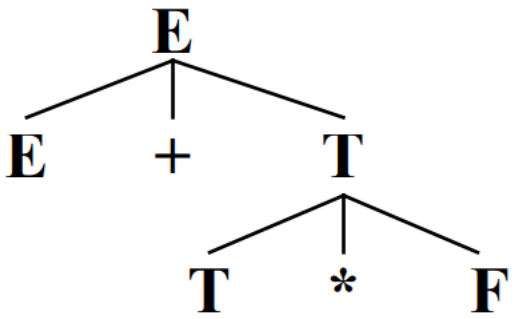
$$E \rightarrow E+T \mid T$$

$$T \rightarrow T*F \mid F$$

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

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。

- 短语： E+T*F, T*F
- 直接短语：T*F
- 句柄：T*F



自下而上分析

- 算符优先算法
- 对文法$G_2(S)$：

$$S \rightarrow a \mid \wedge \mid (T)$$

$$T \rightarrow T, S \mid S$$

(1) 计算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？

- 思路
  - 计算非终结符的FIRSTVT和LASTVT
  - 计算终结符之间的优先关系
  - 检查算符优先文法的条件

自下而上分析

- LR分析法
- 考虑文法

$$S \rightarrow AS \mid b$$

$$A \rightarrow SA \mid a$$

(1) 列出这个文法的所有LR(0)项目。

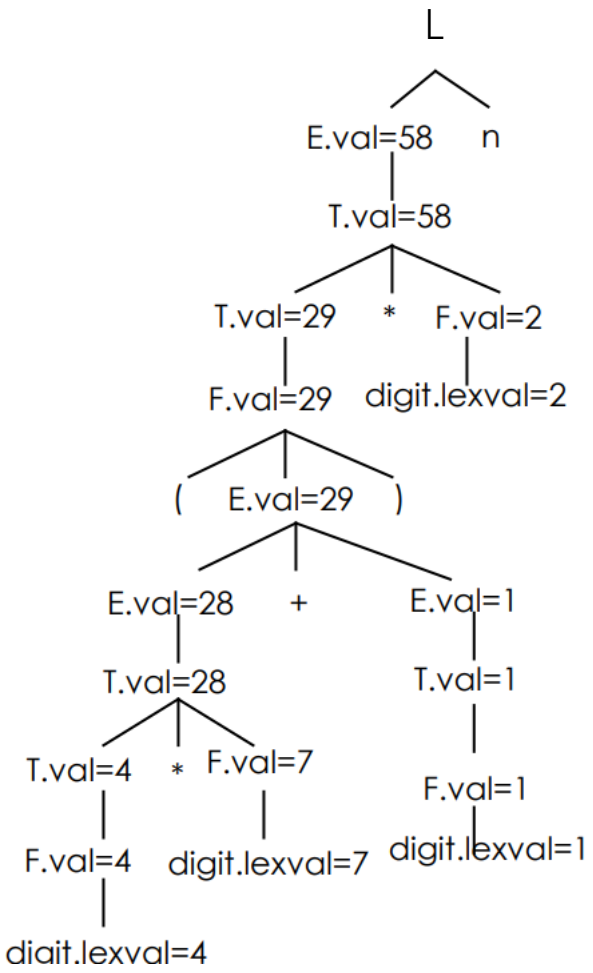
(2) 构造这个文法的LR(0)项目集规范族及识别活前缀的DFA。

- 思路：
  - 对文法进行拓广：S'→ S
  - LR(0)项目
  - NFA
  - DFA

语法制导翻译

- 按照所示的属性文法，构造表达式（4*7+1）*2的附注语法树

产生式	语义规则
L→En	print(E.val)
$E \rightarrow E_1+T$	E.val := E_1.val+T.val
E→T	E.val :=T.val
$T \rightarrow T_1*F$	T.val :=T_1.val* F.val
T→F	T.val :=F.val
F→ (E)	F.val :=E.val
F→digit	F.val :=digit.lexval



语法制导翻译与中间代码生成

- 下列文法对整型常数和实型常数施用加法运算符＋生成表达式；当两个整型数相加时，结果仍为整型数，否则，结果为实型数：

$$E \rightarrow E+T \mid T$$

$$T \rightarrow num\ .\ num \mid num$$

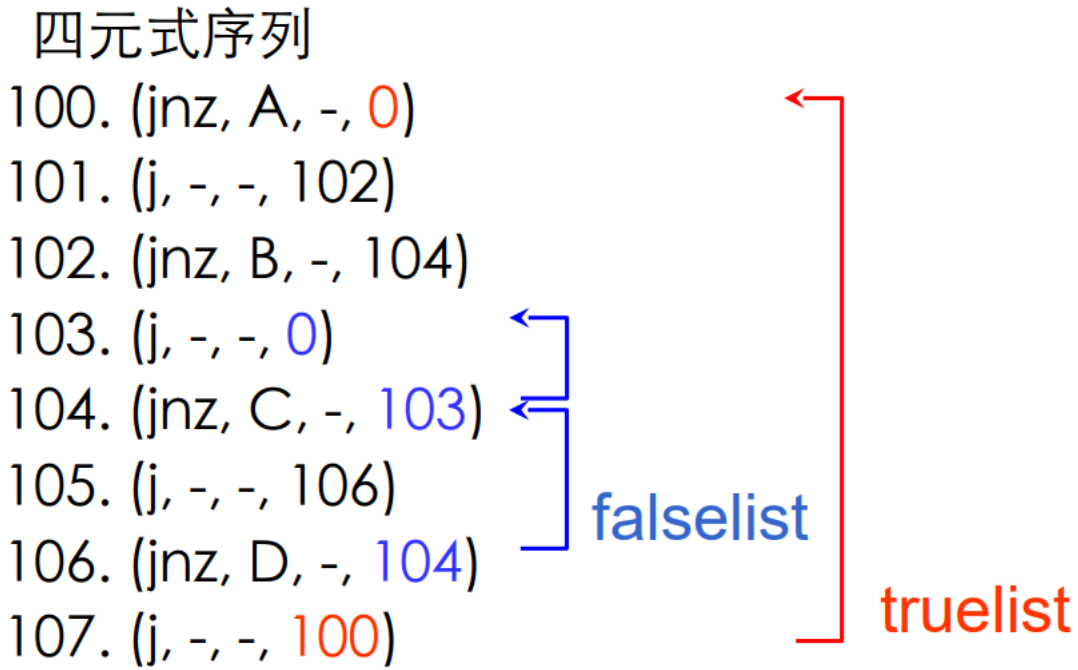
试给出确定每个子表达式结果类型的属性文法

$E \rightarrow E_1 + T$	if (E_1.type = int) and (T.type = int) then E.type := int else E.type := real
$E \rightarrow T$	E.type := T.type
$T \rightarrow num.num$	T.type := real
$T \rightarrow num$	T.type := int

- 将表达式 －(a+b)＊(c+d)－(a+b+c) 分别表示成三元式和四元式序列

中间代码生成

- 写出布尔式 A or （B and not （C or D)）的四元式序列(或者TAC)。



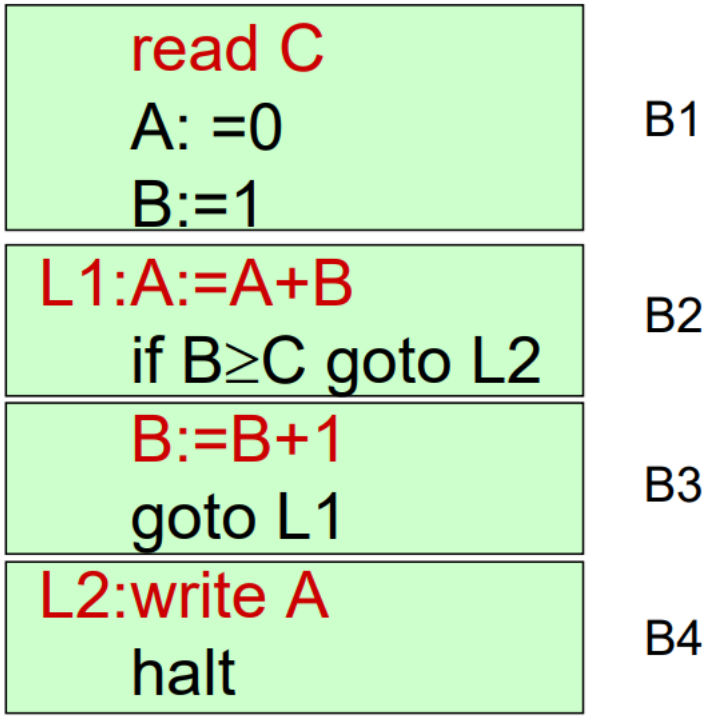
符号表与运行时存储空间

- 符号表组织与操作
- 符号表内容与作用域
- 数据空间使用和管理方法

代码优化

■把以下程序划分为基本块并作出其程序流图。

```
        read C
        A: =0
        B:=1
L1:     A:=A+B
        if B≥C goto L2
        B:=B+1
        goto L1
L2:     write A
        halt
```



代码优化

- 对以下基本块B2用DAG进行优化，并写出优化后的四元式

B2:
B:=3
D:=A+C
E:=A*C
G:=B*F
H:=A+C
I:=A*C
J:=H+I
K:=B*5
L:=K+J
M:=L