



# 编译原理课程复习

陈林

[lchen@nju.edu.cn](mailto:lchen@nju.edu.cn)





# 提纲

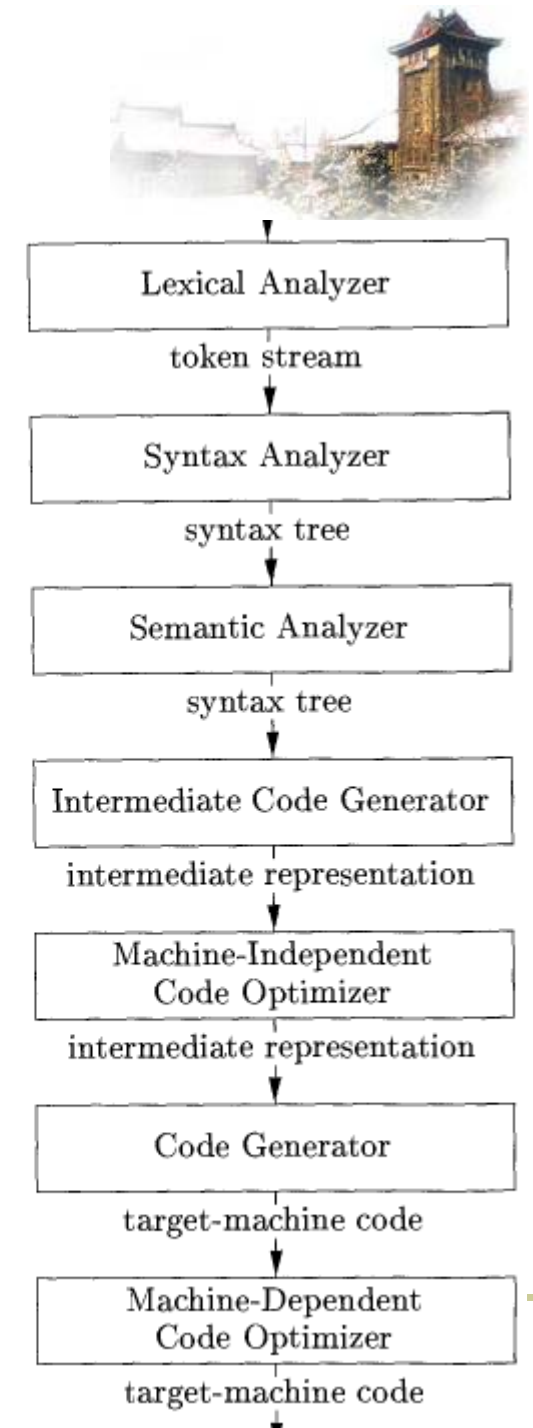
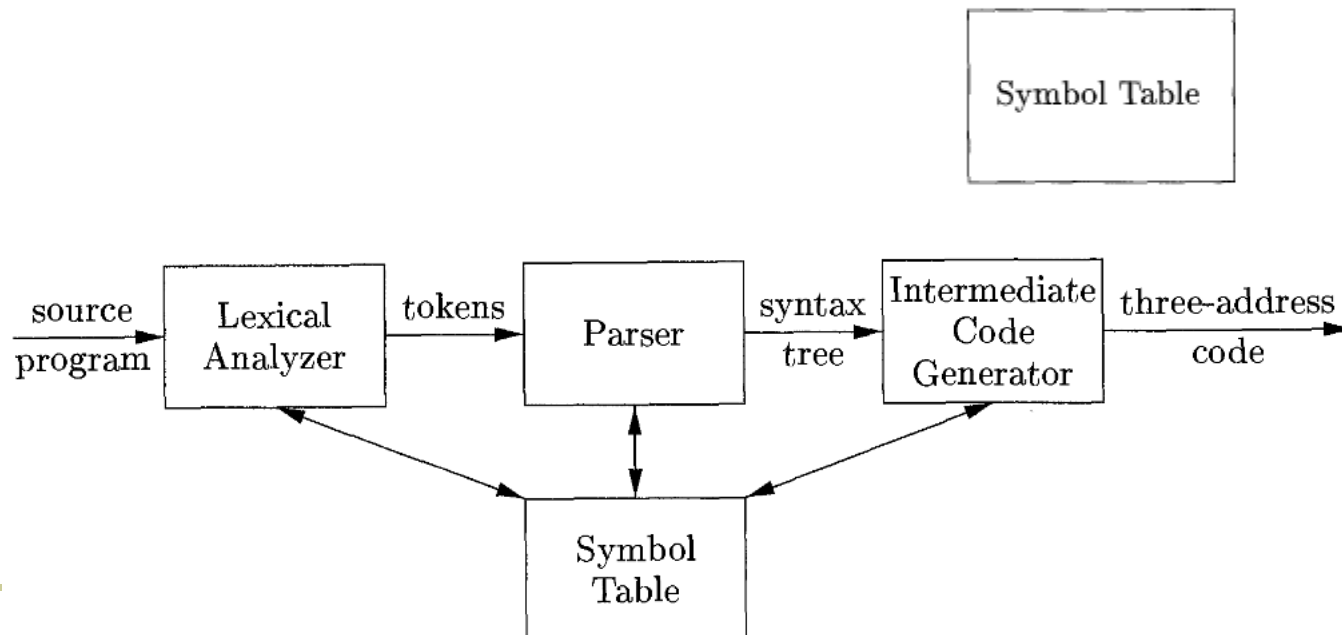


- 第1章：基本概念
- 第3章：词法分析
- 第4章：语法分析
- 第5章：语法制导翻译
- 第6章：中间代码生成
- 第7章：运行时环境
- 第8章：代码生成
- 第9章：机器无关优化



# 第一章 基本概念

- 发展简史
- 基本流程和框架
  - 每个阶段的方法、目标





## 第三章 词法分析



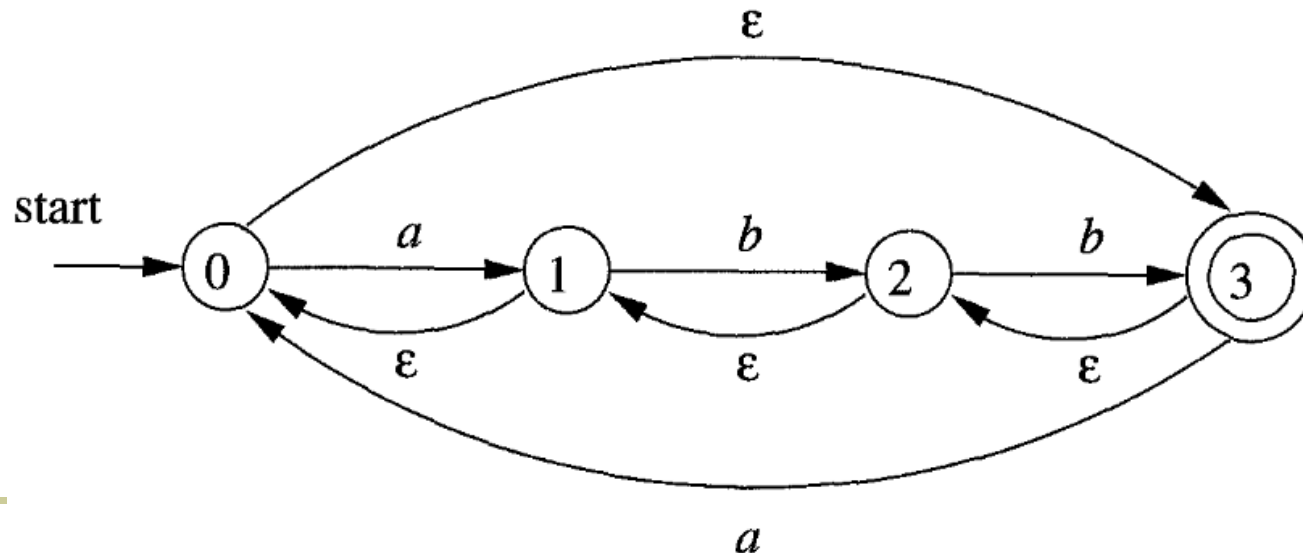
- 正则表达式、正则文法、正则语言
- 有穷状态自动机
  - DFA
  - NFA
- 几种表示之间的转换



# 作业



- 包含**5**个元音的所有小写字母串，这些串中的元音按顺序出现。
- 给出图**3-30**中**NFA**的转换表。
- 将图**3-30**的**NFA**转换为**DFA**。





## 第四章 语法分析



- 最左(右)推导、语法分析树、语言(句型)、句柄
- 文法的二义性
- 提取左公因子
- 消除左递归
- **First, Follow**
- **LL**分析与预测分析表
- **LR**分析与语法分析表
  - 项集
  - **SLR**
  - **LR(1)、LALR**
- 语法分析框架、过程



# 作业



- 考虑上下文无关文法：  
$$S \rightarrow S S + \mid S S * \mid a$$
  
以及串  $aa+a^*$ 。
  - 1) 给出这个串的一个最左推导。
  - 2) 给出这个串的一个最右推导。
  - 3) 给出这个串的一棵语法分析树。
- 为下面的文法给出预测分析表。可能需要在你的文法中消除左递归。
  - $S \rightarrow (L) \mid a$  以及  $L \rightarrow L, S \mid S$



# 作业



- 有没有可能通过某种方法修改练习**4.2.1** 中的文法，给出该文法预测分析表？
  - $S \rightarrow SS + \mid SS^* \mid a$
  
- 计算练习**4.2.1** 的文法的**FIRST** 和 **FOLLOW** 集合。
  - $S \rightarrow SS + \mid SS^* \mid a$





# 作业



- 对于文法  $S \rightarrow S S + \mid S S * \mid a$ ，指出最右句型  $SS+a*a+$  中的句柄。
- 对于文法  $S \rightarrow S S + \mid S S * \mid a$  和串  $aaa*a++$ ，说明自底向上语法分析的过程。
- 为练习4.2.1 中的（增广）文法构造 **SLR** 项集。计算这些项集的 **GOTO** 函数。给出这个文法的语法分析表。这个文法是 **SLR** 的吗？
  - （附4.2.1 的文法：  $S \rightarrow S S + \mid S S * \mid a$ ）



# 作业



- 说明下面文法是**SLR(1)**的，但非**LL(1)**的。

$S \rightarrow SA \mid A$

$A \rightarrow a$

- 为练习4.2.1 的文法**S**

$SS + \mid SS * \mid a$  构造

- 1)规范**LR** 项集族。
- 2)**LALR** 项集族。



## 第五章 语法制导翻译



- 属性文法
  - 继承属性、综合属性
- 翻译方案
  - 求值、类型表达式（变量声明）
  - 语义规则
  - 标注语法分析树



## 第5章作业



### ■ 5.1.2 扩展图5-4中的SDD，使它可以像图5-1所示的那样处理表达式。

产生式	语义规则
1) $L \rightarrow E \mathbf{n}$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow ( E )$	$F.val = E.val$
7) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

图 5-1 一个简单的桌上计算器的语法制导定义

产生式	语义规则
1) $T \rightarrow F T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow * F T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

图 5-4 一个基于适用于自顶向下语法分析的文法的 SDD



## 第5章作业



- 5.2.2 对于图5-8中的SDD，给出下列表达式对应的标注语法分析树：

- **int a, b, c**

产生式	语义规则
1) $D \rightarrow T L$	$L.inh = T.type$
2) $T \rightarrow \text{int}$	$T.type = \text{integer}$
3) $T \rightarrow \text{float}$	$T.type = \text{float}$
4) $L \rightarrow L_1, \text{id}$	$L_1.inh = L.inh$ $addType(\text{id.entry}, L.inh)$
5) $L \rightarrow \text{id}$	$addType(\text{id.entry}, L.inh)$

图 5-8 简单类型声明的语法制导定义



## 第5章作业



### ■ 5.2.4 这个文法生成了含“小数点”的二进制数：

- $S \rightarrow L.L \mid L$
- $L \rightarrow L B \mid B$
- $B \rightarrow 0 \mid 1$

设计一个L属性的SDD来计算S.val，即输入串的十进制数值。比如，串101.11应该被翻译为十进制数5.635。提示：使用一个继承属性L.side来指明一个二进制位在小数点的哪一边。



## 第5章作业



- **5.4.3** 下面的**SDT**计算了一个由**0**和**1**组成的串的值。它把输入的符号当做按照正二进制数来解释。

- $B \rightarrow B_1 0 \{ B.val = 2 * B_1.val \}$
- $\quad \mid B_1 1 \{ B.val = 2 * B_1.val + 1 \}$
- $\quad \mid 1 \{ B.val = 1 \}$

改写这个**SDT**，使得基础文法不再是左递归的，但仍可以计算出整个输入串的相同的**B.val**的值。



## 第六章 中间代码生成



- 三地址代码、**DAG**、抽象语法树.....
- 数组的翻译
- 翻译方案
  - 表达式
  - 控制流
  - 回填





# 作业



- 为下面的表达式构造**DAG**
  - $((x+y)-((x+y)*(x-y)))+(x+y)*(x-y))$
- 对下列赋值语句重复练习**6.2.1**
  - $a=b[i]+c[j]$
  - $a[i]=b*c-b*d$
- 使用图**6-22**所示的翻译方案来翻译下列赋值语句：
  - 2)  $x=a[i][j]+b[i][j]$



# 作业



- 在图6-36的语法制导定义中添加处理下列控制流构造的规则
  - 1) 一个repeat语句, **repeat S while B**
- 使用图6-43中的翻译方案翻译下列表达式。给出每个子表达式的真假值列表。可以假设第一条指令地址是**100**。



## 第七章 运行时环境



- 分配空间
- 活动树
- 活动记录 (栈)
- 垃圾回收：引用计数、标记清扫



## 第八章 代码生成及优化



- 代码生成
  - 寄存器和地址描述符
- 基本块的优化
  - **DAG**、公共子表达式、死代码、恒等式、归纳变量和强度消减.....



# 作业



- 假设**a** 和**b** 是元素为4 字节值的数组，为下面的三地址语句序列生成代码
  - (1) 四个语句的序列
    - $x = a[i]$
    - $y = b[j]$
    - $a[i] = y$
    - $b[j] = x$
  - (2) 三个语句的序列
    - $x = a[i]$
    - $y = b[i]$
    - $z = x * y$



# 作业



- 假设 $x$ 、 $y$  和 $z$  存放在内存位置中，为下面的三地址语句序列生成代码

**if  $x < y$  goto L1**

**$z = 0;$**

**goto L2**

**L1:  $z = 1$**

- 为下列基本块构造**DAG**图。

**$a[i] = b$**

**$b = c + a$**

**$d = a[j]$**

**$d = c + b$**



# 作业



- 假设有三个可用的寄存器，适用本节中的简单代码生成算法，把在练习**8.6.1-(1)**中的带的三地址代码转换为机器代码。请给出每一个步骤之后的寄存器和地址描述符。



## 第九章 优化



- 优化的来源
- 循环的优化
- 数据流分析
  - **Gen/kill、IN/OUT**
  - 活跃变量、到达定值、可用表达式





# 作业



- 对于图9-10中的流图：
  - 1) 找出流图中的循环。
  - 2) **B1**中的语句(1)和(2)都是复制语句。其中**a**和**b**都被赋予了常量值。我们可以对**a**和**b**的哪些使用进行复制传播，并把对它们的使用替换为对一个常量的使用？在所有可能的地方进行这种替换。
  - 3) 对每个循环，找出所有的公共子表达式。
  - 5) 寻找每个循环的全部循环不变计算。



# 作业



- 图9-11中是用来计算两个向量**A**和**B**的点积的中间代码。尽你所能，通过下列方式优化这个代码：消除公共子表达式，对归纳变量进行强度消减，消除归纳变量。
- 对图9-10中的流图，计算下列值：
  - 1) 每个基本块的**gen**和**kill**集合。
  - 2) 每个基本块的**IN**和**OUT**集合。
- 对图9-10的流图，计算活跃变量分析中的**def**、**new**、**IN**和**OUT**集合。