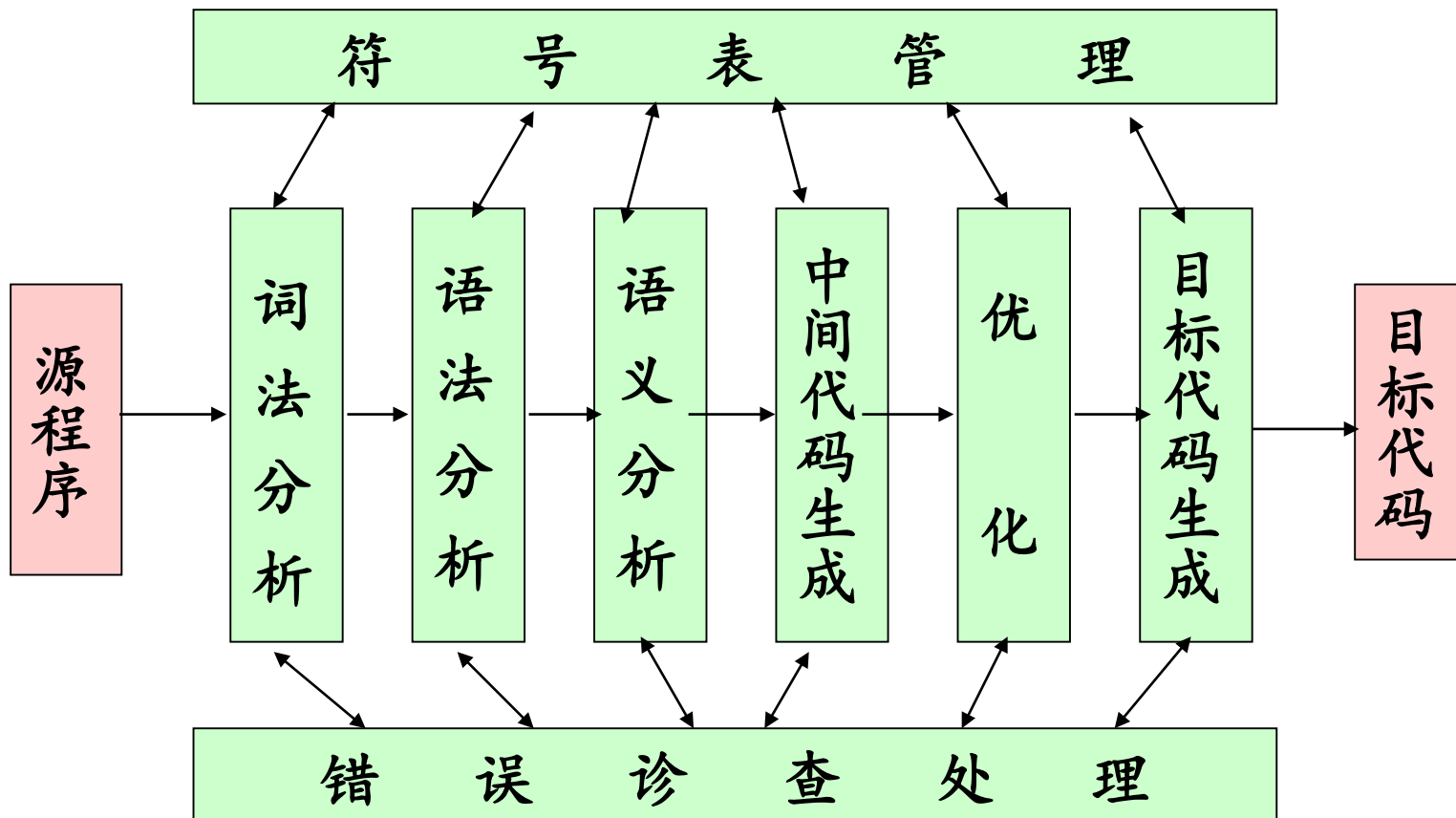




《编译技术》 内容回顾

Course Review

编译器的流程





词法分析：源代码->Token流

□ 词法分析的工作流程（以Lex为例）

正则式 \Rightarrow NFA \Rightarrow DFA \Rightarrow min DFA

□ 关键技术

■ 正则表达式的表示

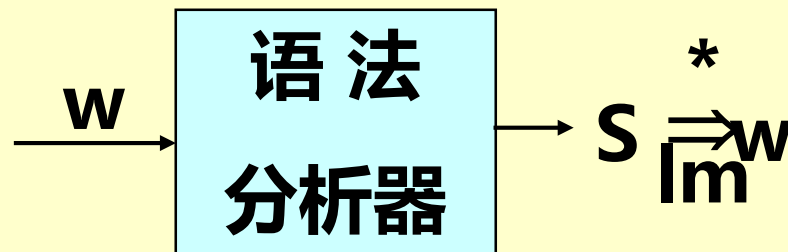
□ 正则语言

■ 正则表达式、NFA、DFA之间的转换

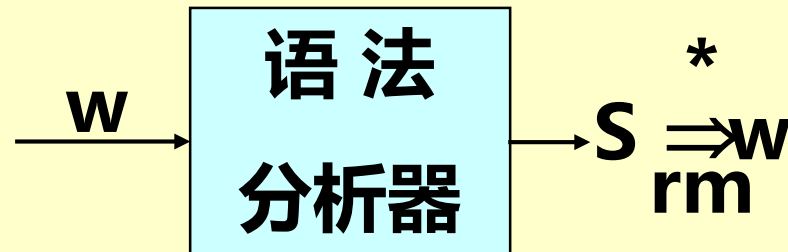
□ DFA的最小化

语法分析：Token流->分析树

自顶向下分析



自底向上分析





自顶向下分析

自顶向下分析算法的基本思想为：

对于文法 $G(Z)$ ：

若 $S \Rightarrow^+ w$ 则 $w \in L(G[Z])$ 否则 $w \notin L(G[Z])$

主要问题

- 左递归问题
- 回溯问题

主要方法

- LL分析法(first, follow)
- 递归子程序法
- 表驱动的预测分析



自底向上分析

自底向上分析算法的基本思想为：

对于文法 $G(Z)$ ：

若 $S \xleftarrow{+} w$ 则 $w \in L(G[Z])$ 否则 $w \notin L(G[Z])$

主要问题

- 句柄的识别问题

主要方法

- LR(k)分析法

句柄

- 对输入从左到右扫描，并进行自底向上语法分析，实际上可以反向构造出一个最右推导
- 句柄 (handle) :
 - 最右句型中和某个产生式右部匹配的子串，对它的归约代表了该最右句型的最右推导的最后一步；
 - 定义：如果 $S \xRightarrow{*}_{rm} \alpha A w \xRightarrow{rm} \alpha \beta w$ ；那么紧跟 α 之后的 β 是 (对应 $A \rightarrow \beta$ 的) 一个句柄
- 在一个最右句型中，句柄右边只有终结符号
- 如果文法是无二义性的，那么每个句型都有且仅有一个句柄。



语法分析的关键技术

- 文法的构造与转换
 - 消除左递归
 - 提取左公因子
- 自顶向下的语法分析
 - 递归下降子程序法
 - LL(1)分析: First/Follow集合, LL(1)分析表
- 自底向上的语法分析
 - LR分析: LR(0), SLR(1), LR(1), LALR(1)
 - LR项集族的构造
 - 识别LR(0)/LR(1)活前缀的DFA



语法制导翻译：语法分析的辅助功能

□ 语法制导定义（SDD）

- 综合属性：N的子结点或N本身的属性值来定义
- 继承属性：N的父结点、N本身和N的兄弟结点上的属性值
- S-属性, L-属性的SDD

□ 语法制导的翻译方案（SDT）

- 分析过程中实现SDT, 但并不总是可能的

□ 可以用SDT实现两类重要的SDD

- 基础文法是LR的, SDD是S属性的
- 基础文法是LL的, SDD是L属性的



中间代码生成：分析树->三地址代码

- 中间代码的常用格式
 - DAG、三地址代码等
- 通过语法制导翻译实现中间代码生成
 - 为每种语法结构构造中间代码的表示形式
 - 构造生成上述中间代码的SDD/SDT
 - 语句声明、类型构造
 - 赋值语句的翻译、数组的处理、类型检查
 - 布尔表达式与控制语句的翻译
 - if, while, do-while, for, switch
 - 回填！



目标代码生成：三地址代码->目标代码

□ 目标代码生成的关键技术

- 指令选择
- 寄存器分配
 - 图着色的寄存器分配方法

□ 简单优化

- 基本块的划分
- 控制流图（CFG）的构造
- 基本块内的活跃变量分析



编译优化：改进性能

- 了解编译优化的基本概念
- 掌握简单的数据流分析方法
 - 可达定义分析
 - 可用表达式分析
 - 活跃变量分析
- 优化是编译技术的当前研究热点
 - 新型语言
 - 新型体系结构



期末考试

□ 线上和线下结合

- 闭卷考试，百分制
- 线上：腾讯会议，手机自拍监考
- 线下：教室集中

□ 考试题目

- 50分：基本题
- 30分：中等难度题
- 20分：难题
- 基本题：平时作业，课堂练习，讲义



谢谢大家！