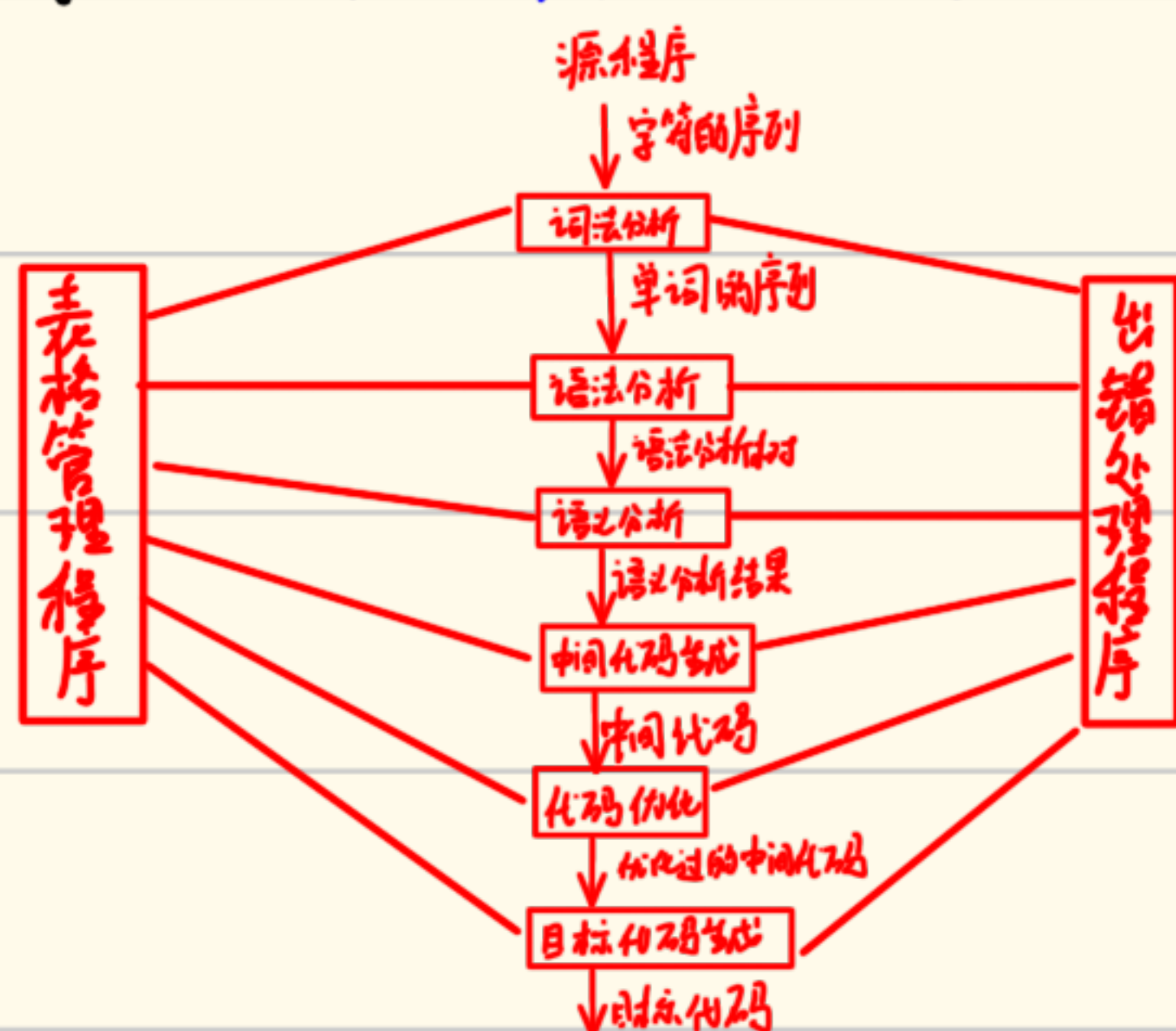


# 第一章 引论

什么是编译程序？从功能上看，编译程序就是一个语言翻译程序

编译过程：

与结构。



编译程序的结构：语法树

词法分析：从左到右一个字符一个字符地读入程序，对构成源程序的字符流进行扫描和分解，从而识别出一个单词

语法分析：将单词序列分解成各类语法短语，如“程序”、“语句”、“表达式”等，生成语法分析树

语义分析：审查源程序有无语义错误，为代码生成阶段搜集类型信息

中间代码生成：“三地址指令”的“四元式”中间代码，并称为（运算符，运算对象1，运算对象2，结果）

代码优化：使生成的目标代码更高效，即省时间又省空间。

目标代码生成：把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码

## 第2章 文法与语言

文法: 语言的结构描述

文法  $G$  定义为四元组  $(V_N, V_T, P, S)$

$V_N$ : 非终结符  $V_T$ : 终结符  $P$ : 规则  $\alpha \rightarrow \beta$  的集合,  $\alpha \in (V_N \cup V_T)^*$  且至少包含一个非终结符,  $\beta \in (V_N \cup V_T)^*$ ;  $V_N, V_T, P$  是非空有限集

$V$ : 文法的字母表

0型文法: 能力相当于图灵机

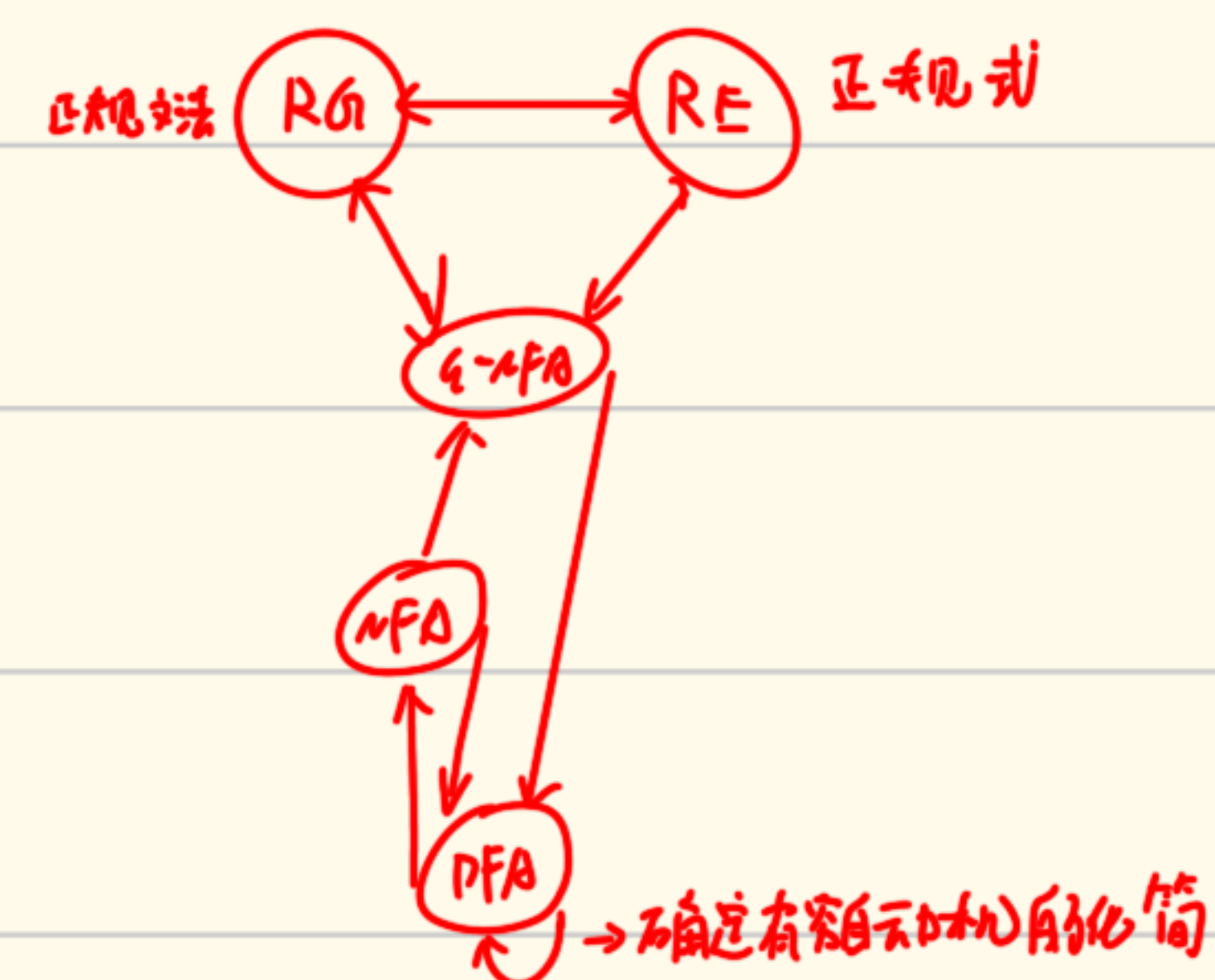
1型文法:  $\alpha \rightarrow \beta$  且  $|\alpha| \leq |\beta|$ , 上下文敏感文法

2型文法:  $\alpha \rightarrow \beta$  且  $\alpha$  是一个非终结符, 上下文无关文法

3型文法:  $A \rightarrow a$  或  $A \rightarrow aB$ , 正规文法 (RG)



## 第三章 词法分析



## 第四章 自顶向下语法分析方法

$$FIRST(\alpha) = \{a \mid \alpha \xrightarrow{*} a\beta, a \in V, \alpha, \beta \in V^*\}$$

$$FOLLOW(A) = \{a \mid S \xrightarrow{*} \dots Aa \dots, a \in V\}$$

$$SELECT(A \rightarrow \alpha) = (FIRST(\alpha) - \{\epsilon\}) \cup FOLLOW(A)$$

★ 一个上下文无关文法是 LL(1) iff 两个不同产生式  $A \rightarrow \alpha, A \rightarrow \beta$  满足  $SELECT(A \rightarrow \alpha) \cap SELECT(A \rightarrow \beta) = \emptyset$

LL(1) 文法判定方法:

① 求出能推出  $\epsilon$  的非终结符

② 求出 FIRST 集合

③ 求出 FOLLOW 集合

④ 求出 SELECT 集合

两种实现 LL(1) 方式:

递归下降 LL(1) 分析程序

表驱动 LL(1) 分析

→ 文法的预测分析表

## 第六章 LR分析

LR分析概述: (1) 总控程序  
(2) 分析表分析函数: ACTION与GOTO  
(3) 分析栈

什么是活前缀? 通俗来说, 活前缀是句柄的子集

$LR(k)$  向右查看  $k$  个符号

$SLR(1)$ : 容许  $LR(1)$  规范族中有冲突的项目集, 即向前查看  $1 \Rightarrow Follow$  集

$LRA(1)$ : 解决  $SLR(1)$  在某些情况下存在的无效归约问题

$LALR(1)$ :  $LR(1)$  合并同心集





## 第1章 语法引导的语义计算

以语法定义(上下文无关文法)为基础

用于各种语义分析与翻译过程

综合属性: 用于自下而上传递信息

继承属性: 用于自上而下传递信息

