# 第8章：程序设计语言与语言处理程序基础

【考点梳理】

## 考点1、编译与解释（★★★）

### 【考法分析】

#### 1、本知识点的考查形式主要有：给出编译与解释相关的描述，判断正误；给出编译各个阶段的描述，判断正误。

### 【要点分析】

#### 1、解释程序，也称解释器；直接解释执行源程序，或者将源程序翻译成某种中间代码后再加以执行。

#### 2、编译程序，也称编译器；将源程序翻译成目标语言程序，然后再计算机上运行目标程序。

#### 3、两者的根本区别：编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程，因此执行时效率较高；解释方式下，解释程序和源程序（或某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序，边解释边执行，执行效率较低。即：解释方式，翻译程序不生成独立的目标程序，而编译方式则生成独立保持的目标程序。

#### 4、编译过程：

#### （1）词法分析阶段：是编译过程的第一阶段，其任务是对源程序从前到后（从左到右）逐个字符扫描，从中识别出一个个“单词”符号。词法分析过程的依据是语言的词法规则，即描述“单词”结构的规则。

#### 词法错误：非法字符，关键字或标识符拼写错误。

#### （2）语法分析阶段：其任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位。通常语法分析是确定整个输入串是否构成一个语法上正确的程序。一般来说，通过编译的程序，不存在语法上的错误。

#### 语法错误：语法结构出错，if　endif不匹配，缺分号。

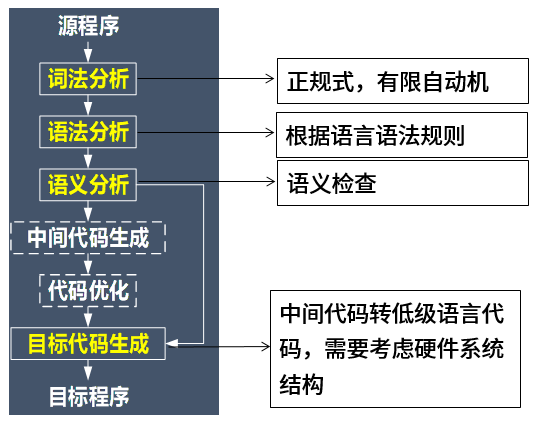
#### （3）语义分析阶段：其任务主要检查源程序是否包含静态语义错误（动态语义错误在执行过程中才能发现），并收集类型信息供后面的代码生成阶段使用。语义分析的一个主要工作是进行类型分析和检查。

#### 语义错误：死循环，零除数，其它逻辑错误。

#### （4）中间代码生成：其任务是根据语义分析的输出生成中间代码。此阶段不是必须的。常见的中间代码有：树、后缀式、三地址码（四元式）。

#### （5）代码优化：其任务是优化中间代码。此阶段不是必须的。

#### （6）目标代码生成：是编译器工作的最后一个阶段。其任务是把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码。本阶段与具体机器密切相关。



#### （7）符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的存在可以贯穿编译所有阶段。

### 【备考点拨】

#### 1、掌握编译与解释的区别；

#### 2、掌握编译器的工作过程。

## 考点2、文法（★★）

### 【考法分析】

#### 1、本知识点的主要考查方式有：给出一些概念的描述判断正误；给出一个文法的描述，判断能够识别的字符串。

### 【要点分析】

#### 1、文法相关的概念：一个形式文法是一个有序四元组G=(V，T，S，P)，其中：

#### 1）V：非终结符。不是语言组成部分，不是最终结果，可理解为占位符。

#### 2）T：终结符。是语言的组成部分，是最终结果。 VT=

#### 3）S：起始符。是语言的开始符号。

#### 4）P：产生式。用终结符替代非终结符的规则。形如α→β

#### 2、文法的分类：

#### 

#### 注：常见的程序设计语言一般是上下文无关文法。

#### 3、文法与语法树的推导：

#### 例：文法G=（{a, b}, {S, A}, S, P），其中：S→aAS|a；A→SbA|SS|ba。请构造句型aabAa的推导树。

#### S → aAS； S → a； A → SbA；A → SS； A → ba。

#### 

### 【备考点拨】

#### 1、掌握文法相关的概念和分类、对应的自动机；

#### 2、掌握语法推导树。

## 考点3、正规式（★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给出正规式指出符合要求的字符串；或给出一些相关的描述判断其正误；或结合有限自动机考查。

### 【要点分析】

#### 1、正规式是描述程序语言单词的表达式，对于字母∑，其上的正规式及其表示的正规集可以递归定义如下。

#### ① ε是一个正规式，它表示集合L(ε)={ε}。

#### ② 若a是∑上的字符，则a是一个正则式，它所表示的正规L(a)={a}。

#### ③ 若正规式r和s分别表示正规集L(r)=L(s)，则

#### （a）r|s是正规式，表示集合L(r)∪L(s)；

#### （b）r·s是正规式，表示集合L(r)L(s)；

#### （c）r\*是正规式，表示集合(L(r))\*；

#### （d）(r)是正规式，表示集合L(r)。

#### 仅由有限次地使用上述三个步骤定义的表达式才是∑上的正规式。由此可见，正规式要么为空，要么由字母、或、连接、闭包运算符组成。其中闭包运算符“\*”具有最高的优先级，连接运算具有次高优先级，或运算符“|”具有最低优先级。

#### 2、常见正规式的含义：

#### 

### 【备考点拨】

#### 1、掌握正规式相关的概念；

#### 2、掌握常见的正规式表示含义。

## 考点4、有限自动机（★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给出一个确定或不确定的有限自动机，指出其能够识别的字符串，或指出对应的正规式表示。

### 【要点分析】

#### 1、定义：M=(S,∑, δ,S0,Z)

#### 1）S是一个有限集，每个元素为一个状态

#### 2）∑是一个有穷字母表，每个元素为一个输入字符

#### 3）δ是转换函数：是一个单值对照

#### 4）S0,属于S，是其唯一的初态

#### 5）Z是一个终态集（可空）

#### 2、一个有限自动机所识别的语言是从开始状态到终止状态所有路径上的字符串的集合。要判断一个字符串能否被指定的自动机识别，就看在该自动机的状态图中能否找到一条从开始状态到达终止状态的路径，且路径上的字符串等于需要识别的字符串。而对于其正规式，可以通过能够识别的字符串去总结规律。

#### 例：下图所示的有限自动机中，s0是初始状态，s3为终止状态，该自动机不能识别（）。

#### A．abab B．aaaa C．babb C．abba

a

a

a

a

b

b

b

#### 问题解析：

#### 一个有限自动机所识别的语言是从开始状态到终止状态所有路径上的字符串的集合。要判断一个字符串能否被指定的自动机识别，就看在该自动机的状态图中能否找到一条从开始状态到达终止状态的路径，且路径上的字符串等于需要识别的字符串。

#### 对于字符串“abab”，其识别路径为s0→s1→s2→s1→s2，字符串结束时的状态不是终止状态，所以该自动机不能识别“abab”。

#### 对于字符串“aaaa”，其识别路径为s0→s1→s3→s3→s3，字符串结束时的状态是终止状态，所以该自动机可以识别“aaaa”。

#### 对于字符串“babb”，其识别路径为s0→s2→s1→s2→s3，字符串结束时的状态是终止状态，所以该自动机可以识别“babb”。

#### 对于字符串“abba”，其识别路径为s0→s1→s2→s3→s3，字符串结束时的状态是终止状态，所以该自动机可以识别“abba”。

### 【备考点拨】

#### 1、掌握有限自动机相关的基本概念；

#### 2、掌握有限自动机能够识别的字符串判断；

#### 3、掌握有限自动机与正规式的对应关系。

## 考点5、后缀表达式（★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给出概念描述判断正误；给出表达式或其语法树，要求指出其后缀表达形式。

### 【要点分析】

#### 1、前缀表达式(+ab)：把运算符写在运算对象的前面。 （语法树的先序遍历）

#### 2、中缀表达式(a+b)：把运算符写在运算对象的中间。

#### （语法树的中序遍历，是常见的表达式形式，父节点是运算符（括号不用表示），孩子结点是运算符的参数，据此可以构造出其语法树。）

#### 3、后缀表达式(ab-)：又叫做逆波兰式，把运算符写在运算对象的后面。

#### （语法树的后序遍历，常与栈结合使用。）

### 【备考点拨】

#### 1、掌握表达式构造语法树的过程；

#### 2、掌握后缀表达式相关的概念和求取过程。

## 考点6、传值与传址（★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给出一些相关的描述判断其正误；给出一些代码，其中部分参数是传值调用，部分参数是传址调用，求取最终结果。

### 【要点分析】

#### 1、函数的格式：

#### 

#### 2、传值和传址（引用）调用的对比：

#### 

#### 例：

#### 

### 【备考点拨】

#### 1、掌握传值与传址调用相关概念；

#### 2、能够根据代码，得出最终的结果（注意其中传值与传址调用的区别）。

## 考点7、多种程序语言特点（★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给出特定描述指出其对应的程序语言；或给出程序语言指出其对应描述的正误。

### 【要点分析】

#### 1、常见的程序设计语言：

#### （1）Fortran语言(第一个高级程序设计语言，科学计算，执行效率高）

#### （2）Pascal语言（结构化程序设计语言，表达能力强，Delphi）

#### （3）C语言（通用、结构化程序设计语言，指针操作能力强，高效）

#### （4）Lisp语言（函数式程序语言，符号处理，人工智能）

#### （5）C++语言（C语言基础上增加了类机制，面向对象，高效，与C兼容）

#### （6）Java语言（面向对象，中间代码，跨平台，通用的程序设计语言）

#### （7）Python（面向对象，解释型程序设计语言，胶水语言，通用的脚本语言）

#### （8）PHP（服务器端脚本语言，制作动态网页）

#### （9）Ruby（简单快捷、面向对象、脚本语言）

#### （10）Delphi（快速应用程序开发工具，可视化编程环境）

#### （11）COBOL（数据处理领域最为广泛的程序设计语言，高级编程语言）

#### （12）XML（可扩展标记语言，标准通用标记语言的子集 ）

#### （13）PROLOG（逻辑式语言，间接性，表达能力强，建造专家系统、数据库、自然语言理解、智能知识库等）

#### 注：C/C++常被用于操作系统开发；脚本语言是解释性语言。

### 【备考点拨】

#### 1、掌握常用程序设计语言的特点，能够加以区分。