### 汇编、编译、解释系统基础

注释

序言性注释 位于开头部分，给出程序整体说明，可以包括程序对软件硬件的资源要求，重要变量和参数的说明，以及程序的作者、审查者、编程日期、修改日期等。

功能性注释 一般嵌入源程序之中，主要描述相关语句的作用或注释后续程序实现什么功能。

编译过程的工作原理（核心）

词法分析 输入源程序，对源程序的字符串进行扫描分解，识别符号。有效工具为正规式和优先自动机。

语法分析 在词法分析的基础上，将单词符号分解为各类语法单位。如短语、句子、程序段

语义分析中间代码生成 对语法范畴进行静态语义检查，若正确则翻译为中间代码。通常使用属性文法描述语义规则。中间代码其实是含义明确便于处理的记号系统。

中间代码有三元式 间接三元式 四元式 树型 逆波兰记号。

优化 对中间代码加工，期望得到更有效的代码，优化是为等价交换过程。方法有公共子表达式的提取，循环优化 删除无用代码。

目标代码生成 把中间代码或优化后的代码转化为低级语言代码，依赖于机器指令。

对于逆反过程，只能得到等价的高级语言程序，并不是源程序。

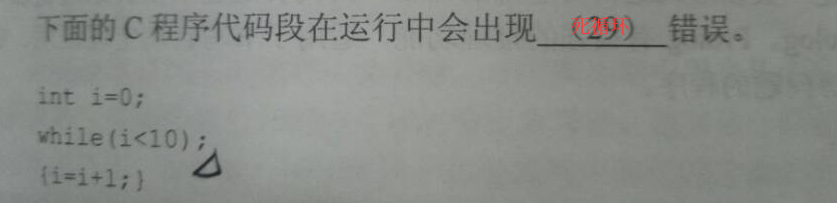
编译器 并不是所有的编译器会有目标代码和代码优化的过程。

词法错误 非法字符标识符等

语法错误 缺少分号 begin/end不匹配等

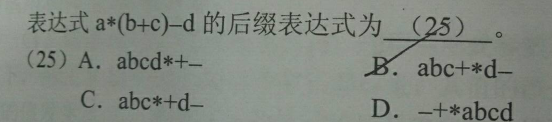
静态语义错误 类型不一致，参数不匹配

动态语义错误 逻辑错误，如死循环 分母为0等



这里的分号导致while死循环，不仅仅是分号错误，而应该是动态语义错误。

后缀表达式



编译程序与解释程序的概念（核心）

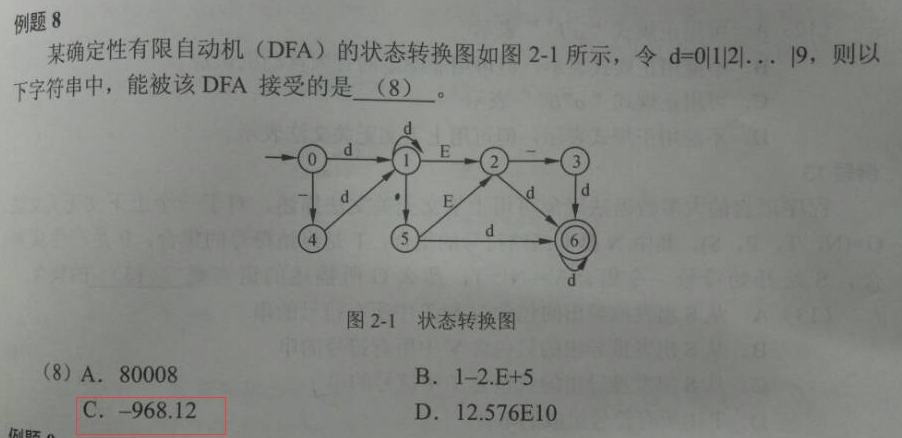
解释性：接收源程序，直接解释执行，读一句翻译一句执行一句，无目标代码。BASIC语言

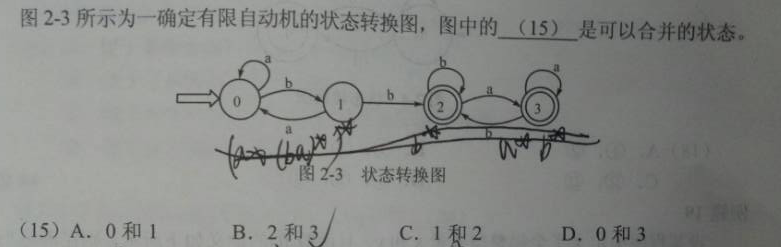
编译性：将源程序直接翻译目标语言程序，包括编译全过程。C语言 C++语言。

总结为解释性程序无编译过程，无目标代码 编译性程序有编译过程，有目标代码。

有限自动机（核心）

有限自动机分为确定有限自动机和不确定有限自动机，若他们等价，则他们可识别的记号完全相同。



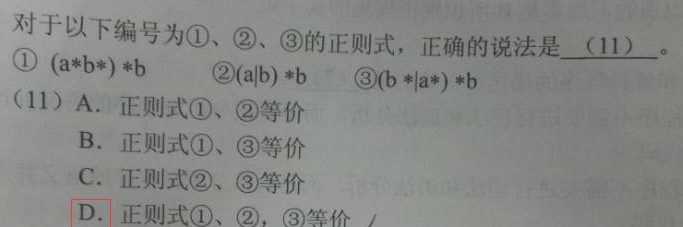


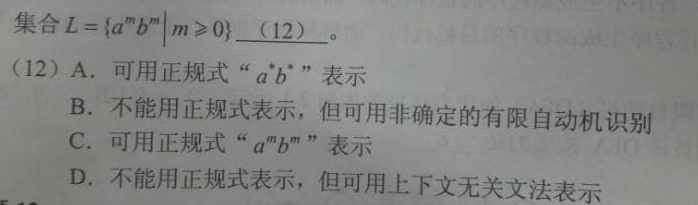
有限自动机的合并 非终端与非终端结合这里（0和1） 终端与终端结合这里（2和3）

|  |  |  |
| --- | --- | --- |
|  | 然后对于非终端（0和1）取任意状态 | 然后对于终端（2和3）取任意状态 |
| a过程 | 0过a为0 1过a为0 | 2过a为3 3过a为3 |
| b过程 | 0过b为1 1过b为2 | 2过b为2 3过b为2 |
| 总过程 | 不符合 | 符合 |

正则式与正规式（核心）

这1 2 3 都是以 a b任意组合 b结尾的正则式

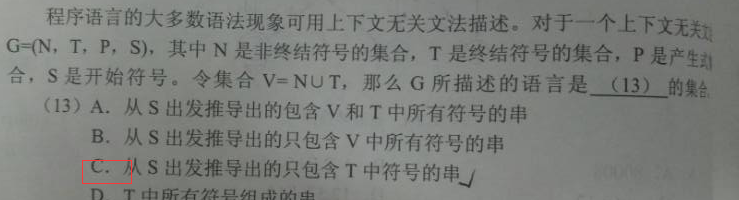




正规式只能表示没有指定次数的重复和固定次数的重复

非确定有限自动机 对于每个非确定的自动机都有一个对应的正规式。

上下无关文法 可以表示次数不固定的重复。（核心）



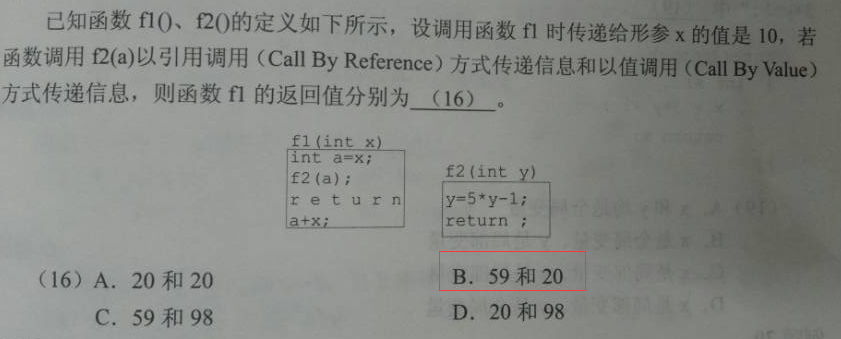
### 程序设计语言基础

函数调用

传值调用 形参取得的是实参的值，形参为局部变量，不会影响实参的值改变。（C语言支持按值调用）

引用调用 形参取得的是实参的地址，任何改变形参的操作都会改变实参（C++语言支持按值调用 引用调用）

值结果调用 形参拷贝实参，然后等调用完成后再把形参拷贝给实参。



程序语言规定

面向对象程序设计不支持对一个对象的成员变量直接访问。

程序的数据 程序数据都必须具有固定类型，作用为便于系统数据合理分配的存储单元；便于了解数据的取值范围；便于对参数表达式计算的数据对象进行检查。

C语言中，变量和常量都具有类型属性，变量可赋值，常量不可以。

### 程序设计语言特点

命令式程序设计语言

命令式程序语言是基于动作的语言，例如赋值。语言代表作：fortran Pascal C语言  
**函数式程序设计语言**

函数对映一种规则（映射）使定义域中的每个元素和值域中唯一的元素对应。

代表作 Lisp ML

面向对象程序设计语言

面向对象程序设计语言核心的是对象与类，核心概念是封装 继承 多态。

代表作C++ java smalltalk C#

逻辑程序设计语言

逻辑程序设计语言以逻辑为基础。适用于书写自动定理的证明，专家系统和自然语言理解等问题。代表作Prolog

C语言 它通过指针完成地址操作，一种低级语言但能编写高效的程序。

Fortran 应用于科学计算 一个主程序，若干子程序，大部分代码用硬件实现，执行效率高

Pascal 为教学而开发，语言表达能力强。

Prolog 以特殊的逻辑推理完成用户查询，多用于数据和专家系统。

