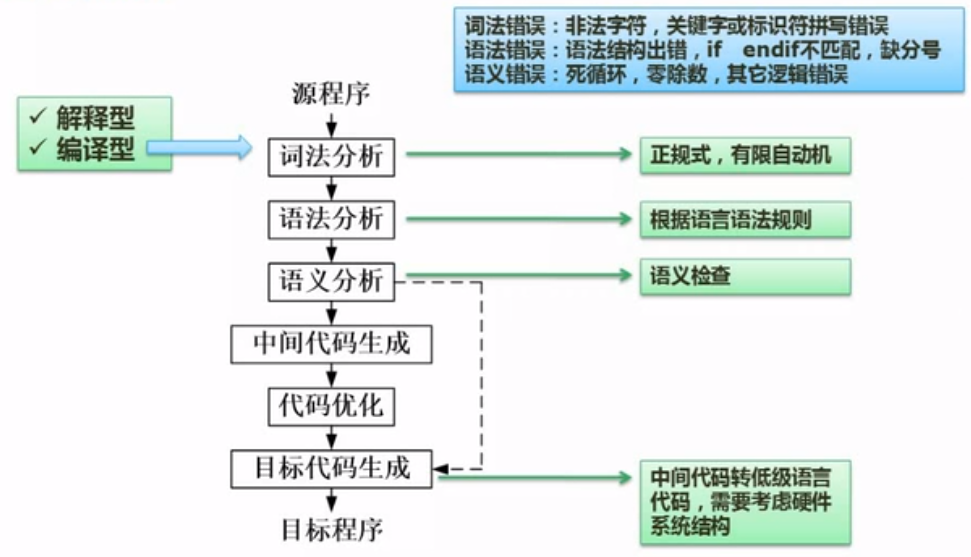
# 程序设计语言与语言处理程序

## 编译与解释

### 编译过程

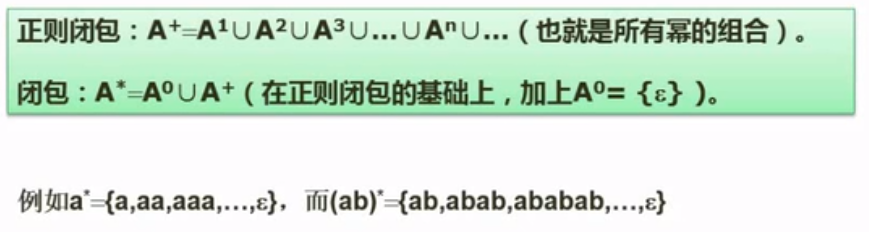


## 文法

### 文法定义

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。
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

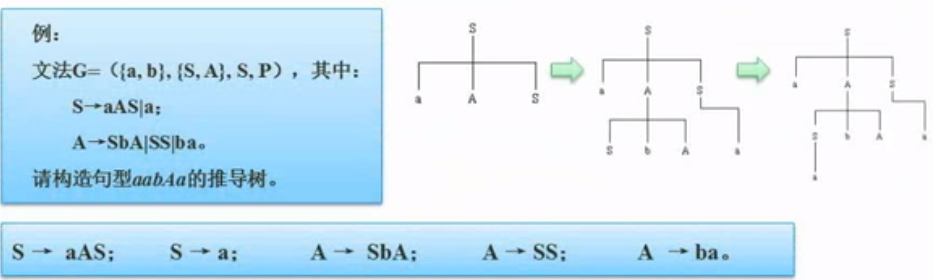


|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 别称 | 说明 | 对应自动机 |
| 0型 | 短语文法 | G的每条产生式满足α属于V的正则闭包且至少含有一个非终结符，而β属于V的闭包 | 图灵机 |
| 1型 | 上下文有关文法 | G的任何产生式满足，仅仅例外，但S不得出现在任何产生式右部 | 线性界限自动机 |
| 2型 | 上下文无关文法 | G的任何产生式为，A为非终结符，β为V的闭包 | 非确定的下推自动机 |
| 3型 | 正规文法 | G的任何产生式为或，α属于非终结符的闭包，A、B都属于非终结符 | 有限自动机 |

### 语法推导树

一颗语法树应具备以下特征：

1. 每个节点都有一个标记，此标记是V的一个符号；
2. 根节点的标记S；
3. 若一节点n至少有一个它自己除外的子孙，并且有标记A，则A肯定在VN中；
4. 如果节点n的直接子孙，从左到右的次序是节点n1,n2,…,nk，其标记分别为：A1，A2，…,Ak，那么A->A1,A2,…,Ak，一定是P中的一个生产式。



## 有限自动机



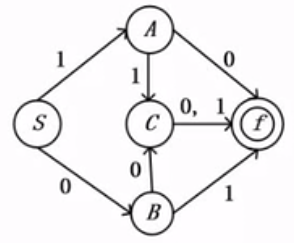
1. S是一个有限集，每个元素为一个状态
2. 是一个有穷字母表，每个元素为一个输入字符
3. 是转换函数：是一个单值对照
4. S0，属于S，是其唯一的初态
5. Z是一个终态集（可空）

有限状态自动机可以形象的用状态转换图表示，设有限状态自动机：

DFA = ({S,A,B,C,f}，{1,0}，{f})，

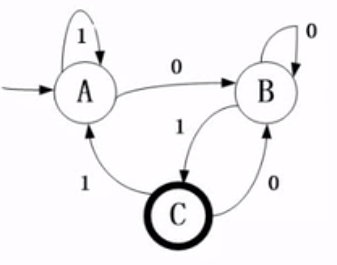
其中：





有限自动机例题：

下图所示为一个有限自动机（其中A是初态，C是终态），该自动机可识别：C



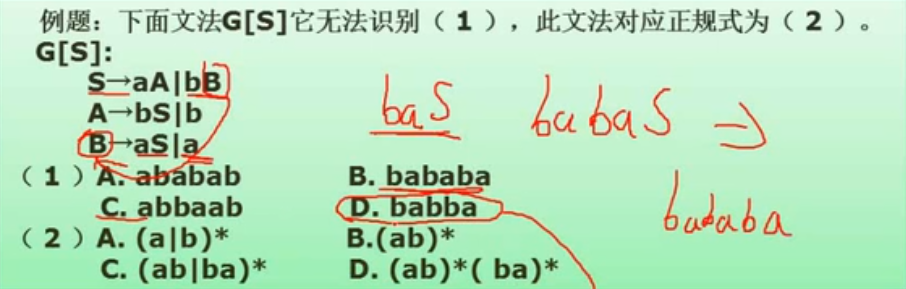
A．0000 B.1111 C.0101 D.1010

## 正规式\*\*\*

正规式式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：

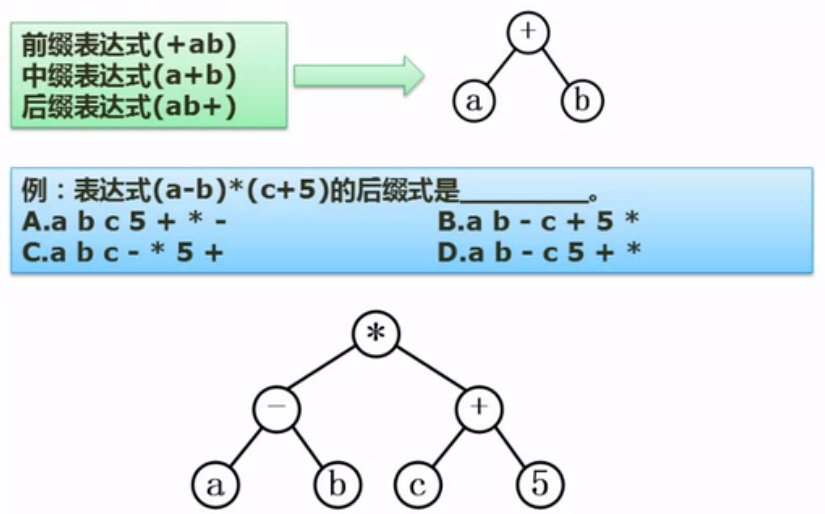
1. 是一个正规式，他表示集合。
2. 若a是上的字符，则a是一个正规式，他所表示的正规L(a)={a}
3. 若正规式r和 s分别表示正规集L(r )=L(s)，则
4. r|s是正规式，表示集合L(r)∪L(s)
5. r·s是正规式，表示集合L(r)L(s)
6. r\*是正规式，表示集合(L(r))\*
7. (r)是正规式，表示集合L(r)

仅由有限次的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“\*”具有最高优先级，连接运算符具有次高优先级，或运算符“|”具有最低优先级。



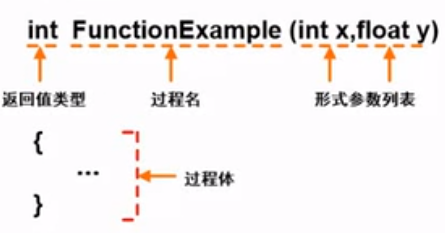
D C

## 表达式\*



D

## 传值与传址\*\*



|  |  |
| --- | --- |
| 传递方式 | 主要特点 |
| 传值调用 | 形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变 |
| 引用（传址）调用 | 形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值 |

## 多种程序语言特点

1. Fortran语言（科学计算，执行效率高）
2. Pascal语言（为教学而开发，表达能力强，Delphi）
3. C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（面向对象，高效）
6. Java语言（面向对象，中间代码，跨平台）
7. C#语言（面向对象，中间代码，.NET）
8. Prolog语言（逻辑推理，简洁性，表达能力，数据库和专家系统）