追寻知识的过程

编程语言通识
50 年代计算机语言发展黄金期

语言按语法分类
- 非形式语言
  - 中文、英文
- 形式语言（乔姆斯基谱系）
  - 0 型，无限制文法
  - 1 型，上下文相关文法
  - 2 型，上下文无关文法
  - 3 型，正则文法。主流语言大部分语言达不到，但是基本上都会划分为词法和语法两个部分，词法用正则做一遍粗略的处理，把语言变成单个的词，作为输入流，去做语法分析。特例：LISP

产生式（BNF）
- 用尖括号括起来的名称来表示语法结构名
- 语法结构氛围基础结构和需要用其他语法结构定义的复合结构
  - 基础结构称终结符
  - 复合结构称非终结符
- 引号和中间的字符表示终结符
- 可以有括号
- * 表示重复多次
- | 表示或
- + 表示至少一次

```
<Number> = "0" | "1" | "2" | ..... | "9"
<DecimalNumber> = "0" | (("1" | "2" | ..... | "9") <Number>*)
<AdditiveExpression> = <DecimalNumber> | <AdditiveExpression> "+" <DecimalNumber>

<MultiplicativeExpression> = <DecimalNumber> |
                             <MultiplicativeExpression> "*" <DecimalNumber> |
                             <MultiplicativeExpression> "/" <DecimalNumber>
=>
<AdditiveExpression> = <MultiplicativeExpression> |
                       <AdditiveExpression> "+" <MultiplicativeExpression> |
                       <AdditiveExpression> "-" <MultiplicativeExpression>

<LogicalExpression> = <AdditiveExpression> |
                      <LogicalExpression> "||" <AdditiveExpression> |
                      <LogicalExpression> "&&" <AdditiveExpression>

<PrimaryExpress> = <DecimalNumber> |
                   "(" <LogicalExpression> ")"
```

通过产生式理解
- 0 型
  - ? ::= ?
- 1 型
  - ? <A> ? ::= ? <B> ?
- 2 型
  - <A> ::= ?
- 3 型（只允许左递归）
  - <A> ::= <A> ? ✔
  - <A> ::= ?<A> ✘

终结符 加粗
词法定义 双冒号::
语法定义 单冒号:

现代语言的特例

- C++中，* 可能表示乘号或者指针，具体是哪个，取决于星号前面的标识符是否被声明为类型
- VB中，< 可能是小于号，也可能是XML直接量的开始，取决于当前位置是否可以接受XML直接量
- Python中，行首的tab符和空格会根据上一行的行首空白以一定规则被处理成虚拟终结符indent或者dedent
- JavaScript中，/ 可能是除号，也可能是正则表达式开头，处理方式类似于VB，字符串模板中也需要特殊处理}，还有自动插入分号规则

C++ 语法和语义相关 ==> 非形式语言！

图灵完备性
- 命令式 - 图灵机
  - goto
  - if 和 while
- 声明式 - lambda
  - 递归

动态与静态
- 动态
  - 在用户的设备/在线服务器上
  - 产品实际运行时
  - Runtime
- 静态
  - 在程序员的设备上
  - 产品开发时
  - Compiletime

类型系统
- 动态类型系统与静态类型系统
- 强类型（**无隐式转换**）与弱类型（**有隐式转换**）
- 符合类型
  - 结构体
  - 函数签名
- 子类型
  - 逆变/协变

一般命令式编程语言
- Atom
  - Identifer
  - Literal
- Expression
  - Atom
  - Operator
  - Punctuator
- Statement
  - Expression
  - Keyword
  - Punctuator
- Structure
  - Function
  - Class
  - Process
  - Namespace
  - ...
- Program
  - Program
  - Module
  - Package
  - Library

语法 → 语义 → 运行时