计算和计算机

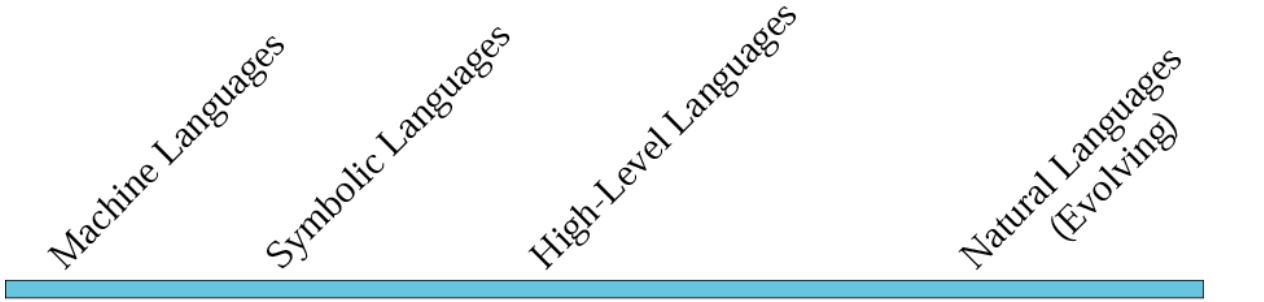
- 计算机：实现计算的自动机器
- 计算：一类按规则操作抽象符号 (序列) 的过程
  - 例如，四则运算即是按算术规则，操作四则运算表达式 (由 10 个数字符号和一些表示运算符的符号构成的序列)
- 用计算机实现计算的基础
  - 定义一套方式，描述各种基本操作 (如加减乘除等)
  - 定义一套方式，描述如何执行一批操作的过程
  - 制造出一种机器 (基于电子技术)，能自动地
    - 识别基本操作的描述，并实现相应操作
    - 识别有关执行过程的描述，并实现相应的过程

程序

- (现实生活中的) 程序：为完成某项事务，将一系列 (简单) 步骤按确定的方式排列所成的操作序列
- (计算机) 程序：一段静态的符号序列 (文本)
  - 用途：指挥计算机的执行，从而实现特定的计算过程
    - 计算机是 (基于程序的) 通用的计算机器
  - 内容：指令的序列
    - 描述了计算机执行中需要完成的操作及其执行顺序
  - 依赖于某种适当的描述方式 — 编程语言
    - 计算机能处理 — 有严格定义的形式
    - 便于人使用 — 适合写、读、理解、学习等

程序

- (现实生活中的) 程序：为完成某项事务，将一系列 (简单) 步骤按确定的方式排列所成的操作序列
- (计算机) 程序：一段静态的符号序列 (文本)
  - 用途：指挥计算机的执行，从而实现特定的计算过程
    - 计算机是 (基于程序的) 通用的计算机器
  - 内容：指令的序列
    - 描述了计算机执行中需要完成的操作及其执行顺序
  - 依赖于某种适当的描述方式 — 编程语言



Evolution of computer languages

编程语言 (1)

- 计算机的核心功能部件：CPU (中央处理器)
  - CPU 能执行 (不多的) 一组指令，每条指令对应一个特定操作 (如移动一个单元的数据，或一次算术运算等)
  - CPU 可执行的指令 (集) 用一套规定形式的二进制编码表示 (0/1 串)，构成了一种描述程序的语言 — 机器语言 (Machine language)
- 机器语言程序：CPU 指令的简单序列
  - CPU 能直接执行
  - 不适合人使用 (编写/修改/阅读/理解…)

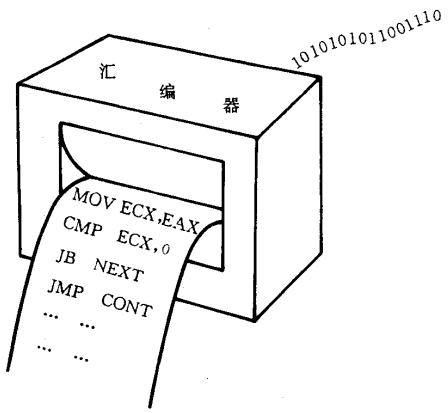
00000001000000001000	将内存单元 1000 的数据装入寄存器 0
00000001000100001010	将内存单元 1010 的数据装入寄存器 1
00000101000000000001	将寄存器 1 的数据乘到寄存器 0的原有数据上
⋮	⋮

编程语言 (2)

- 机器语言不适合人使用，人们开发了与机器语言直接对应的符号形式的语言 — 汇编语言 (Assembly language)
  - 汇编语言具有易读的指令形式

load 0 a	将内存单元 a 的数据装入寄存器 0
load 1 b	将单元 b 的数据装入寄存器 1
mult 0 1	将寄存器 1 的数据乘到寄存器 0的原有数据上
⋮	⋮

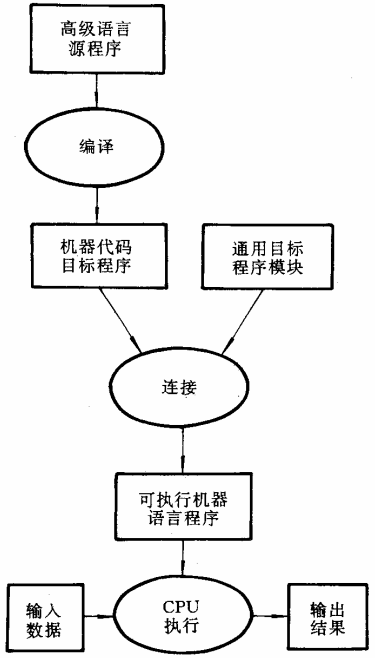
  - 通过汇编程序 (Assembler) 完成从汇编语言到机器语言的 (机械) 翻译



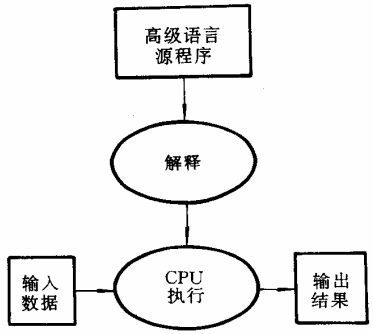
- 汇编语言仍然是面向计算机的语言，概念低级，难以描述复杂的计算过程
  - 1954 年 IBM 发布第一个完全符号形式的高级语言 Fortran，开创了计算机程序设计的新时代

编程语言 (3)

- 高级语言程序：不能直接被计算机执行，有两种基本处理方式
  - 编译 (compile)：通过编译程序把高级语言程序翻译为等价的机器语言程序，然后交计算机执行
  - 解释 (interpret)：在计算机上实现一个程序 (解释系统) 来识别某种高级语言程序，模拟其计算过程，实现其行为



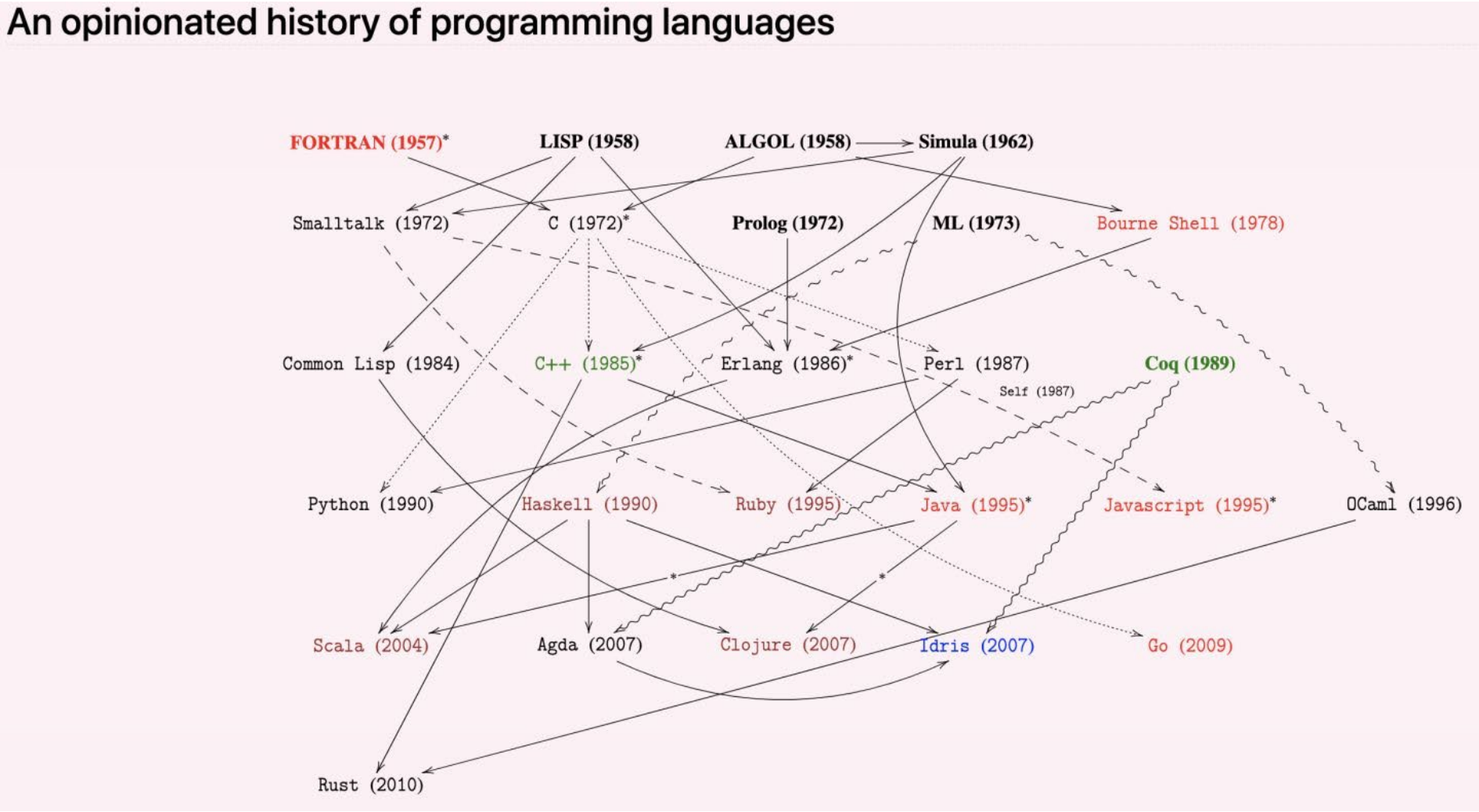
- (编程) 语言的三要素
  - 语法：合法描述 (的程序或程序部分) 的形式
  - 语义：程序或程序部分的意义 (计算机处理它的时候做什么，完成什么计算)
  - 语用：使用语言的技术 (程序开发的技术、方法、经验等)



高级语言的发展

- **Ref: History and Evolution of Programming Languages**
  - **Years 50: Creation of high-level languages (closer to humans).**
  - **Years 60: Expansion of specialized languages. Forth. Simula I. Lisp, Cobol. Trying unsuccessfully to impose general languages: Algol, PL / 1.**
  - **Years 70: Duel between structured programming with Pascal and efficiency of C language.**
  - **Years 80: Experimentating other ways including objects. ML. Smalltalk.**
  - **Years 90: Generalization of object-oriented programming with the performance of microcomputers. Java, Perl, Python languages.**
  - **2000's: Internet Programming**
  - **2010's: Concurrency and asynchronicity. JavaScript and Go languages among others help to create online fluid applications.**

History of programming languages



截图来源：https://artagnon.com/articles/pl