函数式编程与程序语言理论简介（内容尚不完整）

Shreck Ye

2023 年 12 月

编程语言的常见范式

[3]

- **命令式编程**
  the programmer instructs the machine how to change its state
  包括常见的工业编程语言 C、C++、Java、JavaScript **等**
  - **过程式编程**
  - **面向对象编程**
- **函数式编程**
  the desired result is declared as the value of a series of function applications
- **逻辑编程**
  in which the desired result is declared as the answer to a question about a system of facts and rules

更具体地回答什么是函数式编程——一些显著特征

- 不可变变量（由于不可变，“变量”在很多函数式语言中也叫做绑定（binding）或定义（definition））
- 表达式代替语句
- 没有循环控制结构（for、while、goto 等）
  问题：怎么样实现循环？

如果一门语言上面三者的要求都严格满足，那么这门语言可以被称作“纯”函数式编程语言。
根据以上内容很容易回答问题：为什么 C/C++、Java、Python、JavaScript 等一般不被称为函数式编程语言？

函数式编程的一些历史

- Alonzo Church（阿隆佐 · 邱奇）提出（无类型）Lambda 演算法 [2]
  作者简介：函数式编程与程序语言理论的祖师爷、图灵在美国留学期间的导师
  这可以看作函数式编程的理论起源。
  Lambda calculus (also written as -calculus) is a formal system in mathematical logic for expressing computation based on function abstraction and application using variable binding and substitution.
  邱奇-图灵论题：可（有效）计算的函数可以 = 图灵机可计算的函数 = 邱奇的 Lambda 演算可计算的函数
- Lisp 语言
- Can programming be liberated from the von Neumann style?: a functional style and its algebra of programs
  作者 John Backus，简介：1977 年图灵奖得主、FORTRAN 语言的创造者
  这可以看作在工业界应用函数式语言的起源。

函数式编程的一些概念

- 头等函数（函数作为一等公民）
- 高阶函数
- Lambda 表达式与闭包
- 代数数据类型（ADT）与模式匹配

其他的一些概念

函数式编程在工业界的一些影响和应用

虽然上述众多工业界语言不是函数式语言，但都在近年来逐渐支持了函数式编程这一范式，最显著的一些特性便是 lambda 表达式和类型参数/泛型（Java）/模板（C++）的支持。
常见的函数式编程语言有：Haskell、ML 系（Ocaml、Standard ML）、Scala、Lisp 语言及其各种变体、PureScript、F#/F*
集合高阶函数/集合函数式 API/面向表达式编程
问题：常见的一种编程需求是在一列元素里面找出需要的元素
-> filter

- map
- filter
- reduce

并行计算、分布式计算与大数据：Google MapReduce 与 Apache Spark
一个数学原理：reduce 与结合律和群论

一些常见的争议话题

一个大家比较认可的回答：

- 最好的编程语言是什么？现在来讲，没有一个单一的最好的编程语言，特别是对于工程
  普遍来讲，适合的就是最好的，适合体现在效率（开发效率与运行效率）、生态（第三方开源库生态与劳动力市场生态）等多方面
- 程序语言的语法很重要吗？
  语法的重要性并没有那么大，对语义的研究的重要性高于对语法的研究
  Wadler 定律（Wadler's Law）[4]：
  In any language design, the total time spent discussing a feature in this list is proportional to two raised to the power of its position.
  0. *Semantics*
  1. *Syntax*
  2. *Lexical syntax*
  3. *Lexical syntax of comments*

总结和建议

- 为什么要用函数式编程？
  优点：更为简洁、增加代码复用、让程序更容易分析、提高软件安全性（减少程序 bug）
  增加代码复用的例子：提取函数类型参数的公共函数

推荐学习材料

- 学习你所熟悉的语言的函数式编程教程
- Coursera 课程
  - Programming Languages
  - Functional Programming in Scala
- 知乎博主与其上的 Haskell 等语言的教程

推荐的一些编程语言（含个人见解）

希望学习函数式编程的特性与理论：

Haskell 想学习函数式编程和其中的各种高级特性与体验惰性求值

ML 同上但不想要惰性求值

Racket Lisp 的现代版，动态类型函数式编程

将函数式编程应用到工程开发中：

Scala 想要一门兼容多范式又简洁抽象的语言，能体验函数式编程的同时兼容面向对象编程与 Java 的生态（Spark、Kafka、Twitter 的实现语言，不过近年来发展比较颓势）

Kotlin 简化版的 Scala，现在的最大特性和主要目标是接近平台原生性能的多平台开发（服务器端、web、Android、iOS）

TypeScript 更加类型安全的 JavaScript

Rust 剪掉了很多冗杂特性且更加类型安全、内存安全的同时不牺牲性能的面向未来的 C++

更多

语法糖的含义与讨论
可变性 + OOP 的封装在并发计算方面的缺陷
函数式编程现今的一些局限性和缺点：在一些地方性能较差、学习难度更大

依赖类型语言与辅助证明

Curry-Howard 对应：类型即命题，程序（实现）即证明

依赖类型编程的一些实际应用

带维数的矩阵、参数约束（对比 guarded programming）

Lambda 立方

值与类型的依赖关系

数学归纳法的推广：结构归纳法

第二数学归纳法的推广：良基归纳法

与范畴论的联系

currying 与范畴论
Curry–Howard–Lambek 对应 [1]（Robert Harper 称为计算三位一体论（computational trinitarianism））：计算、逻辑、范畴/空间是看待数学基础的三种不同的角度。

更多话题

类型语言	Coq、Agda、Lean、Idris、Epigram、Isabella 等等
T 与 Coc	
e Axiom)	相等关系在依赖类型中被表示为想等类型，相等的证明被表示为相等类型的元素，那么相等类型的元素是否有多个？(以及它们与同伦群的关系）能不能把同构的类型看作相等的？
type case	是否应该对类型进行模式匹配？
自动证明	有了辅助证明语言以后，是否可以对数学定理进行自动证明？它有那些局限性？(哥德尔不完备定理）现在大火的基于机器学习的 AI 能不能帮上忙？

程序语言理论与依赖类型相关的一些推荐内容：

- 无穷类型咖啡（∞-type Café）暑假学校
- Codewars 上有一些很不错的各方面入门练习题

[1] *Curry–Howard correspondence - Wikipedia*. URL: https://en.wikipedia.org/wiki/Curry%E2%80%93Howard_correspondence#Curry%E2%80%93Howard%E2%80%93Lambek_correspondence. (accessed: 12.09.2023).

[2] *Lambda calculus - Wikipedia*. URL: https://en.wikipedia.org/wiki/Lambda_calculus. (accessed: 12.09.2023).

[3] *Programming paradigm - Wikipedia*. URL: https://en.wikipedia.org/wiki/Programming_paradigm. (accessed: 12.06.2023).

[4] *Wadler's Law - HaskellWiki*. URL: https://wiki.haskell.org/Wadler's_Law. (accessed: 12.10.2023).