

**函数式编程报告**

姓名：钟子琛

学院：计算机科学与技术

专业：计算机科学与技术

班级：CS1703

学号：U201714607

指导教师：莫益军

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2020 年. 4月. 29日

目录

[一、目标 3](#_Toc39164327)

[二、动机 4](#_Toc39164328)

[三、报告主题内容 5](#_Toc39164329)

[函数式编程的起源与发展过程 5](#_Toc39164330)

[函数式编程语言的家族成员及其简介 7](#_Toc39164331)

[LISP 7](#_Toc39164332)

[Haskell 7](#_Toc39164333)

[Scheme 7](#_Toc39164334)

[Erlang 8](#_Toc39164335)

[函数式编程语言的优势 8](#_Toc39164336)

[四、收获 10](#_Toc39164337)

[五、建议 11](#_Toc39164338)

# 一、目标

1.了解函数式编程的发展历史，了解泛函程序的几种代表性语言和特点。

2.理解函数式编程与命令式编程的差异，掌握函数式程序设计思想、基本概念、原理和方法。

3.以ML作为教学语言，掌握常用函数的应用及函数程序的测试方法，包括函数的基本使用、求值类型、程序的推理和证明，串/并行程序的运行及效率分析等，理解并行编程的思想、掌握并行程序的设计方法。

4.通过应用案例的分析，理解多态、高阶函数的特点，介绍如何编写函数式程序以解决实际问题，培养函数式编程能力。

# 二、动机

在完成了函数式编程的课程后，为了巩固自己所学习的知识，我准备从以下两个方面来讨论：

① 函数式编程的起源与发展过程

② Standard ML编程语法规范

一方面，通过了解函数式编程的起源与发展，我能够更加清晰地了解到函数式编程地重要性和实用性。另一方面，通过再次巩固ML语法规范，能够让我更加熟练地使用ML进行函数式编程。

# 三、报告主题内容

## 函数式编程的起源与发展过程

ML是一个通用的函数式编程语言，它是由爱丁堡大学的Robin Milner等人在二十世纪七十年代晚期开发的。ML是作为LCF项目的元语言（Meta Language）设计的，这也是其名字的来历。它的最初用途是写一些可以生成数学证明的程序。今天，大多数著名的推理系统都是用ML写的。目前ML有两个发展分支：StandardML和Caml。 StandardML是标准ML，简称为SML。 ML是一种强类型函数程序设计语言（强类型：变量有且只有一种类型）。它拥有自然的语法和较少的基本概念，其理论基础是λ演。

ML一般被归为非纯函数式编程语言，因为它允许副作用和指令式编程。这一点和纯函数式编程语言例如Haskell不一样。今天在ML家族中有好几种语言：两种主要的语言是Standard ML和Caml，其他的包括F# － 针对Microsoft .NET平台的开放研究项目。 ML中的思想影响了众多的语言，例如Haskell，Cyclone和Nemerle。ML的大多被用于语言设计和操作（编译器、分析器、定理证明机）， 但是它作为通用语言也被用于生化，金融系统，和宗谱数据库，一个P2P的客户/服务器程序等等。

ML可以算一种具备命令式语言特点的函数型语言，或者说面向函数的命令式语言。和Lisp一样，ML具有非常灵活的函数功能。例如一个表达式的值可能就是一个函数，这个函数可以被作为参数传递给另一个函数，或者函数的返回值就是一个函数。同时和Algol类的语言比较接近的是，ML的语法象命令型的，而且用起来象用Algol家族的很多比较新的后代们一样方便。而且ML有并行扩展，可以用来写并行系统；甚至还有面向对象扩展。

ML是Robin Milner主管LCF项目时设计的。LCF项目是受Dana Scott给出的一组逻辑原则启发而设立的，致力于开发一种“可计算函数逻辑”（Logic of Computable Functions）。Robin Milner的目标是构造一个方便实用的系统，来自动的或者半自动的证明函数程序中一些有趣的性质。他的LCF项目于1970年在Standford开始，并于1980年代在Edinburge继续进行。期间取得了很多重要进展，并且激发了相关领域的一系列研究工作。

## 函数式编程语言的家族成员及其简介

### LISP

LISP是一种通用高级计算机程序语言，长期以来垄断人工智能领域的应用。LISP作为因应人工智能而设计的语言，是第一个声明式系内函数式程序设计语言，有别于命令式系内过程式的C、Fortran和面向对象的Java、C#等结构化程序设计语言。

### Haskell

Haskell是一种标准化的、通用纯函数式编程语言，有非限定性语义和强静态类型。它的命名源自美国逻辑学家Haskell Brooks Curry，他在数学逻辑方面的工作使得函数式编程语言有了广泛的基础。在Haskell中，函数是一等公民。作为函数式编程语言，主要控制结构是函数。Haskell语言是1990年在编程语言Miranda的基础上标准化的，并且以λ演算（Lambda-Calculus)为基础发展而来。具有“证明即程序、结论公式即程序类型”的特征。这也是Haskell语言以希腊字母「λ」（Lambda）作为自己标志的原因。Haskell语言的最主要的执行环境是GHC。

### Scheme

Scheme 编程语言是一种Lisp方言，诞生于1975年，由 MIT 的 Gerald J. Sussman 和 Guy L. Steele Jr. 完成。它是现代两大Lisp方言之一。Scheme的一个主要特性是可以像操作数据一样操作函数调用。Scheme 是 MIT 在70年代创造出来，目的之一是训练人的计算思维，以其简洁的语言环境和大量的脑力思考而著称。

### Erlang

Erlang是一种通用的面向并发的编程语言，它由瑞典电信设备制造商爱立信所辖的CS-Lab开发，目的是创造一种可以应对大规模并发活动的编程语言和运行环境。Erlang问世于1987年，经过十年的发展，于1998年发布开源版本。Erlang是运行于虚拟机的解释性语言，但是现在也包含有乌普萨拉大学高性能Erlang计划（HiPE）开发的本地代码编译器，自R11B-4版本开始，Erlang也开始支持脚本式解释器。在编程范型上，Erlang属于多重范型编程语言，涵盖函数式、并发式及分布式。顺序执行的Erlang是一个及早求值, 单次赋值和动态类型的函数式编程语言。

Erlang是一个结构化，动态类型编程语言，内建并行计算支持。最初是由爱立信专门为通信应用设计的，比如控制交换机或者变换协议等，因此非常适 合于构建分布式，实时软并行计算系统。使用Erlang编写出的应用运行时通常由成千上万个轻量级进程组成，并通过消息传递相互通讯。进程间上下文切换对于Erlang来说仅仅 只是一两个环节，比起C程序的线程切换要高效得多得多了。

使用Erlang来编写分布式应用要简单的多，因为它的分布式机制是透明的：对于程序来说并不知道自己是在分布式运行。Erlang运行时环境是一个虚拟机，有点像Java虚拟机，这样代码一经编译，同样可以随处运行。它的运行时系统甚至允许代码在不被中断 的情况下更新。另外如果需要更高效的话，字节代码也可以编译成本地代码运行。

### 函数式编程语言的优势

#### 代码简洁，开发快速

函数式编程大量使用函数，减少了代码的重复，因此程序比较短，开发速度较快。

#### 接近自然语言，易于理解

函数式编程的自由度很高，可以写出很接近自然语言的代码。

前文曾经将表达式(1 + 2) \* 3 - 4，写成函数式语言：

subtract(multiply(add(1,2), 3), 4)

对它进行变形，不难得到另一种写法：

add(1,2).multiply(3).subtract(4)

这基本就是自然语言的表达了。

#### 更方便的代码管理

函数式编程不依赖、也不会改变外界的状态，只要给定输入参数，返回的结果必定相同。因此，每一个函数都可以被看做独立单元，很有利于进行单元测试和除错，以及模块化组合。

#### 易于并发编程

函数式编程不需要考虑"死锁"，因为它不修改变量，所以根本不存在"锁"线程的问题。不必担心一个线程的数据，被另一个线程修改，所以可以很放心地把工作分摊到多个线程，部署"并发编程"。

多核CPU是将来的潮流，所以函数式编程的这个特性非常重要。

#### 代码的热升级

函数式编程没有副作用，只要保证接口不变，内部实现是外部无关的。所以，可以在运行状态下直接升级代码，不需要重启，也不需要停机。Erlang语言早就证明了这一点，它是瑞典爱立信公司为了管理电话系统而开发的，电话系统的升级当然是不能停机的。

# 四、收获

这次课程的收获主要有以下几点：

1. 学习了一门新的语言SML，这种语言配合函数式编程的思维让程序看起来十分简洁。
2. 在学习了SML语言的基础上，了解了函数式编程的思维。这种思维让我在编写程序时的思路更加清晰，效率更高。
3. 函数式编程的思维不仅仅只在本次课堂学习中使用，这种思维将会一直影响着我，无论以后使用何种语言编写程序，都能够按照这种思路，让自己的程序更有逻辑，在工作合作时让其他人更容易理解。

# 五、建议

这次的学习总体来说还是相当顺利的，一方面是老师的视频讲解清晰，另一方面是函数式编程的思想本身也比较容易理解，并且确实十分方便。在学习中遇到的主要问题在于，对SML语言的不熟悉，很多时候遇到的问题需要上网查找资料来解决，实验的时候配值环境等等也遇到了很多问题，希望老师在以后的课堂中能够增加这方面的内容，让我们的学习能够更加有效率。