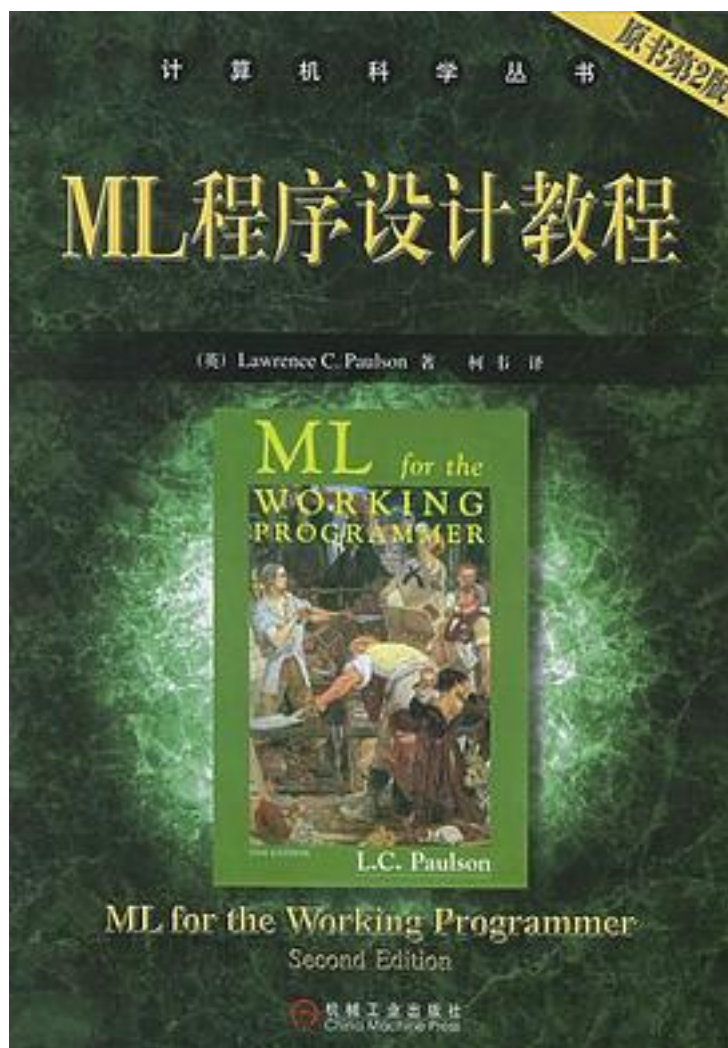


# ML程序设计教程



[ML程序设计教程\\_下载链接1](#)

著者:保罗森

出版者:机械工业出版社

出版时间:2005-5

装帧:平装

isbn:9787111161219

本书是关于ML程序设计的经典教材，详细介绍如何使用ML语言进行程序设计，并讲解函数式程序设计的基本原理。

书中含有大量例子，涵盖了排序、矩阵运算、多项式运算等方面。大型的例子包括一个一般性的自顶向下语法分析器、一个L-演算归约程序和一个定理证明机。书中也讲述了关于数组、队列、优先队列等高效的函数式实现，并且有一章专门讨论函数式程序的形式论证。本书的代码均可以从作者网站 (<http://www.cl.cam.ac.uk/users/lcp/>) 得到。

本书详细讲解如何使用ML语言进行程序设计，并介绍函数式程序设计的基本原理。书中特别讲述了为ML的修订版所设计的新标准库的主要特性，并且给出大量例子，涵盖排序、矩阵运算、多项式运算等方面。大型的例子包括一个一般性的自顶向下语法分析器、一个L-演算归约程序和一个定理证明机。书中也讲述了关于数组、队列、优先队列等高效的函数式实现，并且有一章专门讨论函数式程序的形式论证。

本书可作为高等院校计算机专业相关课程的教材，也适合广大程序设计人员参考。

作者介绍:

目录: 第1章 Standard ML 1

函数式程序设计 2

1.1 表达式和命令 2

1.2 过程式程序设计语言中的表达式 3

1.3 存储管理 3

1.4 函数式语言的元素 4

1.5 函数式程序设计的效率 7

Standard ML概述 8

1.6 Standard ML的演化 8

1.7 ML的自动定理证明传统 9

1.8 新标准库 10

1.9 ML和工作中的程序员 11

第2章 名字、函数和类型 13

本章提要 13

值的声明 14

2.1 命名常量 14

2.2 声明函数 15

2.3 Standard ML中的标识符 16

数、字符串和真值 17

2.4 算术运算 17

2.5 字符串和字符 19

2.6 真值和条件表达式 20

序偶、元组和记录 21

2.7 向量：序偶的例子 21

2.8 多参数和多结果的函数 22

2.9 记录 24

2.10 中缀操作符 27

表达式的求值 29

2.11 ML中的求值：传值调用 29

2.12 传值调用下的递归函数 30

2.13 传需调用或惰性求值 33

书写递归函数 36

2.14 整数次幂 36

2.15 斐波那契数列 37

- 2.16 整数平方根 39
- 局部声明 39
- 2.17 例子：实数平方根 40
- 2.18 使用local来隐藏声明 41
- 2.19 联立声明 42
- 模块系统初步 44
- 2.20 复数 44
- 2.21 结构 45
- 2.22 签名 46
- 多态类型检测 47
- 2.23 类型推导 47
- 2.24 多态函数声明 48
- 要点小结 50
- 第3章 表 51
- 本章提要 51
- 表的简介 51
- 3.1 表的构造 52
- 3.2 表的操作 53
- 基本的表函数 54
- 3.3 表的测试和分解 55
- 3.4 与数量有关的表处理 56
- 3.5 追加和翻转 58
- 3.6 表的表，序偶的表 60
- 表的应用 61
- 3.7 找零钱 61
- 3.8 二进制算术 63
- 3.9 矩阵的转置 64
- 3.10 矩阵乘法 66
- 3.11 高斯消元法 67
- 3.12 分解一个数为两个平方数之和 70
- 3.13 求后继排列的问题 71
- 多态函数中的相等测试 72
- 3.14 相等类型 73
- 3.15 多态集合操作 73
- 3.16 关联表 76
- 3.17 图的算法 77
- 排序：案例研究 81
- 3.18 随机数 81
- 3.19 插入排序 82
- 3.20 快速排序 83
- 3.21 合并排序 84
- 多项式算术 86
- 3.22 表示抽象数据 87
- 3.23 多项式的表示 87
- 3.24 多项式加法和乘法 88
- 3.25 最大公因式 90
- 要点小结 91
- 第4章 树和具体数据 93
- 本章提要 93
- 数据类型声明 93
- 4.1 国王和他的臣民 94
- 4.2 枚举类型 95
- 4.3 多态数据类型 97
- 4.4 通过val、as、case进行模式匹配 99

- 异常 101
  - 4.5 异常初步 101
  - 4.6 声明异常 102
  - 4.7 抛出异常 103
  - 4.8 处理异常 105
  - 4.9 对异常的异议 106
- 树 107
  - 4.10 二叉树类型 107
  - 4.11 枚举树的内容 109
  - 4.12 由表建立树 111
  - 4.13 为二叉树设计的结构 112
- 基于树的数据结构 112
  - 4.14 字典 113
  - 4.15 函数式数组和弹性数组 116
  - 4.16 优先队列 120
- 重言式检测器 124
  - 4.17 命题逻辑 124
  - 4.18 否定范式 126
  - 4.19 合取范式 127
- 要点小结 129
- 第5章 函数和无穷数据 131
  - 本章提要 131
  - 作为值的函数 131
    - 5.1 使用fn记法的匿名函数 132
    - 5.2 柯里函数 132
    - 5.3 数据结构中的函数 135
    - 5.4 作为参数和结果的函数 135
  - 通用算子 137
    - 5.5 切片 137
    - 5.6 组合子 138
    - 5.7 表算子map（映射）和filter（过滤） 139
    - 5.8 表算子takewhile和dropwhile 141
    - 5.9 表算子exists（存在）和all（全称） 141
    - 5.10 表算子foldl（左折叠）和foldr（右折叠） 142
    - 5.11 更多递归算子的例子 144
  - 序列，或无穷表 147
    - 5.12 序列类型 147
    - 5.13 基本的序列处理 149
    - 5.14 基本的序列应用 151
    - 5.15 数值计算 153
    - 5.16 交替和序列的序列 155
  - 搜索策略和无穷表 156
    - 5.17 用ML实现的搜索策略 158
    - 5.18 生成回文 159
    - 5.19 八皇后问题 160
    - 5.20 迭代深化 161
  - 要点小结 162
- 第6章 函数式程序的论证 163
  - 本章提要 163
  - 一些数学证明的原理 163
    - 6.1 ML程序和数学 164
    - 6.2 数学归纳法和完全归纳法 165
    - 6.3 程序验证的简单例子 168
  - 结构归纳法 171

- 6.4 关于表的结构归纳法 171
- 6.5 关于树的结构归纳法 175
- 6.6 函数值和算子 178
- 一般性归纳原理 181
- 6.7 计算范式 182
- 6.8 良基归纳和递归 185
- 6.9 递归程序模式 187
- 描述和验证 189
- 6.10 有序谓词 190
- 6.11 通过多重集合表示重新排列 191
- 6.12 验证的意义 194
- 要点小结 195
- 第7章 抽象类型和函子 197
- 本章提要 197
- 队列的三种表示方法 198
- 7.1 将队列表示为表 198
- 7.2 将队列表示为新的数据类型 199
- 7.3 将队列表示为表的序偶 200
- 签名和抽象 201
- 7.4 队列应具有签名 202
- 7.5 签名约束 202
- 7.6 抽象类型 (abstype) 声明 204
- 7.7 从结构导出的签名 206
- 函子 207
- 7.8 测试多个队列结构 208
- 7.9 泛型矩阵运算 210
- 7.10 泛型的字典和优先队列 214
- 利用模块建立大型系统 217
- 7.11 多参数函子 217
- 7.12 共享约束 221
- 7.13 全函子式程序设计 224
- 7.14 open声明 228
- 7.15 签名和子结构 232
- 模块参考指南 234
- 7.16 签名和结构的语法 235
- 7.17 模块声明的语法 237
- 要点小结 237
- 第8章 ML中的命令式程序设计 239
- 本章提要 239
- 引用类型 239
- 8.1 引用及其操作 240
- 8.2 控制结构 242
- 8.3 多态引用 245
- 数据结构中的引用 249
- 8.4 序列，或惰性表 249
- 8.5 环形缓冲区 252
- 8.6 可变更的数组和函数式的数组 255
- 输入和输出 259
- 8.7 字符串处理 259
- 8.8 文本输入输出 262
- 8.9 文本处理的例子 264
- 8.10 美化打印程序 267
- 要点小结 271
- 第9章 书写 $\lambda$ -演算的解释器 273

本章提要 273  
函数式语法分析器 273  
9.1 扫描或词法分析 273  
9.2 自顶向下的语法分析套件 275  
9.3 语法分析器的ML代码 277  
9.4 例子：分析和显示类型 280  
l-演算简介 284  
9.5 l-项和l-归约 284  
9.6 在替换中防止变量的捕获 286  
在ML中表示l-项 288  
9.7 基本操作 288  
9.8 l-项的语法分析 290  
9.9 显示l-项 291  
作为程序设计语言的l-演算 293  
9.10 l-演算中的数据结构 293  
9.11 l-演算中的递归定义 296  
9.12 l-项的求值 296  
9.13 演示求值程序 299  
要点小结 301  
第10章 策略定理证明机 303  
本章提要 303  
一阶逻辑的相继式演算 303  
10.1 命题逻辑的相继式演算 304  
10.2 证明相继式演算中的定理 305  
10.3 量词的相继式规则 307  
10.4 带量词的定理证明 308  
在ML中处理项和公式 310  
10.5 表示项和公式 310  
10.6 分析和显示公式 312  
10.7 合一 316  
策略和证明状态 319  
10.8 证明状态 319  
10.9 ML签名 320  
10.10 用于基本相继式的策略 321  
10.11 命题策略 323  
10.12 量词策略 324  
搜索证明 326  
10.13 变换证明状态的命令 326  
10.14 两个使用策略的证明实例 328  
10.15 策略算子 330  
10.16 一阶逻辑的自动策略 333  
要点小结 336  
项目建议 337  
参考文献 339  
Standard ML语法图 347  
语法图中英词汇对照表 357  
索引 359  
预定义标识符 367  
• • • • • ([收起](#))

[ML程序设计教程\\_下载链接1](#)

## 标签

ML

函数式编程

FP

计算机科学

编程

计算机

编程语言

程序设计

## 评论

截至收藏本书前，书架上只有两本计算机书是被我反复研读过的：SICP和这本。其实何必呢，编程只是乐趣，我没想过做程序员啊

-----  
放弃了，看不下去了，连个sml的编译器都没编译出来

-----  
想念柯老师，程序入门

-----  
看完这个喜欢的Ocaml

-----  
我觉得比sicp更适合入门

-----  
很小众的语言。。。

-----  
国内不多的fp书籍，二手才买到的绝版啊...

-----  
翻得比较一般

-----  
新年的第一本技术书。大概两年前打算通过阅读这本书入门函数式编程，但是看完此书才发现已经潜移默化地学会了很多，所以看起来也尤其地快（当然也跳过了很多习题...）。

-----  
巨好的一本书, 内容广, 代码翔实, 还有2.5章节太难看不懂

-----  
非常棒的函数式入门教材！

-----  
这本书拖了很久，终于看完了。这本书讲的是SML的函数式编程，不过说到底函数式语言最不可替代最大的优势的地方还是做程序证明，这本书讲的很细甚至罗嗦，不过最有意思的地方还是最后一章，简单讲述了证明定理机的组成和设计。

-----  
很有意思的书，翻过一遍了，还得再多翻几遍，想买一本纸质的，但是已经不卖了？

-----  
[ML程序设计教程\\_下载链接1](#)



## 书评

ML意味着meta language, 本书是学习ML排名第一的课本. 英文标题信息是这样的:  
PAULSON, LAWRENCE C. (Univ. of Cambridge, Cambridge, UK) ML for the working  
programmer (2nd ed.). Cambridge University Press, New York, NY, 1996, 478 pp., \$32.95,  
ISBN 0-521-56543-X....

-----  
这本书适合没有接触过functional programming的同学，也适合没有学过编程的同学。作者显然不满足于写一个语言教程，而是着重于灌输fp知识。  
所以在我看来这本书的废话稍微多了些。好几次我迅速的向后跳，但有意思的是每次我都被迫backtracing。因为他经常引用之前的例子和作业...

-----  
建议先看SICP，再看这本书，首先LISP语法比较简单，其次这本书会经常拿ML跟LISP做对比。  
SICP在大的方向上比较清晰，章节安排上更注重思想的延伸；而这本书的确如书名一样，ML的教程，从简单的类型，表，树到匿名函数无穷表，抽象类型，章节安排完全是学习语言的顺序。这样造成...

-----  
如果以前没有接触过FP，比如彻底的C/汇编程序员，看这本书能慢慢建立一些不同的编程模式；  
如果已经对haskell或者其他FPL有一些了解，看这本书可能会嫌啰嗦了，可以去看看<>，内容不算很完整，不过要点都到了。

-----  
[ML程序设计教程\\_下载链接1](#)