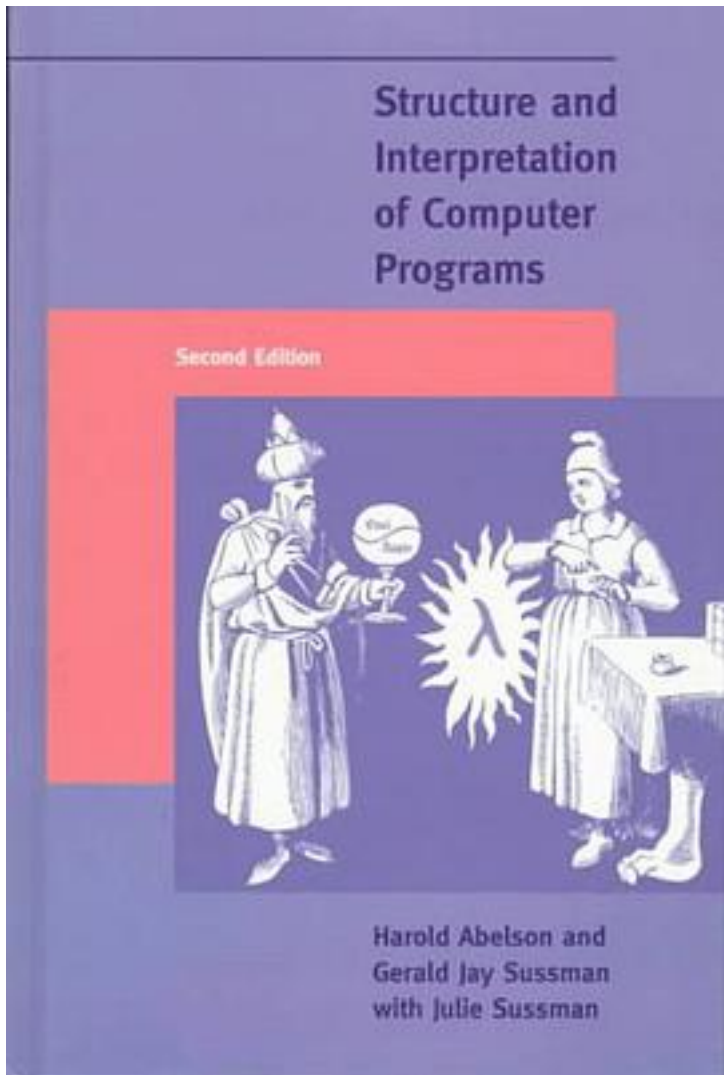


Structure and Interpretation of Computer Programs, Second Edition



[Structure and Interpretation of Computer Programs, Second Edition_下载链接1](#)

著者:Harold Abelson

出版者:McGraw-Hill Science/Engineering/Math

出版时间:1996-8-1

装帧:Hardcover

isbn:9780070004849

With an analytical and rigorous approach to problem solving and programming techniques, this book is oriented toward engineering. "Structure and Interpretation of Computer Programs" emphasizes the central role played by different approaches to dealing with time in computational models. Its unique approach makes it appropriate for an introduction to computer science courses, as well as programming languages and program design.

作者介绍:

Harold Abelson是MIT1992年度MacVicarFacultyFellow。Gerald JaySussman是Matsushita电子工程教授。他们都在MIT电子工程和计算机科学系工作。都得到过最重要的计算机科学教育奖：如Abelson得到了IEEE计算机学会的Booth奖。Sussman得到了ACM的Karlstrom奖。

Julie Sussman是作家和编辑，同时使用自然语言和计算机语言写作。

目录:

[Structure and Interpretation of Computer Programs, Second Edition 下载链接1](#)

标签

计算机

计算机科学

技术

成长

2010

评论

自己没有写个编译器，遗憾。

[Structure and Interpretation of Computer Programs, Second Edition_下载链接1](#)

书评

这个学期花了大量的时间在这本书上，同时旁听了裘宗燕老师（本书译者）以本书为教材开的课“程序设计技术和方法”，到学期末，总算是把这本书看完了。
豆瓣上关于本书的评论大多是形而上的，都说这本书怎样怎样好，但很少说明为什么好。我读这本书之前看了很多豆瓣上的评论， ...

「先说几个八卦」 -

本书曾经是MIT本科第一门课的教材。前两年被Python取代，在geek中引发了轩然大波。有兴趣可以Google一下[sicp mit python]。 -
本书在Amazon上的评分严重两极分化，五星（>90）和一星（>50）为主，彻底反正态分布。 - 本书在Amazon上排名最高的书评出自Pet...

（先得报一下进度：看完了前三章，前两章的习题大部分都做了，不过第三章就偷懒几乎没做）

如果要问现代数学最重要的概念是什么，那毫无疑问就是函数了，或者更确切地说，是映射。泛函这个词，或许对非数学系的同学来说有些陌生，但如果写成英语 functional，看起来就眼熟多了...

<http://swiss.csail.mit.edu/classes/6.001/abelson-sussman-lectures/>

1.
涵盖面很广。从数据抽象、过程抽象、迭代、高阶函数等编程和控制系统复杂性的思想，到数据结构和算法，到编译器/解释器、编程语言设计。MIT这门课的课程讲义（在MIT OCW里可找到）里还增加了面向对象编程的内容。虽然很多内容涉及并不深入，但是这是MIT EECS（电子工程与计...

<http://www.cppblog.com/cuigang/archive/2008/06/27/44801.html> 我为什么推荐

SICP? 向大家推荐

SICP, 不知道有多少人看了, 也不知道有多少人明白了, 更不知道有多少人惊叹了。或者你根本不屑一顾, 或者你看见 Lisp 那层层括号心生畏惧, 又或者你了了一瞥, 觉得没什么精彩...

我在豆瓣上第一本力荐的书。本来是打算留给TAOCP的, 可惜天资不够, 一直无缘窥得天书真髓。好在SICP同样对得起“力荐”二字。

有人说看这本书主要看第四第五章。这不是一件容易的事情, 尤其对于我们这个计算机教育落后的国家。好在还有平易近人的一二三章, 基本上可以在网上找...

曾经发表过对此书的简短评价, 如果说国内科班的计算机课程可以磨灭一个程序员真正的天性, 那么这本书无疑是我们的救星。

建议可以和OCW一起学习本书, 可以亲身领略Hal和Gerry两位大牛的课程是个很不错的经历。(由于Youtube被封, 建议翻墙) 该书从计算机语言的本质讲起, 通过L...

这是一本很有趣的书, 任何对编程真正感兴趣的人都应该看看。它讲了程序结构的很多方面, 但始终围绕着一个主题, 那就是从各个层次上来减少计算的复杂度。这和我读过的另外几本书核心是一样的, 只是维度不同。比如《代码大全》厚厚的一本书讲的也是管理复杂度 (<http://book.douba...>)

Underlying our approach to this subject is our conviction that 'computer science' is not a science and that its significance has little to do with computers. The computer revolution is a revolution in the way we think and in the way we express what we think...

读书时隔一年, 我才发现这本书在讲什么: 一切都是浮云, 编程范式才是王道
这本书几乎隐性的把主流和非主流的编程范式都介绍个遍:
过程式, 泛型式, 函数式, 元语言, 逻辑式, 对象式, 并发式。。。 ...

这本书提到的很多次的词就是abstraction: 对于函数进行抽象, 对于数据进行抽象, 这种抽象能力其实是非常重要的。阅读代码时的抽象
在学好编程之前总是对于所有函数的所有实现都感兴趣, 碰到一个大型的项目就恨不得将所有函数都弄明白, 但是这种方法其实很不明智, 在开发大...

最初买这本书是冲着学习函数式编程来的，但读完本书以后才发现原来作者的用意并不在介绍函数式编程，而在于介绍编程的本质。
我身边绝大多数的程序员都是用谭浩强老师的那本《C程序设计语言》入门的。如果把C比作剑宗，那么Lisp则是气宗，二者并不矛盾，但初学者却容易先入为主...

SICP 讲了什么, 没讲什么, 以及带来了什么 ----- ## SICP 讲了什么? 前三章是对 Scheme 的一个入门, 阐述了 Scheme 和 FP 通常的用法.
第一章过程抽象包含通常语言入门要介绍的东西: 函数和变量的定义 (define), 控制结构 (if, cond), 递归, 一阶函数 (随之而来的 lambda); ...

如果你觉得这本书旧了点的话，我推荐下UC伯克利最近的课程cs61a:Structure and Interpretation of Computer Programs。
<http://www-inst.eecs.berkeley.edu/~cs61a/su12/>
伯克利在这门课程上已经开始用python了。 “These fundamental ideas have long been taught at Berkel...

第四章scheme做个小小的interpreter, 學過compiler的人一定覺得還不如看dragon book比較實際.
但是他精彩就是在於用很簡潔的方式(scheme的特性)去解釋複雜的interpreter原理.
但是interpreter不是最重要的,
精彩的還是看他如何在精心設計之下可以順利的拓展他需要的功能. 又...

上帝是魔法師，魔法師是無神論者。 聖經灌輸教義，魔法書揭示魔力。
這本書使胡言亂語的癡迷者硬起來。 這本書讓魔法師重拾一切元素盡在掌握的驕傲。
這本書寫給那些制定規則的人。

hi gears,
这个礼拜把sicp的第一章之前的部分读了3遍,发现有一些其实很明显的隐喻在去年的hacking时都没怎么注意,2009年大多去关注技术细节问题了,作为hacker这种低级错误简直不应该犯,其实计算本质的hacking方向Alan在foreward就已经讲的很清楚了,前面的隐喻都很简单,后面...

Brian Harvey在cs61a
2011spring第一节课程的开场白是：“这里是cs61a，世界上最好的计算机课。”接着他又说道：“并不是因为我，而是因为我们用的教材——《计算机程序的构造与解释》（

SICP)。“即使这样也无法掩盖Brian Harvey作为一名特立独行的教授。他喜欢西兰花和披头士， ...

很早就听说过这部书，一直到最近才看完前三章，实在惭愧。
书名叫《计算机程序的构造和解释》，显然内容分为两块：1) 程序的构造，其实就是抽象；2) 程序的解释。讲程序构造，离不开编程语言，本书采用 Lisp（这本书做了 MIT 20 几年的教程，最近几个学期把这门课程替换了， ...

[Structure and Interpretation of Computer Programs, Second Edition_下载链接1](#)