

《计算的本质》

图书基本信息

书名：《计算的本质》

13位ISBN编号：9787115361541

出版时间：2014-11

作者：[英] Tom Stuart

页数：300

译者：张伟

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《计算的本质》

内容概要

《计算的本质：深入剖析程序和计算机》借助Ruby全面介绍计算理论和编程语言的设计。作者注重实用性，不仅尽量抛开复杂难懂的数学符号，而且特别选用简单快捷的编程语言Ruby，在读者熟知的背景知识下，以明晰的可工作代码阐明形式语义、自动机理论，以及通过lambda演算进行函数式编程等计算机科学知识，并为让其自行探索做足准备。

本书适合计算机科学系学生，以及熟知现代编程语言，想要系统地学习计算机科学知识的程序员、软件工程师阅读参考。

《计算的本质》

作者简介

Tom Stuart

Tom Stuart 伦敦数字产品咨询公司Codon的创始人、计算机科学家、程序员，擅长Ruby、Rails、Web应用、用户体验、面向对象设计和行为驱动开发。另外，作为顾问、导师和培训师，他经常通过网络帮助各家公司高质高效地创建软件产品。他还曾在剑桥大学做编译器优化方面的演讲，与人联合组织过Ruby大会（Ruby Manor），而且是伦敦Ruby用户组的成员。

书籍目录

封面介绍	X
前言	XI
第1章 刚好够用的Ruby基础	1
1.1 交互式Ruby Shell	1
1.2 值	2
1.2.1 基本数据	2
1.2.2 数据结构	3
1.2.3 proc	4
1.3 控制流	4
1.4 对象和方法	5
1.5 类和模块	6
1.6 其他特性	7
1.6.1 局部变量和赋值	7
1.6.2 字符串插值	8
1.6.3 检查对象	8
1.6.4 打印字符串	8
1.6.5 可变参数方法 (variadic method)	9
1.6.6 代码块	9
1.6.7 枚举类型	10
1.6.8 结构体	11
1.6.9 给内置对象扩展方法 (Monkey Patching)	12
1.6.10 定义常量	13
1.6.11 删除常量	13
第一部分 程序和机器	
第2章 程序的含义	17
2.1 “含义”的含义	18
2.2 语法	19
2.3 操作语义	19
2.3.1 小步语义	20
2.3.2 大步语义	40
2.4 指称语义	46
2.4.1 表达式	46
2.4.2 语句	49
2.4.3 应用	51
2.5 形式化语义实践	52
2.5.1 形式化	52
2.5.2 找到含义	53
2.5.3 备选方案	53
2.6 实现语法解析器	54
第3章 最简单的计算机	59
3.1 确定性有限自动机	59
3.1.1 状态、规则和输入	60
3.1.2 输出	60
3.1.3 确定性	61
3.1.4 模拟	62
3.2 非确定性有限自动机	65
3.2.1 非确定性	65

3.2.2	自由移动 (free move)	71
3.3	正则表达式	74
3.3.1	语法	75
3.3.2	语义	78
3.3.3	解析	86
3.4	等价性	88
第4章	增加计算能力	97
4.1	确定性下推自动机	100
4.1.1	存储	100
4.1.2	规则	101
4.1.3	确定性	103
4.1.4	模拟	103
4.2	非确定性下推自动机	110
4.2.1	模拟	113
4.2.2	不等价	115
4.3	使用下推自动机进行分析	116
4.3.1	词法分析	116
4.3.2	语法分析	118
4.3.3	实践性	122
4.4	有多少能力	123
第5章	终极机器	125
5.1	确定型图灵机	125
5.1.1	存储	126
5.1.2	规则	127
5.1.3	确定性	131
5.1.4	模拟	131
5.2	非确定型图灵机	136
5.3	最大能力	137
5.3.1	内部存储	137
5.3.2	子例程	140
5.3.3	多纸带	141
5.3.4	多维纸带	142
5.4	通用机器	142
5.4.1	编码	144
5.4.2	模拟	145
第二部分	计算与可计算性	
第6章	从零开始编程	149
6.1	模拟lambda演算	150
6.1.1	使用proc工作	150
6.1.2	问题	152
6.1.3	数字	153
6.1.4	布尔值	156
6.1.5	谓词	160
6.1.6	有序对	161
6.1.7	数值运算	161
6.1.8	列表	168
6.1.9	字符串	172
6.1.10	解决方案	174
6.1.11	高级编程技术	178

6.2	实现lambda演算	184
6.2.1	语法	184
6.2.2	语义	186
6.2.3	语法分析	191
第7章	通用性无处不在	193
7.1	lambda演算	193
7.2	部分递归函数	196
7.3	SKI组合子演算	201
7.4	约塔 (Iota)	210
7.5	标签系统	213
7.6	循环标签系统	220
7.7	Conway的生命游戏	229
7.8	rule 110	231
7.9	Wolfram的2,3图灵机	234
第8章	不可能的程序	235
8.1	基本事实	236
8.1.1	能执行算法的通用系统	236
8.1.2	能够替代图灵机的程序	239
8.1.3	代码即数据	239
8.1.4	可以永远循环的通用系统	241
8.1.5	能引用自身的程序	245
8.2	可判定性	250
8.3	停机问题	251
8.3.1	构建停机检查器	251
8.3.2	永远不会有结果	254
8.4	其他不可判定的问题	258
8.5	令人沮丧的暗示	260
8.6	发生上述情况的原因	261
8.7	处理不可计算性	262
第9章	在“玩偶国”中编程	265
9.1	抽象解释	266
9.1.1	路线规划	266
9.1.2	抽象：乘法的符号	267
9.1.3	安全和近似：增加符号	270
9.2	静态语义	274
9.2.1	实现	275
9.2.2	好处和限制	281
9.3	应用	284
后记		285

《计算的本质》

精彩短评

- 1、可以跑的ruby代码，好兴奋呀！<https://github.com/tomstuart/computationbook>
- 2、书本身还是不错的，看得出作者的用心，但是定位就有点尴尬，因为理论部分的缺失所以没有办法学到很多真正有用的知识，如果算作科普书籍的话难度又有些大，没有基础很容易看的似懂非懂。归根到底是计算理论这部分知识太抽象太难了吧，不过如作者所说的用来“满足自己的好奇心”还是足够有趣的。
- 3、SICP
- 4、#图灵PDF# 4.3分。有点像大三大四两门外教课的课本欸！可爱也可爱在 Ruby 实现，啰嗦也啰嗦在这儿。如果能够压缩成 150 来页的小品，就更可爱啦！还蛮有趣的~
- 5、蛮适合用于入门的；
- 6、2016年最有收获的一本技术书，从有限状态机到下推自动机，再到终极机器图灵机，能力一步步增强。中间介绍了一些图灵完备的系统，但是图灵机也有做不到的事情，也就是判断一个系统或者一段程序做了什么，最后通过介绍几个案例来弥补这些不可能做到的事情。
- 7、练习到了第六章，后面三章大概看了下。
- 8、抽象描述计算机物体和人之间关系。机器，语言，编程。以及结果的预判，可能与不可能。处理对象中方法，消息类，模块，数据，语法，语义。
- 9、对编译原理和计算机组成原理解释，思想新颖
- 10、循序渐进而又启发性很强的一本书
- 11、计算理论~
- 12、《计算的本质》以Ruby程序为工具讲解了计算机、程序、程序语言等基本的计算机科学问题，是一本非常好的计算机科学实践书籍。使用Ruby语言模拟了各种机器的计算能力、计算机语言的语义，深入浅出的讨论了计算的本质。书中详细探讨了各种类型机器的计算能力，机器按照计算能力分为有限自动机（FA），下推自动机（PDA），图灵机（TM），通用计算机；除图灵机与通用计算机等价外，计算能力逐渐增强。有限状态机增加外部存储的栈，扩大计算能力就是下推自动机；有限状态机增加无限长的纸带访问就是图灵机，图灵机的实质就是能够访问无限长纸带的有限状态机。DFA可以识别正则式，NPDA可以识别回文字符串，确定性图灵机(DTM)可以进行二进制递增运算。每种机器都有相当明显的能力限制，FA无法解决涉及无限制的计数问题，例如判
- 13、你竟然用ruby!!于是我没看下去
- 14、难得的好书，简明有趣。
- 15、这本书的读者对象是“没有正规学习过数学或者计算机科学的朋友”，然而根据我的目前的阅读经验来看，没有基础的人是完全不可能看得懂这本书的。因为这本书主要以实战为主，理论讲的超级烂的，内容编排也不够到位.....看不懂的建议先补理论吧。
- 16、必须是力荐，讲解通俗易懂，非常平易，可以让人轻松学习，里面的资料也比较丰富，我很喜欢。
- 17、非常精彩，这么薄的一本书，覆盖了图灵机、lambda演算、SKI 组合子等计算理论，并且提供了可供运行和测试的代码，还有啥不知足的呢？略微可惜的可能是因为篇幅或者作者故意限制，没有特别深入。
- 18、从有限状态机到图灵机，全书在引导读者剖析理论层面的计算机，lambda算法激动人心，就是Ruby语法有点琐碎。遗憾就是看的半懂。
- 19、习惯了站在巨人的肩膀上,偶尔也要往下看
- 20、在不到三百页的篇幅里，把自动机，图灵机，lambda过了一遍，而且还附了实现及测试代码。唯一不足的就是用工程的态度去讲解科学，会缺乏更细致的理解。看来还得先看看几本导论性质的计算理论的书，再来刷一遍，相信理解会深入一些。
- 21、满足好奇心
- 22、SICP风格
- 23、Ruby版编译原理
- 24、这半年读过的最漂亮的一本书，各种计算模型及某些通用计算模型的等价性、语义分析/语法分析

《计算的本质》

以及最后一章的类型检查系统（给我感觉就是在写一个小的编译器或者解释器）、还有停机问题的具体介绍都非常精彩。最最重要的是，所有的论述都用上了 Ruby 代码，大部分例子理论上都是可以运行的！Ruby 的灵活性（比如代码即数据在某些语言中就不是很方便）给本书某些章节写实现时带来不少便利，第八章作者甚至一本正经胡说八道地用 evaluate function 写了个停机问题的函数，然后想尝试解决哥德巴赫猜想。

笔记过几天附上。

25、请教：谁能概括一下，这本书揭示计算的本质是什么？

26、1星不是说这本书写的不好,是我看不懂

27、知识储备不足，仅看懂了跟正则表达式相关的一部分。

28、看了这本书，对学习计算理论导引有很大的帮助，这本书可以看做是计算理论导引的实战版本。

29、编程要思考这样几个关键的概念：时间（算法和流），空间（数据结构和存储），不确定性（大数分解和程序的处理未知结果），无限（堆栈和递归）。其实康德的《纯粹理性批判》不就是研究这几个概念么？

30、前半部分不错，后半部分的形式语言讲的太浅了

31、读了一半，还是比较有意思的，但是基础不好看起来还是要用心，放下了就没有读完。

32、可惜是 ruby 描述的

33、Ruby版SICP

《计算的本质》

精彩书评

1、非科班，ruby零基础，初学爱好者的一番胡说八道1.我不关心书上的代码是ruby实现，或是JS实现、或是Python实现，我只关心这些奇怪的符文能不能在我的机器上跑起来，何况我又不靠这本书学ruby、JS、或者python。代码效果相当好，搭配本书托管在Github上的代码[1]使用，简直愉悦！2.我也不关心世界上存在着的其余那些神书、奇书、动物书的奇迹效果，我只关心这本书对我而言是不是可理解的输入(comprehensive input)，作者直接给我看代码(show me code)而且还是循序渐进地(show me code)，如下：“首先怎么做，为什么我们要这样做”>>代码>>>“然后怎么做，为什么要这样做”>>代码>>“写好了，看看效果”>>代码实力写代码，实力教怎么看代码，实在很佩服这个作者！3.DFA NFA RegularExpression DPDA DTM lambda HaltingProblem TagSystem 满足好奇兴奋之余也倍感踏实，还不是因为有可以运行的代码！[1] 本书配套代码 [githubhttps://github.com/tomstuart/computationbook](https://github.com/tomstuart/computationbook)

章节试读

1、《计算的本质》的笔记-第127页

“纸带头”，译者指出原文是 tape head。

这里指的是读写头，不是纸带的最初部分。
记下来我自己备忘，也提醒后来者，避免误读。

2、《计算的本质》的笔记-第193页

八卦一下

书中提到了几种与图灵机等价的模型，lambda算子、部分递归函数、SKI组合子、Iota、标签系统、循环标签系统、Conway生命游戏、rule 110、Wolfram的2,3图灵机。

部分递归函数，可能就是维基百科在递归函数词条中提到的“在可计算性理论中证明了递归函数精确的是图灵机的可计算函数。”

Wolfram，就是著名的Mathematica的发明者之一。他的另一个成果是 Wolfram Alpha 计算型知识引擎。

数理逻辑学家、哲学家、人工智能先驱王浩（1921~1995），我记得也发表过一个图灵机等价模型，似乎是翻纸牌的，可能叫做Tile系统。可惜似乎由于有位法官同名，在维基百科上的王浩词条不能浏览，而百度百科语焉不详。

王浩研究自动定理证明，用早期的IBM计算机，几分钟就证明了罗素花十年心血才在其名著《数学原理》中证明的220条命题。王浩是历史学家何兆武先生的同学和好朋友，他俩都研究过哥德尔-艾舍尔-巴赫这本书。记得何兆武先生有张照片，背景正是书架，书名都被遮挡起来，除了这本很厚的哥德尔-艾舍尔-巴赫。这本书讨论的内容就包括图灵机，也就是人的计算能力的限制。他们同样也都对罗素感兴趣，刚刚提到王浩证明了数学原理中的命题，何兆武先生则翻译了罗素的《西方史学史》上卷，对罗素的史学观也有专门论述。

王浩的工作，维基百科英文上查到了，从图灵机英文词条找过去，在[http://en.wikipedia.org/wiki/Wang_B-machine]提到：

Hao Wang (1957), A Variant to Turing's Theory of Computing Machines, JACM (Journal of the Association for Computing Machinery) 4; 63-92. Presented at the meeting of the Association, June 23-25, 1954.

3、《计算的本质》的笔记-第138页

磁带，应作纸带。

4、《计算的本质》的笔记-第163页

让INCREMENT工作的关键...由ZERO组成的有序对调用slide n次，然后使用SELECT从结果的有序对中拉出左边的数来。

slide操作像这样：

>> slide ([3,4])

=> [4,5]

《计算的本质》

>> slide ([8, 9])
=> [9, 10]

我不记得SICP如何实现减法了。
这一实现非常巧妙。

5、《计算的本质》的笔记-第138页

一台机器不是只用一个状态表示“向右扫描查找一个空白方格”，而是可以为“向右扫描一个空白方格（记住我之前读取到了一个a）”设置一个状态，再为“向右扫描一个空白方格（记住我之前读取到了一个b）”设置另一个状态，所有可能的字符都以此类推 - - 字符数目也是有限的，所以这样的复制总是有限的。

规则数量有限，非常重要。

如果无限，就只能递归或迭代，不能复制。

无限的情况，参见 Algorithmic Adventures [<http://book.douban.com/subject/4473922/>].

6、《计算的本质》的笔记-第141页

一条图灵机的纸带通过交叉存取

交叉存取应为: 读写（存取）字符X

如果我们在每一个交叉字符的边上放上一个空白方格

应为: 每一个字符

- - - -

原文中的实例，希望实现 adgbehcfi，用一条纸带实现三条纸带（abc, def, ghi）的效果。

用字符X标识三个纸带中每个读写头的位置:

a_dXg_b_e_hXcXf_i_

三个X分别标识出ab(c)，(d)ef，g(h)i三条纸带及读写头的位置（用括号表示）。

7、《计算的本质》的笔记-第134页

规则的语法

(当前状态，读入字符，下一状态，写入字符，读写头移动方向)

'_'表示空白。

8、《计算的本质》的笔记-第122页

lookahead 翻译为 递推，似乎是误译。

《计算的本质》

一般译为 前瞻 或 超前，“LL(1)或者LR(0)这样的命令的时候，括号里的数字就是要超前的tokens的数量。”

9、《计算的本质》的笔记-第167页

匿名lambda函数实现递归。

Y组合子是一种 不动点组合子。

不动点组合子 见

[<http://zh.wikipedia.org/wiki/%E4%B8%8D%E5%8A%A8%E7%82%B9%E7%BB%84%E5%90%88%E5%AD%90>]

不动点，这个概念似乎最初来自牛顿法。

10、《计算的本质》的笔记-第144页

注5:二元基于2，一元基于1.

猜测“基于”是误译。

应作:二元的基数是2，一元的基数是1.

原文中，把用三个不同的数字 1, 11, 111 编码a,b,c。

用特定字符1的个数不同来表示不同的被编码字符。
因为只使用1个字符，称为1元。类似于“一进制”。

11、《计算的本质》的笔记-第40页

似乎 大步语义等价于SICP中的正则序，递归，先展开，后求值；

小步语义等价于 应用序，迭代，先求值，后展开。

12、《计算的本质》的笔记-第89页

这两个例子的唯一真正区别是, DFA的模拟是从一个当前状态移动到另一个, 而NFA的模拟是从当前可能状态的集合移动到另一个可能状态的集合. 所以任何NFA都可以转化成等价的DFA, 只不过因为转化后的DFA的每个节点表示的是转化前NFA的节点的集合, 所以转化后DFA理论的最大节点数是 2^n (n =转化前NFA的节点数)

13、《计算的本质》的笔记-第1页

《计算的本质》以Ruby程序为工具讲解了计算机、程序、程序语言等基本的计算机科学问题，是一本非常好的计算机科学实践书籍。使用Ruby语言模拟了各种机器的计算能力、计算机语言的语义，深入浅出的讨论了计算的本质。

书中详细探讨了各种类型机器的计算能力，机器按照计算能力分为有限自动机（FA），下推自动机（PDA），图灵机（TM），通用计算机；除图灵机与通用计算机等价外，计算能力逐渐增强。有限状态机增加外部存储的栈，扩大计算能力就是下推自动机；有限状态机增加无限长的纸带访问就是图灵机，图灵机的实质就是能够访问无限长纸带的有限状态机。

DFA可以识别正则式，NPDA可以识别回文字符串，确定性图灵机(DTM)可以进行二进制递增运算。每种机器都有相当明显的的能力限制，FA无法解决涉及无限限制的计数问题，例如判定一个括号组成的字

《计算的本质》

字符串是否平衡的问题；PDA无法处理任何信息需要在多处重用的问题，例如判定一个字符串是否含有同样数目的字符a,b和c；最先进的机器就是图灵机，拥有无限制的存储，这个存储可以以任何顺序，在任意循环中，在任意条件语句中以及子例程中访问。

计算规则又可以区分为确定性，非确定性，例如确定性有限自动机，非确定性图灵机等，确定性就是状态转换规则集中没有冲突的规则，没有遗漏的规则；非确定性就是放松确定性的限制，允许一个状态和输入包含多条规则或无规则。

对用通用的计算性，图灵机等价于lambda演算，等价于部分递归函数计算，等价于SKI演算，这些演算、计算及图灵机都具有通用计算机的能力，几者是完全等价的。尽管图灵机等的计算能力非常强大，但是其能力也是有极限的，比如停机问题，判定程序是否输出特定字符串等问题，都是不可判定问题。判定“程序是否做了我想让他做的事情”这是不可判定的，例如程序是否一定崩溃，是否调用了私有的API，是否执行了从网上下载的任何代码。这些判定程序自身性质的问题都是不可判定的。

14、《计算的本质》的笔记-第132页

注1.

原译文 纸带是纯功能性的。

猜测原意是 纸带是纯函数式的。参见函数式语言，无付作用的。

15、《计算的本质》的笔记-第245页

不停机程序是通用性的一个不可避免的结果只要在使用一种强大到能对自身求值的程序，我们就知道一定可能使用#evaluate的等价物构建永不停机的程序如果移除了一个特性让一个程序无法永远循环，一定也不可能实现#evaluate了
哥德尔不完备定理的通俗解释。

《计算的本质》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com