

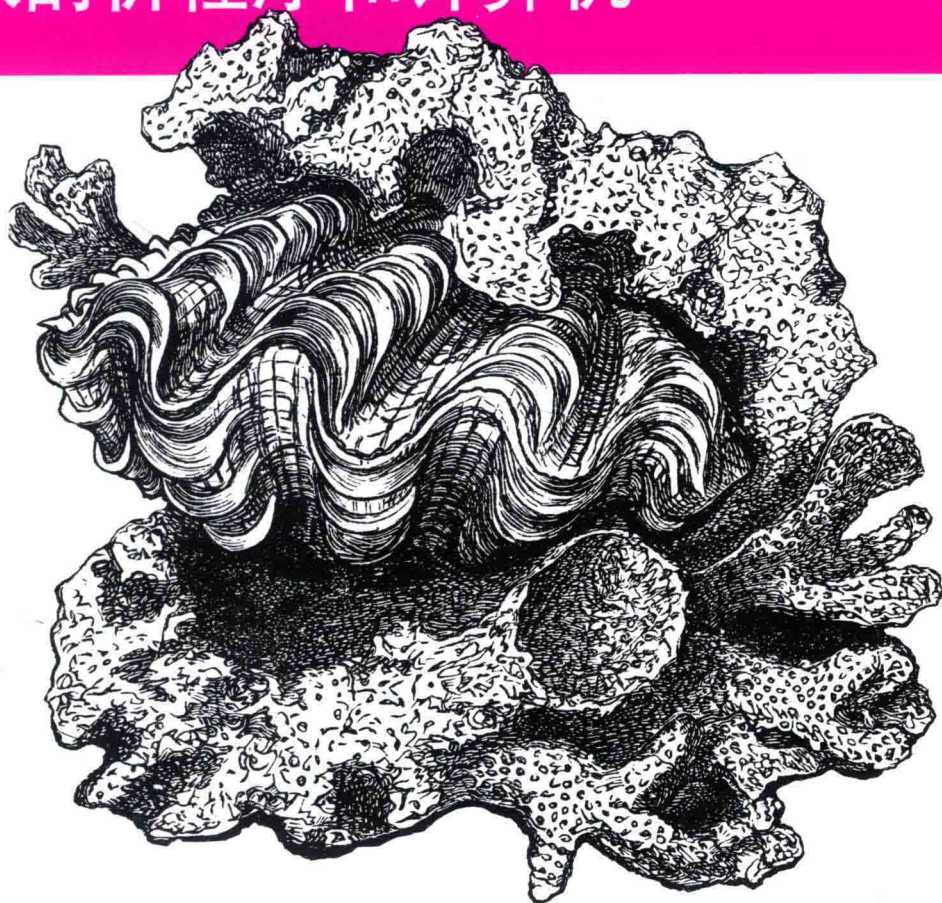
TURING

图灵程序设计丛书

*Understanding Computation:
From Simple Machines to Impossible Programs*

计算的本质

深入剖析程序和计算机



[英] Tom Stuart 著
张伟 译

O'REILLY®



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

计算的本质

深入剖析程序和计算机

Understanding Computation
From Simple Machines to Impossible Programs



[英] Tom Stuart 著

张伟 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

计算的本质 : 深入剖析程序和计算机 / (英) 斯图尔特 (Stuart, T.) 著 ; 张伟译. -- 北京 : 人民邮电出版社, 2014. 11

(图灵程序设计丛书)

ISBN 978-7-115-36154-7

I. ①计… II. ①斯… ②张… III. ①程序设计—基本知识 IV. ①TP311.1

中国版本图书馆CIP数据核字(2014)第135541号

内 容 提 要

本书借助简单的 Ruby 代码示例, 全面、深入地介绍计算理论和编程语言设计。作者注重实用性, 在读者熟知的背景知识下, 以明晰的可工作代码阐释了形式语义、自动机理论, 以及通过 lambda 演算进行函数式编程等计算问题, 并为读者自行探索打下了良好基础。

本书面向熟悉某种现代编程语言却非科班出身的程序员, 是一本帮你真正理解计算机科学和计算原理的优秀参考书。

-
- ◆ 著 [英] Tom Stuart
 - 译 张 伟
 - 责任编辑 李松峰 毛倩倩
 - 执行编辑 程 芑
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 18.75
 - 字数: 433千字 2014年11月第1版
 - 印数: 1-3 500册 2014年11月河北第1次印刷
 - 著作权合同登记号 图字: 01-2013-5148号
-

定价: 69.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

©2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版, 2013。

简体中文版由人民邮电出版社出版, 2014。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部内容不得以任何形式重制。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

封面介绍

本书封面上的动物是砗蚝（拉丁学名为 *Hippopus hippopus*）。砗蚝因其形状又叫马蹄蛤，又因其颜色泛红称为草莓蛤。砗蚝是砗磲科巨蛤亚科的一部分，而砗磲科又是乌蛤科的一部分。砗蚝主要生活在印度洋—太平洋区域的礁石中。

砗蚝有两个相同而对称的铰合部。它还有着深深的嵴和与众不同的红白图案。砗蚝待在一个地方使用虹管过滤周围的水之后，以周围的浮游生物为食。

前言

读者对象

本书写给那些对编程语言和计算理论充满好奇的程序员，特别是没有正规学习过数学或者计算机科学的朋友。

如果你对涉及程序、语言以及机器，且能开阔思维的计算机科学知识感兴趣，却被常常用于阐明它们的数学语言打击的话，那么本书恰恰是你需要的。我们抛开复杂的数学符号，用可工作的代码来描述理论性概念，并为大家自行探索做足准备。

本书读者至少要了解一种现代编程语言，如 Ruby、Python、JavaScript、Java 或者 C#。书中所有示例程序都采用 Ruby 语言编写而成，但了解其他语言的读者亦能看懂。注意，本书目标并不是展示 Ruby 或面向对象设计的最佳实践。本书代码意在简明清晰，但并不一定都容易维护，因为我们的目标是使用 Ruby 阐明计算机科学，而不是用计算机科学讲解 Ruby。本书亦非教材或者百科全书，所以并没有给出形式论证或者严密的证明，它试图让你能接近一些有趣的思想，启发你更深入地了解它们。

排版约定

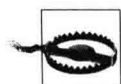
本书中使用以下排版约定。

- 楷体
用于标记新名词。
- 等宽字体 (constant width)
用于程序代码，在段落中用于表示程序的组成部分，如变量或函数名、数据库、数据类型、环境变量、语句、关键字。

- 等宽粗体 (*constant width bold*)
命令或是其他应该由用户输入的内容。
- 等宽斜体 (*constant width italic*)
应该由用户提供或由上下文确定的值。



提示、建议或一般注解会放在这里。



警告或警示信息会放在这里。

使用代码

本书旨在帮助读者解决实际问题。也许你需要在自己的程序或文档中用到本书中的代码，但除非大段大段地使用，否则不必与我们联系取得授权。因此，用本书中的几段代码写个程序不用向我们申请许可，但是销售或者分发 O'Reilly 图书随附的代码光盘则必须事先获得授权；引用书中的代码来回答问题也无需我们授权，而将大段的示例代码整合到自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处。出处一般要包含书名、作者、出版商和书号，例如：“*Understanding Computation by Tom Stuart* (O'Reilly). Copyright 2013 Tom Stuart, 978-1-4493-2927-3”。

如果还有其他使用代码的情形需要与我们沟通，可以随时与我们联系：permissions@oreilly.com。

Safari® Books Online



Safari Books Online (www.safaribooksonline.com) 是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM

Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://oreil.ly/Understanding_Computation

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

致谢

非常感谢 Go Free Range 的热情款待，他们在本书写作期间为我提供了办公地点、友好的交流以及茶水。没有他们慷慨的支持，我肯定早就放弃了。

感谢 James Adam、Paul Battley、James Cogan、Peter Fletcher、Chris Lowis 和 Murray Steele 对早期草稿的反馈，并且感谢 Gabriel Kerneis 和 Alex Stangl 的技术审校。有了他们的贡献，本书的质量大为提高。我还要对剑桥大学的 Alan Mycroft 提供资料和支持表示感谢。

O'Reilly 的许多人帮助保证了本书的顺利完工与出版，我要特别感谢 Mike Loukides 和 Simon St.Laurent 一开始就对这个想法抱有热情和信念，感谢 Nathan Jepson 对如何把想法转变成实际图书的建议，以及 Sanders Kleinfeld 对我要求完美的语法高亮这一严苛要求的迁就。

感谢我的父母让一个恼人的小孩有办法、有动机，也有机会在计算机上混日子。还要感谢 Leila，每次我忘记工作应该如何完成时，她都会耐心地提醒我把该死的词儿一个接一个地排起来。最后我做到了。

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

刚好够用的Ruby基础

本书中的代码全部使用 Ruby 写成。Ruby 是一种简单、友好而且有趣的编程语言。因为 Ruby 清晰与灵活，我选择了它，但本书并不依赖于 Ruby 专有的特性，所以这些示例代码均可转换成你喜欢的其他任何语言，特别是像 Python 或者 JavaScript 这样的动态语言，如果那样你更容易理解的话。

所有的示例代码都兼容 Ruby 2.0 和 Ruby 1.9。你可以在 Ruby 官方站点 (<http://www.ruby-lang.org/>) 详细了解 Ruby，还可以下载一份官方的实现。

我们会快速浏览一下 Ruby 的特性，并集中介绍本书中用到的部分。如果你想学习更多内容，推荐从 O'Reilly 的《Ruby 编程语言》(*The Ruby Programming Language*) 一书起步。



如果已经了解 Ruby，你完全可以从第 2 章开始阅读本书。

1.1 交互式 Ruby Shell

Ruby 最友好的一个特性就是交互式控制台 IRB，它可以让我们在输入 Ruby 代码后立即看到执行结果。本书将广泛使用 IRB 与所写的代码进行交互，并探索这些代码是如何工作的。

在开发机器的命令行中输入 `irb`，就可以运行 IRB 了。IRB 显示提示符 `>>` 时，表明当前可以输入一个 Ruby 表达式。输入一个表达式并敲回车键之后，代码执行，结果会显示到提示符 `=>` 之后：

```
$ irb --simple-prompt
>> 1 + 2
=> 3
>> 'hello world'.length
=> 11
```

本书中只要出现提示符 `>>` 和 `=>`，就是在与 IRB 交互。为了让长代码更易读，本书显示它们的时候会去掉提示符，但是仍然假定这些代码已经输入或者粘贴进了 IRB。所以一旦本书中有像下面这样的 Ruby 代码：

```
x = 2
y = 3
z = x + y
```

我们之后就可以在 IRB 中得到它们的结果：

```
>> x * y * z
=> 30
```

1.2 值

Ruby 是一种面向表达式的语言：每一段有效的代码执行之后都要产生一个值。下面快速浏览一下 Ruby 中不同类型的值。

1.2.1 基本数据

如我们所料，Ruby 支持布尔型（Boolean）、数值型（number）和字符串（string），且它们都支持常规运算：

```
>> (true && false) || true
=> true
>> (3 + 3) * (14 / 2)
=> 42
>> 'hello' + ' world'
=> "hello world"
>> 'hello world'.slice(6)
=> "w"
```

一个 Ruby 符号表示一个名字，是一个轻量级、不可变的值。作为字符串的简单化、非内存密集化（less memory-intensive）的替身，符号在 Ruby 中被广泛使用——通常是作为散列表中的键使用（参见 1.2.2 节）。符号字面量的开头会有一个冒号：

```
>> :my_symbol
=> :my_symbol
>> :my_symbol == :my_symbol
=> true
>> :my_symbol == :another_symbol
=> false
```