通信顺序进程

C.A.R Hoare 著

周巢尘 译

北京大学出版社

通信顺序进程

C.A.R.Hoare 著

周巢尘 译

北京大学出版社

通信顺序进程

C.A.R.Hoare 著

周巢尘 译

责任编辑：李采华

*

北京大学出版社出版

（北京大学校内）

北京大学印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

850×1168毫米 32开本 9.875 印张 250千字

1990年1月第一版 1990年1月第一次印刷

印数：0001—3,000册

ISBN 7-301-00813-9/TP·010

定价：3.95元

内 容 提 要

通信顺序进程一书，为计算机科学和数学的高年级大学生、研究生，亦为职业的系统设计人员、经理人员，介绍了用于并发性和通信研究的新的数学途径。这一新领域最适用于描述、设计和实施那些连续操作并不断和其环境交互作用的计算机系统。

在阐明这一理论的基础概念时，作者使用了大量有实际应用背景的，但被简化了的例子——从自动售货机到游戏博奕、通信协议及操作系统。Hoare教授基于一种统一的数学理论来处理这些例子，这种理论是由一组系统化的代数法则所刻画的。书中也解释了有关的数学知识。

本书内容已由非正式的讨论班和正式的课程成功地试用过。

本书作者，C.A.R.Hoare，是英国牛津大学计算科学教授，著名图林奖的获得者，伦敦皇家学会会员，不列颠计算机学会杰出会员。

向中国读者致意

非常感谢我的朋友和同事周果尘翻译了我的书；我也感谢他曾邀请我于一九八三年四月，在北京中国科学院研究生院举办了一次系列讲座；在那里，我非常高兴地讲解了本书的初稿。

我希望你们会喜爱这一中文译本，从而我能赢得更多的中国朋友和同事。

Tony Hoare

牛津，1988年6月

前　言

出自各种原因，知道本书正在著写的人都急切地等待着它；现在完全可以说，他们的耐心等待已经得到了酬谢。

原因之一，这是 Tony Hoare 的第一本书。很多人是在他在世界各地不知疲倦地作学术讲演时认识他的；更多的人知道他是一位行文清晰、细腻的论文作者。他的论文数量多、涉及面广，而且几乎在墨迹未干时，就成了经典著作。但是，书毕竟是一种不同的媒介：作者在书中表述自己的观点时，可不受通常十分严格的时空限制；书还给与作者一种机会，可以更深刻地说明自己的看法，并可谈及一些涉及面更广的论题。Tony Hoare 已经最充分地利用了书的这些优点。

更为实质的原因是来自本书的内容。25年前，计算界开始遇到并发现象，从此它就成为计算界的无穷无尽的困惑的发源地；这是由于并发现象可以出现在技术上是很不同的各类环境中，也由于并发现象导入了非确定性，非确定性的导入是计算界的一个历史性事件。为消除这一困惑，需要一个成熟的、具有献身精神的科学家的艰苦劳动；愿他幸运，能澄清这一局面。Tony Hoare已用他科学生涯中的主要精力对付这一挑战，我们应为此而感谢他。

最根本的原因是由读过本书早期初稿的人强烈意识到的。他的手稿以令人惊奇的透彻，表明了计算科学可以——或者甚至说，应该——成为怎样的一门科学。不要被自己造成的复杂情况弄糊涂了，这是计算科学家面临的主要挑战。感觉到或者声明这一挑战是一回事；而发现和说明如何依赖数学法则的准确无误和十分清晰的优点达到这一崇高的目标，就是完全不同的另一回事

了。具有和我同样鉴赏力的、充满感激之情的读者们，阅读本书时，可巨大得益于 Charles Antony Richard Hoare 的科学上的智慧，写法上的创新，以及处理上的巧妙。

Edsger W. Dijkstra

序　言

本书是献给有抱负的程序人员的，他们有志于对他们所从事的高智能职业有更深入的了解并掌握更高的技能。本书对一个熟知的论题采用新的探索途径，从而揭示了一些原来很难理解的事物的自然本质。这种途径是用一组例子来阐明的，这组例子选自各种应用领域，从自动售货机、童话、游戏，直至计算机操作系统。这些例子的处理是基于一种数学理论，这种理论是由一组系统化的数学法则所刻画的。本书的最根本的目的是让读者具有一种洞察力，能用新的眼光去考察当前和未来的问题，使这些问题能更有效、更切实地得到解决，甚至在某些情况下，可避开这些问题。

很显然，本书中的新观念可应用于描述、设计和实施那些不断动作，而且和环境不断交互作用的计算机系统。基本想法是，这类系统可以容易地分解成多个子系统，这些子系统并行地运行，它们之间以及它们与它们的共同环境之间交互作用。子系统的并行组合和传统程序设计语言中程序行或程序语句的顺序组合是同样简单的。

对系统的这种洞察可带来实际的好处。首先，可避免程序设计中并行性的很多传统问题，如干扰、互斥、中断、多流化、信号量等等。其次，近年来在程序设计语言和程序设计方法学研究中提出的很多高级结构化观念，也都是上述观念的特例，如管程、类程、模块、包、临界区、闭体、形、以及低级的子程序概念。最后，这种新观念还提供避免错误的可靠的数学基础，诸如发散、死锁和非终止等，也是使计算机系统的设计和实施的正确性严格可证的可靠的数学基础。

我已力求按逻辑的和心理的顺序逐步介绍这些观念，以简单

和基本的算子开始，逐渐展向它们的精巧的应用。好学的读者或许会逐页地阅读本书。但很多读者会从他们更有兴趣的题目开始；为此本书的每一章都刻意组织，以允许适宜的选读。

1. 每个新观念冠以非形式的说明，并用一些小例子解释，这样可能有助于所有读者。
2. 用来刻画各种运算的重要特性的代数法则，会引起喜爱数学的优美的读者的兴趣。如有读者愿用保持正确性的变形方法来优化他们的系统设计，那末这些法则也是有益于他们的。
3. 书中建议使用著名的程序设计语言 LISP 的一个非常简单的纯函数式的子集来实施各个算子，这种实施建议是不寻常的。对于曾用 LISP 实施来检验和论证系统设计的读者，对本书的这一建议也会很感兴趣的。
4. 系统分析员对迹的定义和描述应该感兴趣，因为他们在承诺实施前，需要描述顾客的需求。高级程序员对此也应该有兴趣，因为在设计一个系统时，他们需要将其划分为若干个子系统，并且清楚地描述它们间的接口。
5. 证明规则会对实施人员有益，因为他们负有责任，根据已有的描述、规定的日程、规定的价格，生产出可靠的程序。
6. 最后，书中的数学理论给出了进程概念的严格定义；并由算子所构造出的进程，严格给出了算子本身的定义。这些定义构成了代数法则、实施和证明规则的基础。

读者对上述任一论题缺乏兴趣，或者感到一时很难理解，可以随时略去或者暂缓阅读。

章间的承接也有便于读者粗读、选读或者重新安排的要求。第一章和第二章的前几节是导引，所有读者必读，但后面的各节可一掠而过，或者留待第二遍时再读。第三、四、五章是彼此独立的，读者可按兴趣和嗜好，以任意次序或组合进行阅读。我们建议，读者在遇到难懂的内容时，不必中断，可继续阅读下一

节，甚至下一章，因为略过的材料很可能不会立即又提到。在用到前面的内容时，书中常标有参照节码；这时，这些内容就可以读懂了，因为读者已充分了解了提出它们的背景。我希望读者最终能发现书中的每项内容都是有趣的，是值得一读的；但我并不期望每个读者都愿意按本书书写的次序阅读和掌握它。

选用来阐明本书中各观念的都是些小例子。这是有意这样安排的。解释一个观念的最初的几个例子本应十分简单；这样，才不会由于例子的复杂或者不熟悉而影响对观念的理解。阐明观念的后面的几个例子比较难解，所涉及的问题本身很易误解而且很易弄复杂；只是由于所用的概念有力，记号精巧，才得以简单地解答。

每个读者遇到过一些问题，这些问题比之教科书中的例子规模更大，远为复杂，远为重要，读者或许为它们付出过痛苦的代价。这类问题看上去是用任何数学理论都很难处理的。但切不可因此而泄气或者焦躁，不妨接受挑战，试着应用本书中的新方法去解决这些现实的问题。选择问题的某些局部，形成问题的一个大大简化了的模型，以此为出发点；在必要的时刻，再逐步复杂化。令人惊奇的是，最初的、简化的模型往往会增加你对问题的深刻了解，有助于你对整个问题的解决。另外，模型或许还可作为一种结构，然后，准确无误地往上添加复杂细节。最后，还可能惊奇地发现，某些添加的复杂细节根本是多余的。如能这样，掌握一种新方法所作的努力是完全值得的。

人们常常抱怨所用的记号。初学俄语的学生经常抱怨陌生的斯拉夫字母造成的障碍，特别是其中不少字母的发音奇特。尽管如此，这还是俄语学习中最容易的阶段。学会书写后，必须学习文法和词汇，然后掌握成语和文体，再改进用俄语表达自己思想的熟练程度。为掌握这一切，必须花时间去学，去练，不会一蹴而成。学习数学也是这样。开始时，符号也是一道难逾的路障；但是实质性的问题是理解符号的含义和特性，了解如何对它们进行操作，并且学会熟练地使用它们来表示新的问题、寻求新的解

答、给出新的证明。最终，你能培养自己具有对数学优美风格和文体的鉴赏力。达到这一境界时，符号也随之消失；由符号你可直接看到其含义。数学的优点在于，它的规则比之自然语言的规则简单得多，它的词汇比之自然语言的词汇也少得多。因此，当你遇到一些陌生的数学问题时，你仍有可能使用逻辑演绎和推测来解决这些问题，而不必请教书籍或专家。

这就是为什么数学会象程序设计那样有趣。可惜并不总是很容易的。数学家们在研究学科新分支时，也会遇到困难。通信进程的理论是数学的新分支；研究这一分支时，程序人员比之数学家，着手进行时并不处于更不利的地位；而在完成时却获得明显的优势，因为他们可实际应用他们的知识。

本书的内容已试用于非正式的工作会议和正式的课程教学。本书曾用于软件工程专业的硕士课程，讲授一学期，但大部分材料可作计算科学的学士课程，在最后一年或甚至第二年时讲授。要求的预备知识主要是高中代数、集合论概念、谓词演算符号。这些都列于序言后的符号汇编中。基于本书的内容，可为有经验的程序人员举办为时一周的集中讲授课程。在讲授时，教师应重点讲解例子和定义，而将数学味更重的材料留作课后自学。课时少于一周时，即使只能讲完第二章也是值得举办的；仔细选材的话，还可举办一小时的讨论班，一直讲到有教育意义的五个哲学家就餐的故事。

讲授通信顺序进程是十分有趣的工作，因为书中的例子提供讲演者发挥演剧才能的机会。一个例子相当一个剧本，演出时一般着重表演有关人物的感情。听众通常会从演出中感到死锁是特别滑稽的。但是听众要始终警惕这种人格化的危险性。讲演者藉助表演动机、爱憎、感情波动，“使枯燥、荒诞的故事听起来逼真”，而数学公式有意摆脱了这些人为因素。大家应该专心致志于理解数学公式的无人情味的、干巴巴的含义，并且学会欣赏数学抽象的优美。比如，某些递归定义的算法就具有J.S.Bach所创作的逃亡曲般的惊人的优美。

概　要

第一章介绍进程概念，进程是系统及其环境间交互作用的一种数学抽象。习用的递归技术可用来刻画经久不衰，或者永存的进程。该章中的概念先释以例子和图形；更完整的解释是由代数法则和用函数式程序设计语言的实施给出的。该章后半部分中说明，为什么可由进程所从事的一系列动作的迹来记载进程的行为。定义了许多迹的运算。在实施一个进程前，可用迹的特性描述它的行为。为实施进程，使其和描述的一致性是可以严格证明的，该章中给出了一些规则。

第二章说明如何将多个进程组装成系统，在系统中各部件交互作用，也和它们的外部环境彼此打交道。引入并发性本身并不产生任何非确定性。该章中的主要例子是处理五个就餐哲学家的传统故事。第二章后半章中说明，如何用改变进程事件的名字的方法，使进程很方便地具有新用途。这章以确定性进程的数学理论为结束，其中包括递归式的论域理论的一个简单介绍。

第三章对莫衷一是的非确定性问题给出一个最简单的现成的解。非确定性是实现抽象的重要技术，因为决定抹除系统行为中我们所不再感兴趣的方面时，就自然地会出现非确定性。非确定性保持某种对称性，这种对称性见于数学理论部分的算子的定义中。非确定进程的证明方法比之确定进程稍微麻烦些，因为必须证明每一种可能的非确定选择产生的行为，都符合进程的描述。幸运的是，我们可以找到迴避非确定性的技术，这些技术广泛地用于第四和第五章。从而，第三章的学习或精通可推迟到学习第六章前，第六章中就不能再避免引入非确定性了。

第三章的后面一些章节中，给出了非确定进程概念的一个完整的数学定义。纯粹数学家对这个定义是会有兴趣的，他们喜欢探究这门学科的基础，或者喜欢用证明的办法，检验代数法则和其它进程特性的有效性。而应用数学家（包括程序设计人员）可

能认为这些法则是不证自明的，或者喜欢用法则的效用来证明法则的有效"这样，他们可能就不去阅读这些更理论的章节。

直到第四章才介绍通信：通信是两个进程间相互作用的一种特例，其中的一个进程输出一个消息，与此同时，另一个进程输入这个消息。因此，通信是同步的；如果要在通道上使用消息缓冲，可在两个进程间插入一个缓存进程。设计并发系统的重要目的是在求解实际问题时获得高速计算。为说明这一目的，设计了一些计算数组的简单的脉动式（或迭代式）算法。导管就是一个简单例子，它由一串进程所定义，其中每一个进程仅从它的先行进程输入消息，仅向它的后继进程输出消息。导管在实施单向通信协议时很有用，这种协议具有层次结构。这章中用附庸进程模拟抽象数据类型这一重要概念，附庸进程的每个实例只和申明它的程序块通信。

第五章讲述如何在通信顺序进程的框架内集成顺序程序设计的惯用算子。有经验的程序人员可能会大吃一惊，想不到这些惯用的算子也和常用的数学理论中的算子一样，具有同一类的漂亮的代数性质；也想不到证明顺序程序满足它们的描述，和证明并发程序满足描述的方法非常相象。即使外部启动的中断也可用通信顺序进程定义，该章中并说明其用途，证明其遵从某些漂亮的法则。

第六章讨论怎样构造和实施这样一个系统，在这个系统中有很多的进程，它们共享有限的物理资源，诸如磁盘、行式打印机等，而且进程对资源的需求是随着时间而变化的。每种资源也由一个进程来表示。每当用户进程需要一种资源时，就建立一个新的虚拟资源。一个虚拟资源也是一个进程，它有点象用户进程的附庸进程；但也可和实际资源通信。这些通信是和其它同时活动着的虚拟进程的通信穿插进行的。故实际和虚拟进程和 PASCAL PLUS 中的管程和闭体扮演一样的角色。这章用模块式开发一系列完整但是非常简单的操作系统来解释各个概念，堪称本书中

规模最大的例子了。

第七章记叙了研究并发性和通信的其他途径，并且阐明了导致本书中的理论的技术的、历史的和作者个人的动因。在此我应感谢其他作者对我的影响，并推荐和介绍本领域中进一步阅读的材料。

致　谢

兹感谢Robin Milner的深刻的、创造性的工作。这些工作在他的有创见的著作——通信系统的演算(Calculus for Communicating Systems)中有详尽的说明。他的独有的洞察力，他的友谊和他专业上的执着，一直是本书竟成的各项工作的灵感和勇气的源泉。

在过去的20年中，我一直考虑着并行计算的程序设计问题，想设计一种程序设计语言以缓解这一问题。在这段时间中，我极大得益于与很多科学家的合作，包括Per Brinch Hansen, Stephen Brookes, Dave Bustard, Zhou Chao Chen, Ole-Johan Dahl, Edsger W. Dijkstra, John Elder, Jeremy Jacob, Ian Hayes, Jim Kaubisch, John Kennaway, T. Y. Kong, Peter Lauer, Mike Mckeag, Carroll Morgan, Ernst-Rudiger Olderog, Rudi Reinecke, Bill Roscoe, Alex Teruel, Alastair Tocher和Jim Welsh。

最后，特别感谢O.-J. Dahl, E. W. Dijkstra, Leslie M. Goldschlager, Jeff Sanders等人，他们仔细阅读了本书的初稿，并且指出了原稿中错误和费解处；也要感谢一九八三年一月，Wollongong暑期学校计算机程序设计科学讲座的参加者，一九八三年四月中国科学院研究生院讲习班的听众们，以及一九七九年至一九八四年牛津大学计算专业的硕士生们，他们听取了本书各章的有关内容。

符 号 表

逻辑符号

记号	含义	举例
$=$	相等	$x=x$
$\neq$	不等	$x\neq x+1$
□	例子或证明的结束符	
$P\wedge Q$	P 和 Q（两者为真）	$x\leqslant x+1\wedge x\neq x+1$
$P\vee Q$	P 或 Q（两者或其一为真）	$x\leqslant y\vee y\leqslant x$
$\neg P$	非P（P 不真）	$\neg 3>5$
$P\Rightarrow Q$	若 P 则 Q	$x<y\Rightarrow x\leqslant y$
$P\equiv Q$	P 当且仅当 Q	$x<y\equiv y>x$
$\exists x.P$	存在 x 使 P 真	$\exists x.x>y$
$\forall x.P$	对一切 x，P 真	$\forall x.x<x+1$
$\exists x:A.P$	存在集合 A 中元素 x，使P真	
$\forall x:A.P$	对集合 A 中一切元素 x，P 真	

集合符号

记号	含义	举例
$\in$	属于	拿破仑$\in$人类
$\notin$	不属于	拿破仑$\notin$俄罗斯人
$\{\ \}$	空集（无元素集）	$\neg$（拿破仑$\in\{\ \}$）
$\{a\}$	a组成的单元素集； a是其仅有的元素	$x\in\{a\}\equiv x=a$
$\{a,b,c\}$	a,b,c组成的集	$c\in\{a,b,c\}$
$\{x\mid P(x)\}$	使$P(x)$为真的全体x的集合	$\{a\}=\{x\mid x=a\}$
$A\cup B$	A并以B	$A\cup B=\{x\mid x\in A\vee x\in B\}$
$A\cap B$	A交以B	$A\cap B=\{x\mid x\in A\wedge x\in B\}$
$A-B$	A减去B	$A-B=\{x\mid x\in A\wedge\neg x\in B\}$
$A\subseteq B$	A包含于B	$A\subseteq B\equiv\forall x:A.x\in B$
$A\supseteq B$	A包含B	$A\supseteq B\equiv B\subseteq A$

$\{x:A\mid P(x)\}$	使$P(x)$真的A中全体x	
$\mathbf{N}$	自然数集	$\{0,1,2,\cdots\}$
$\mathbf{P}A$	A的幂集	$\mathbf{P}A=\{X\mid X\subseteq A\}$
$\bigcup_{n>0} A_n$	集合族的并集	$\bigcup_{n>0} A_n=\{x\mid \exists n\geqslant 0.x\in A_n\}$
$\bigcap_{n>0} A_n$	集合族的交集	$\bigcap_{n>0} A_n=\{x\mid \forall n\geqslant 0.x\in A_n\}$

函　数

记号	含义	举例
$f:A\to B$	f是将A中每个元素映射到B中元素的一个函数	square: $\mathbf{N}\to\mathbf{N}$
$f(x)$	（A中）x经f得到的B中的映象	
单射	将A中元素映射到B中不同元素的函数	$x\neq y\Rightarrow f(x)\neq f(y)$
f^{-1}	单射f的逆	$x=f(y)\equiv y=f^{-1}(x)$
$\{f(x)\mid P(x)\}$	将f作用于使P为真的全体x所得到的集合	$\{y\mid \exists x.y=f(x)\wedge P(x)\}$
$f(C)$	由f形成的C的映象集	square$(\{3,5\})=\{9,25\}$
$f\circ g$	f复合以g	$f\circ g(x)=f(g(x))$
$\lambda x.f(x)$	将x的每个值映射至$f(x)$的函数	$(\lambda x.f(x))(3)=f(3)$

迹

节号	记号	含义	举例
1.5	$\langle\ \rangle$	空迹	
1.5	$\langle a\rangle$	仅含a的迹	（单元序列）
1.5	$\langle a,b,c\rangle$	由a然后b然后c构成的迹	
1.6.1	$\frown$	（迹间的）相接	$\langle a,b,c\rangle=\langle a,b\rangle\frown\langle\ \rangle\frown\langle c\rangle$
1.6.1	s^n	重接s n次	$\langle a,b\rangle^2=\langle a,b,a,b\rangle$
1.6.2	$s\restriction A$	s受限于A	$\langle b,c,d,a\rangle\restriction\{a,c\}=\langle c,a\rangle$
1.6.5	$s\leqslant t$	s是t的前缀	$\langle a,b\rangle\leqslant\langle a,b,c\rangle$
4.2.2	$s\overset{n}{\leqslant}t$	从t的尾部至多移走n个符号后可得到s	$\langle a,b\rangle\overset{2}{\leqslant}\langle a,b,d,c\rangle$
1.6.5	$s\ \mathrm{in}\ t$	s在t中	$\langle c,d\rangle\ \mathrm{in}\ \langle b,c,d,a,b\rangle$

1.6.6	$\#s$	s的长度	$\#\langle b,c,b,a\rangle = 4$
1.6.6	$s \upharpoonright b$	s中b的个数	$\langle b,c,b,a\rangle \upharpoonright b = 2$
1.9.6	$s \upharpoonright c$	s中记载的通道c上的通信	$\langle c.1,a.4,c.3,d.1\rangle \upharpoonright c = \langle 1,3\rangle$
1.9.2	$\frown/s$	压扁后的s	$\frown/\langle\langle a,b\rangle,\langle\ \rangle,\langle c\rangle\rangle = \langle a,b,c\rangle$
1.9.7	$s;t$	s成功地接以t	$(s\frown\langle\surd\rangle);t = s\frown t$
1.6.4	A^*	A中元素的序列集	$A^* = \{\, s \mid s \upharpoonright A = s \,\}$
1.6.3	s_0	s的首元素	$\langle a,b,c\rangle_0 = a$
1.6.3	s'	s的尾部	$\langle a,b,c\rangle' = \langle b,c\rangle$
1.9.4	$s[i]$	s的第i个元素	$\langle a,b,c\rangle[1] = b$
1.9.1	$f^*(s)$	s的f星函数	$\mathrm{square}^*(\langle 1,5,3\rangle) = \langle 1,25,9\rangle$
1.9.4	$\bar{s}$	s的逆置	$\overline{\langle a,b,c\rangle} = \langle c,b,a\rangle$

特殊事件

节号	记号	含义
1.9.7	$\surd$	成功（成功终止）
2.6.2	$l.a$	名为l的进程参予事件a
4.1	$c.v$	在通道c上传递值v
4.5	$l.c$	名为l的进程的通道c
4.5	$l.c.v$	在通道$l.c$上传递消息v
5.4.1	$\lightning$	灾难（闪电）
5.4.3	⊛	交换
5.4.4	©	为恢复所设的备查点
6.2	*acquire*	获取
6.2	*release*	释放

进　程

节号	记号	含义
1.1	αP	进程P的字母表
4.1	αc	在通道c上可传递的消息的集合
1.1.1	$a \to P$	a然后P
1.1.3	$(a \to P \mid b \to Q)$	在a然后P与b然后Q间的选择（要求$a \neq b$）
1.1.3	$(x:A \to P(x))$	在A中选取x然后$P(x)$
1.1.2	$\mu X:A.F(X)$	满足$X = F(X)$的字母表为A的进程X