

交互计算模型概述*

付岩 黄铁军 余珂 李涛 张浩
(中国科学院计算技术研究所 北京 100080)
(中国科学院研究生院 北京 100039)
(中国科学院软件研究所 北京 100080)
(yfu@ict.ac.cn)

摘要 随着计算技术日新月异的发展,以算法为核心、以图灵机和 Church 论题等为理论依据的计算科学理论范式正在面临越来越大的挑战。介绍了一个新的计算模型—交互计算模型的基本思想,它是对算法的扩展,并比算法具有更强的描述能力。一系列基本概念被扩展到交互。

关键字 交互,算法,图灵机,计算模型,逻辑

中图法分类号 TP301

Overview of Interactive Model of Computing

FU Yan , HUANG Tie-Jun , YU Ke , LI Tao , ZHANG Hao

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

(Graduated School, Chinese Academy of Sciences, Beijing 100039)

(Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract The fundamental idea of a new model of computing, interactive model of computing, is introduced. Algorithm, Turing Machine and Church's thesis have been regarded as the theoretical foundation of computing ever since. But the computer technology has changed so rapidly that the algorithm and Turing machine can't play a role of theoretical paradigm of computing science any longer. This fact leads to the new interactive model of computing, which is an extension to algorithm and more expressive than algorithm.

Keywords: interaction , algorithm , Turing Machine , model of computing , logic

长期以来,人们一直将算法和图灵机理论作为计算机科学的理论基础。然而,70年代以来,计算机技术经历了从大型机到工作站和网络、从吞噬数字的机器到嵌入式系统和图形用户界面以及从面向过程到基于对象和分布式编程的巨大转变。人们已经开始觉察到,如今的计算机所能完成的任务以及我们所努力的目标已与图灵机和算法的行为相差甚远。

在中国计算机学会第七届理事会上,中国工程院院士汪成为在他的报告中提出了关于我国计算机发展的12个问题^[1]。第一个问题就是“有人认为:从图灵机的可计算理论可以得出‘可计算’的三个前提:必须把问题形式化;必须有一个算法;必须有合理的复杂度。请问:我们是否应该尝试突破、并有可能(或部分地)突破这些前提?还是只能在这个前提下进

*本文得到中国科学院知识创新工程方向性项目(编号:KGCX2-103)支持

行创新？’我们认为汪成为院士所提出的这一问题对计算机科学的发展具有重大而深远的意义，本文正是以探讨为目的主要介绍了美国计算机科学家 Peter Wegner 及其合作者对算法和图灵机理论所做出的突破—交互计算模型^[2, 3, 4]，希望对创新工作有所启示。

1 从面向对象看算法的局限性

目前，人们已经得到这样一个共识，“软件危机”是由于图灵机或冯·诺伊曼计算机体系结构并非是表示和处理人类思维的合理模型所引起的^[5]。解决的方法就是应使描述问题的问题空间和解决问题的方法空间在结构上尽可能的一致，这就是面向对象技术的方法学所追求的基本原则。但是，面向对象技术与以前计算机科学中的方法是如此不同，以至人们无法在理论和数学上说明面向对象技术到底是什么^[6]。

面向对象程序设计将交互引入其中，就持久对象这方面来说是鼓励一种自顶向下的看待世界的方法，而算法通过独立于时间的输入输出转换来自底向上的模拟世界^[6]。Peter Wegner 就是在试图联结自底向上和自顶向下的计算机科学之间的间隔时认识到算法在本质上是无力描述面向对象和分布式系统的行为的，也就是图灵机(TMs)无法描述交互式计算。Wegner 在 1992 年日本第五代计算机计划的闭幕会议上首次提出 TMs 不能模拟交互这一思想——计划在把计算归约到逻辑上的失败不是由于逻辑编程人员的智慧不够，而是由于在理论上这一归约是不可能的，一阶逻辑无力模拟对象行为^[6]。

关于算法的局限性有很多证据，Milner 早在一九七五年就注意到并发过程是不能用顺序算法来描述的^[7]。再如，Manna 和 Pnueli 已经证明，永远运行的系统如操作系统也不能由算法模拟^[8]。

2 扩展算法到交互

这就导致了一个彻底的理论框架的转变，需要我们重新思考有关计算机科学本质的基本假设，如 Church 论题将计算的直观概念等同于 TMs 计算。这导致了新的计算模型——交互计算的发展，TMs 被扩展到交互机(IMs)，引进了建立在观察能力而不是转换能力上的对描述能力的测量标准，根据这一标准 IMs 比 TMs 更具描述能力。

定义 1 (交互性质^[2]): 如果一个计算主体(agent)具有与不受它控制的外部环境交互的输入和输出动作，则它具有交互性质。

定义 2 (交互机^[2], Interaction Machine): 一个交互机(IM)使用具有交互性质的输入和输出动作扩展了的 TM。TM 有确定的输入带，而 IMs 的输入流的元素(element)是由不在其控制之下的一个外部机制动态提供的。

定义 3 (观察等价性^[2], observation equivalence): 给定一类可观察系统 C，观察集合 O 决定了一个 C 上的观察等价关系使得系统 S, S' 是等价的，当且仅当它们对 O 中所有的观察是不可区分的。

定义 4 (观察者的描述能力^[2], expressiveness of observers): 一个观察者 O1 对一个系统类与 O2 具有同样的描述能力，如果对于 O1 的观察等价意味着对于 O2 的观察等价。

有充足的证据表明交互不能由算法描述，但在理论上证明这一点是比较困难的。为此 Peter Wegner 和他的合作者 Dina Goldin 提出了机器模型和数学模型来达到这一目的^[6]。

3 交互的机器模型

下面首先简要回顾了图灵机，然后分别介绍了持久图灵机、时序交互机和多带交互机，最后比较了它们的表达能力。

3.1 图灵机(TM)

图灵机定义为一个五元组： $M = (S, \Sigma, \Delta, S_0, S_F)$ ，其中 S 是一个状态集， Σ 是一个输入字符集， Δ 是个状态转移关系， S_0 是初始状态，而 S_F 就是最终状态。一个图灵机有一个无限的输入输出带。图灵机通过输入一串有限的输入字符来进行计算，开始时读写头位于带子的头部，然后通过一系列的状态转换直到最终状态。最后任何在工作带上的内容就是输出。图灵机的行为可以通过一个部分递归函数 f_M 来表示 $f_M(\text{input}) = \text{output}$ 。图灵机的行为既不是历史依赖的也不是永久的，其计算并不依赖于以前的计算。

3.2 持久图灵机(PTM)

定义 5 (持久图灵机^[9], PTM): 持久图灵机是对图灵机具有最小扩展的多带图灵机，它具有一个持久的工作带。PTM 的每一步计算(状态转换)是一个图灵计算，工作带记录了 PTM 计算的历史。在每一步计算结束后，工作带上的内容被保存下来，为下一次计算做准备。

PTM 工作带上的内容 w 与每一次的输入共同影响下一次的输出。它的工作行为可以由下面的递归函数 f_M 表示： $f_M(\text{input}, w) = (\text{output}, w')$ 。此即，对于相同的输入 PTM 的输出不一定相同(与 TM 相反)，还要依赖于工作带上的内容。持久图灵机(PTM)在表达能力上比图灵机要强^[10]。

3.3 时序交互机(SIM)

在典型的图灵机模型中，当输入被确定以后，所有的计算过程都是确定的，并且相应的计算结果也是确定的，而这恰与交互机相反。在交互机的计算过程中，包含具有不确定性的 I/O 行为。

定义 6 (时序交互机^[11], SIM)：时序交互机是这样的一种交互机 $M : M = (S, I, m)$ ； S 是一个有限的状态集合； I 是一个有限的动态流输入集， m 是一个状态转换映射： $S \times I \rightarrow S \times O$ 。

SIM 的每一步计算 $(s, i) \rightarrow (s', o)$ 可以看成是一个完全的图灵计算，其中 i 是动态输入字符，输出 o 对下一个输入产生影响， s' 是 SIM 的下一个状态。SIM 的行为能通过 I/O 流来描述。I/O 流具有这种形式 $(i_1, o_1), (i_2, o_2), \dots$ ，每一个 o_k 都是由 i_k 计算得来，但它同时对下一个输入即 i_{k+1} 产生影响，如图 1 所示。

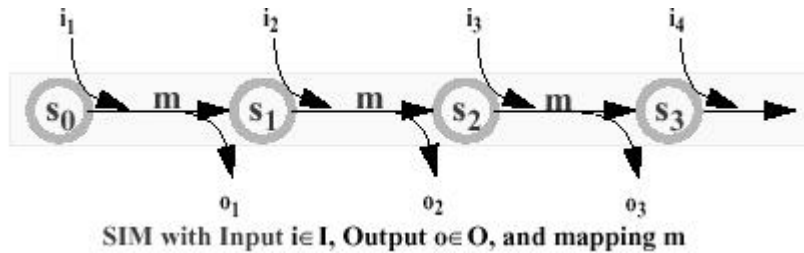


图 1 SIM^[11]

SIM 的表达能力同样是要比 TM 强，而与 PTM 的表达能力是等价的。

3.4 多带交互机(MIM)

多带交互机是与多个独立的流交互的有限机器模型。例如，为多个独立的用户提供服务的分布式数据库或 ATM 系统。MIM 能够比 SIM 执行更大范围的任务，从这个意义上说，MIM 的行为是不能被 SIM 来表达的^[11]，这与多带图灵机与单带图灵机具有相同的表达能力形成了对比。

MIMs 比 SIMs 有更强的表达能力的两个最重要的理由是^[11]：

1. MIMs 可以表达非确定性的行为，而这不能为 SIMs 表达。

MIMs 模拟不能被任何连续的观察者完全观察到的分布式系统。由 SIM 和一个观察者组成的系统是闭合的，而带有一个主观察者的 MIM 是开放的系统，因为它可以被第二位的观察者们不可预测地更改，这种更改对于主观察者来说就看作是一种不确定性的行为。

2. MIMs 可以表达非序列化的行为而 SIMs 不能。

简单地，如此表述：有 I/O 流 1 : i_1 , I/O 流 2 : i_2 , I/O 流 1 : o_1 , I/O 流 2 : o_2 ，但 o_1 受 i_2 影响，因此不能序列化为 $(i_1, o_1), (i_2, o_2)$ 。协作(collaboration), 协调(coordination), 管理(management)是能被 MIMs 模拟的，而不能由 SIMs 或 PTMs 实现。

总之，这几种机器模型的表情能力可简单描述为：MIM>SIM=PTM>TM。

4 交互的数学模型

从算法到交互的转变在数学上表现为一系列的扩展：从归纳 (Inductive) 到余归纳 (Coinductive) 的扩展、从良基集 (Well-Founded Set) 到非良基集 (Non-Well-Founded Set) 从代数 (algebras) 到余代数 (Coalgebras) 的扩展^[12]。从归纳到余归纳的扩展表达了从字符串到流的转变，这是从算法到交互的转变的基础；非良基集作为流的顺序行为的形式化模型被引进；余代数则为流的演算提供了工具。

4.1 归纳方法的扩展

归纳法是有限计算个体的定义方法和推理规则，存在固有的不足。图灵机和一阶逻辑，作为实际计算机的数学模型，其局限性从根本上说，是源于归纳法^[12]的。归纳法是构造主义的方法，可以将归纳法分解成若干基本步骤。归纳法从本质上来讲是封闭的。

定义 7 (归纳法): (1) 初始条件 ; (2) 迭代规则 ; (3) 最小化条件。

归纳法是定义字符串集合、语言、形式系统和可计算函数的基础。

字母表 A 上的字符串集合 A^* 可归纳地定义如下：

(1) 空串 A^* ; (2) 如果 $x \in A^*$, 则 $ax \in A^*$; (3) 除此以外, A^* 不含其它元素。

可计算函数 $f: X \rightarrow Y$ 在两个层次上是归纳的。f 将定义域 X 映射到值域 Y, 其中

(1) 归纳定义的值域 (静态): 定义域 X 是归纳定义的。

(2) 归纳定义的计算 (动态): 计算过程是归纳定义的。

图灵机类似的也通过归纳定义的状态转换步骤对归纳定义的串进行变换。

算法、可计算函数和形式系统在表达能力上之所以与图灵机等价, 而不能超越, 原因就在于它们都使用归纳定义的静态域和归纳定义动态过程。

定义 8 (余归纳法^[12, 13], coinductive): (1) 迭代规则 (循环条件); (2) 最大化条件。

与归纳法相比, 余归纳法去掉了初始条件, 修改了迭代条件使之成为循环条件, 用最大化条件替换了最小化条件。静态的初始化被去除, 取而代之的是交互信息的后期动态绑定, 体现了从封闭系统到开放系统的扩展; 最大化条件影响了对迭代规则的解释, 所有规则没有排除的元素都可以被包含 (最小化规定只有规则指定的元素可以被包含)。

例: 流可以通过余归纳法来定义。

(1) 迭代规则: 如果 t 是字母表 A 上的流, a 是 A 中的元素, 则 (a,t) 也是字母表 A 上的流。

(2) 最大化条件: 字母表 A 上的流的集合是满足迭代规则的最大的集合。

交互计算在两个层次上是余归纳的: (1) 余归纳定义的静态值域; (2) 余归纳定义的动态的下步动作。

4.2 集合论的扩展

非良基集理论为流的顺序交互行为提供了一个一致的形式化模型。它在交互模型中的地位类似于可数集在 Church-Turing 计算模型中的地位^[2]。良基集和非良基集通过良基关系和非良基关系来定义。

定义 9 (良基关系、非良基关系): 一个在集合上的二元关系 R 是非良基关系, 当且仅当存在一个无穷序列 $b_0, b_1, \dots, b_i, \dots, b_i \in S, i=0, 1, \dots$ 使得对于所有 i 都有 $b_{i+1} R b_i$, 否则称 R 是良基关系。

集合 S 的良基性取决于该集合的成员关系。如果存在无穷序列 S_0, S_1, \dots 使得 $S_{i+1} \in S_i, i=0, 1, \dots$ 其中 $S_0=S$, 则该集合的成员关系是非良基关系, 否则是良基关系。

定义 10 (良基集和非良基集): 一个集合是良基集, 如果该集合的集合成员关系是良基关系, 否则就是非良基集。

基础公理将集合限制在良基集的范围内, 将基础公理替换为反基础公理可以实现集合从良基集向非良基集的扩展^[14]。

定义 11 (基础公理, FA): 所有集合都是良基集。

定义 12 (反基础公理, AFA): 每个集合方程组都有唯一的解。

ZFC 公理集合论中包含了基础公理 FA ($ZFC = ZFC_+ + FA$)。非良基集合理论 (ZFA) 用反基础公理 (AFA) 代替基础公理 (FA), 即 $ZFA = ZFC_+ + AFA$ 。

FA 严格描述了可归纳定义的集合。AFA 严格描述了可余归纳定义的集合。集合方程组 $X = \{a, Y\}, Y = \{a, X\}$ 在 ZFC 中是没有解的, 而在 ZFA 中有一个非良基集的解 $X \rightarrow \{a, Y\} \rightarrow \{a, \{a, X\}\} \rightarrow \{a, \{a, \{a, \dots\}\}$ 。FA 对于集合的限制显得不必要的严格, AFA 放宽了这种限制。和 ZFC 一样, ZFA 是一致的^[15]。

4.3 代数的扩展

余代数是进行流的演算的工具。它在交互计算模型中的地位相当于演算在 Church-Turing 模型中的地位^[16]。

定义 13 (代数): 代数是一个结构 $A=(S, m: F(S) \rightarrow S)$, 其中 S 是值集, $m: F(S) \rightarrow S$ 是一个保值同态映射, 将按语法规则构成的表达式集合映射到值的集合上。

代数结构中的表达式是归纳定义的, 演算过程也是归纳的。从代数到余代数的扩展就类似于从归纳到余归纳、从良基集到非良基集的扩展。

定义 14 (余代数, coalgebra): 余代数是一个结构 $CA=(S, m: S \rightarrow G(S))$, 其中 S 是一个状态集, $m: S \rightarrow G(S)$ 是一个保持动作的同态映射, 将状态未知的被观测系统映射到一组动作 (观察)。

这里通过对归纳、良基集和代数三方面的扩展为交互计算模型初步建立了数学模型, 为时序交互机 SIM 奠定了数学基础。对于多带交互机 MIM, 仍需要做进一步的工作。

5 扩展图灵测试到交互

图灵测试测试了非交互提问者区分图灵机和人类行为的能力。图灵认为思维能被行为所模拟是对的, 但他提出的思维的行为模型——TMs 太弱了。通过将询问的机器从图灵机扩展到交互机而扩展了图灵测试。由交互图灵测试定义的交互思维比最初的图灵测试所能定义的思维有更强的形式。交互图灵测试允许被询问的机器能够适应和学习, 能根据提问者的需要调整自己的答案, 就像一个好的演讲者根据观众的需要调整演讲的风格。扩展图灵测试到交互不仅帮助我们理解了思维的交互的本质, 也使我们能够去探索 TMs 和图灵测试 (图灵的两大贡献) 两者间的关系, 这两者通常被认为是没什么关系的, 而实际上是有关的。图灵对计算机模型的定义是他根据计算的精确概念来定义思维的起点。扩展的机器模型对计算概念的扩展使我们能够通过扩展图灵机来定义更强的思维形式^[6]。

6 扩展逻辑到交互

6.1 不完备性

可靠性和完备性将证明的语法过程同公式在其中得以解释的被模拟的论域的语义性质相联系。在推理和模拟间存在着折扣: 纯粹基于规则的语义推理无力涵盖数学、计算或物理系统的语义性质。模型建造者必须在完备可形式化和模拟能力二者间选择其一。TMs 以模拟能力为代价而将注意力集中在可形式化上, IMs 却在可形式化之上选择了模拟能力^[2]。

歌德尔证明, 整数上的算术不能由逻辑完备的形式化, 因为不是其中所有的性质都能被语法描述为定理。扩展歌德尔定理来证明, 交互和经验系统同样是不完备的, 因为它们具有的性质多得无法描述为可靠完备逻辑的定理^[2]。

命题 1 (不完备性的必要条件): 一个可靠且完备的一阶逻辑 (SCL) 只能模拟具有递归可枚举 (RE) 数量的性质的论域。

推论 1.1 (不完备性的充分条件): 任何具有非 RE 数量的性质的系统不能由 SCL 描述。

推论 1.2 (交互) IMs 的行为集不能由 SCL 形式化。

命题 2 (不可归约性) IMs 不能由 TMs 描述或归约到 TMs。

算法被小心的定义, 以使其计算数量是 RE 的^[2]:

命题 3 (算法): 算法是具有 RE 数量的计算的抽象概念。

6.2 逻辑的扩展

一阶逻辑是封闭的、非交互的。它通过较高的抽象、较弱的描述能力而获得了完备性^[2]。交互模型用较弱的抽象性取代了一阶逻辑的完备语义, 牺牲了完备性而获得了描述能力。

Peter Wegner 提出用多语用接口^[2]为交互模型发展一个逻辑框架。将逻辑模型 $L=(R,W)$ 用语法表示 R 来描述模拟世界 W 的语义, 是一个闭系统。通过把模型扩展到形式 $M=(R,W,P)$ 其中 P 是和能与 M 交互并修改 M 的客户端和观察者相联系的语用构件的集合 (P_1,P_1,\dots) , 使逻辑在原则上能够被扩展到可被多个观察者访问的演变的模拟世界。语用构件与接口相联系。从一个主体的角度来看, 语用可以形式地模拟为世界 W 在输入传感器上的投影。

7 对 Chomsky 体系的扩展^[3]

本质上, Chomsky 自动机体系的建立是通过逐步放松对访问带的限制来达到的。有穷自动机要求带为只读, 下推自动机允许一条辅助性的无界的下推型的带, 线形界限自动机允许写带但限界, 图灵机允许无限制但非交互式访问带 (带是被动的)。进一步的扩展自然就是允许带与机器交互, 这种机器即 IM, 主动性的带称为流, 多流的交互机表达能力更强者于单流交互机。

相应文法也有扩展, 即交互式文法 IG ^[3]。传统文法生成的字符串集合只有结构特性, 交互式文法生成的是字符流集合, 字符流体现了时间的无限与不可逆转性。

8 从理性主义到经验主义

柏拉图认为我们所感知到的世界只是理性世界 (真实世界) 的影子, 是不完美的, 因而没有价值的。笛卡尔 “我思故我在” 的论断认为存在源于思维。理性主义者蔑视经验事实, 极端的认为一切知识可以通过非交互的推理规则从必然正确的公理中获得, 这种思想统治了人们两千年, 直到 17、18 世纪遭到了经验主义者的批判, 并在物理学中首先并攻破。但是理性主义仍然吸引着很多数学家和逻辑学家, 如罗素和希尔伯特试图将数学归约到逻辑。

哥德尔的不完备性定理显示出了理性主义数学的局限性, 但其对计算机科学的启示仍然没有被充分把握。很多计算机科学家相信计算的直观概念可归约于图灵可计算这个形式化的概念。图灵机和算法追求完备性, 试图用与外界隔绝的精确的推理计算解决一切问题, 在把解决问题的直观概念限制到内在智力的过程中成为理性化的抽象概念。经验主义接受经验的不完备性, 但认为从不完备的经验中完全能够获得足够的信息, 达到预测和控制的目的。交互计算模型放弃了完备性, 动态的与外界交互, 在计算推理中吸取经验信息, 从而是经验主义的。所以说, 从算法到交互的转变对应着从理性主义到经验主义的转变^[2]。

参考文献

- 1 汪成为. 请教关于我国计算机发展的12个问题. 中国计算机学会第七届理事会, 2000
(Wang Chengwei. Consultation of 12 problems about the development of computer in our country. The 7th council of China Computer Federation, 2000)
- 2 Wegner P. Towards empirical computer science, available at www.cs.brown.edu/people/pw.
- 3 Wegner P. Interactive foundations of computing, Theoretical computer science, 1998, 192(2): 315-351.
- 4 Wegner P. A research agenda for interactive computing, 1998, available at www.cs.brown.edu/people/pw.
- 5 汪成为, 高文, 王行仁. 灵境(虚拟现实)技术的理论、实现及应用. 清华大学出版社, 1996
(Wang Chengwei, Gao Wen, Wang Xing Ren. Theory, implementation and application of virtual reality technology. Beijing: Tsinghua University Press)
- 6 Wegner P. Banquet speech, on the 13th European conference on object-oriented programming, Lisbon, Portugal, 14-18 June 1999.
- 7 Milner R. Elements of interaction. Communications of the ACM, 1993,36 (1):78-89.
- 8 Wegner P. Why interaction is more powerful than algorithms. Communications of the ACM, 1997, 40(5): 81-91.
- 9 Goldin D, Wegner P. Persistence as a form of interaction. Brown University, Tech Rep: CS-98-07,1998.
- 10 Goldin D, Wegner P. Behavior and expressiveness of persistent Turing Machines. Brown University, Tech Rep: CS-99-14,1999
- 11 Wegner P, Goldin D. Models of Interaction. ECOOP'99 Tutorial, the 13th European conference on object-oriented programming, Lisbon, Portugal, 14-18 June 1999.
- 12 Wegner P, Goldin D. Mathematical models of interactive computing. Brown University, Tech Rep: CS 99-13,1999.
- 13 Wegner P, Goldin D. Coinductive models of finite computing agents. Proc. Second International Workshop on Coalgebraic Methods in Computer Science, Amsterdam, March 1999.
- 14 Barwise J, Moss L. Vicious Circles. CSLI Lecture Notes #60, Cambridge University Press, 1996.
- 15 Finsler, p. Finsler set theory: Platonism and circularity. Boston: Birkhauser Verlag, 1996.
- 16 Rutten J. A tutorial on coalgebras and coinduction. The Hyper Bulletin of the European Association for Theoretical Computer Science, 1997, 62: 222-259