思维的计算特征与计算的思维属性

史文崇
(河北科技师范学院数学与信息技术学院 秦皇岛 066004)

摘 要 研究了计算思维的本质和培养计算思维应注意的问题。从分析国际学术界的一些观点出发,认为计算思维是一种学科思维,也是一种计算哲学。指出理解计算思维的关键是把握思维的计算学科特征和计算的思维属性。提出了计算思维的标志性概念的筛选原则,并由此给出了一些标志性概念,认为计算思维就蕴含在这些概念的共性之中,其典型思想有结构化思想、形式化思想、联动化思想、最优化思想。培养计算思维必须以此为基础。

关键词 计算思维,计算学科特征,思维属性,标志性概念,典型思想
中图法分类号 TP301 **文献标识码** A

My Views of Computational Thinking

SHI Wen-chong
(School of Mathematics & Information Technology, Hebei Normal University of Science & Technology, Qinhuangdao 066004, China)

Abstract The author researched the essence of computational thinking and the problems to be paied attention in cultivating computational thinking. After analysing the views of international academia, the author thought computational thinking is a kind of academic thinking and a kind of computing philosophy. He pointed out that grasping the thinking's computing disciplines' characteristics and the computation's thinking properties is critical to understand computational thinking. He presented the selecting principles of computational thinking' s banner concepts, gave some concepts based on the principles and thought the computational thinking is contained in the commonalities of those concepts. The typical thoughts of computational thinking are structuralization, formality, co-movement, optimization. Training computational thinking must be based on the banner concepts and the typical thoughts.

Keywords Computational thinking, Computing discipline's characteristics, Thinking properties, Banner concepts, Typical thoughts

2006年,周以真(Jeannette M. Wing)提出"计算思维",其引起国际社会的共鸣,不仅催生了美国CPATH计划(2007年)和CDI计划,也引起国内学者的热捧,使得高等教育界"九校联盟(C9)"倡议在高校计算机基础教学中培养计算思维。把握计算思维的本质,充实计算思维理论体系,对于培养和运用计算思维大有补益。

1 计算思维的本质内涵

人类数百年前就关注了思维和计算的关系。18世纪,乘法机的发明者、二进制的奠基人莱布尼茨提出的符号思维和演算推论器是"思维的本质是计算"的最早说明[1];1950年图灵在论文《计算机器与智能》中详细论证了心灵的计算本质。此后,"心灵的本质是计算"成为人工智能理论的基本假说,也是"思维的本质是计算"的另一种表述。

学术界也注意了学科对思维、乃至对世界观的影响。上世纪末,美国发展心理学家霍华德·加德纳(Howard Gardner)指出,"每一门学科都以自己独特的方式思考和认识着世界";"用一门学科的符号体系和风格进行交流,就是在发展学科思维";在教学中要注重传授"学科思维"[2],让学生"用学科专家们那样独特的思维方式来理解世界。"[3]同时,美国哲学会计算与哲学分会就"计算机如何在改变哲学"问题作了全国调查,结果认为"涌现了一个新的哲学范式"[4],国内学者也积极地附和了这一观点,认为"计算已经成为一种新的世界观"[5],甚至倡导"计算主义"[6]。2011年,美国图灵奖获得者理查德·卡普(Richard Karp)提出"计算透镜"理论,倡导将计算作为一种通用的思维方式,解决各学科的问题[7],实际上也是倡导计算哲学。

1992年,国内学者黄崇福从人工智能视角最早定义了计算思维的概念[8];2002年,董荣胜等提出计算机科学与技术方法论[9]。这些虽然尚未触及对于非计算学科的普遍性指导意义,还不能成为计算哲学,也没有提及计算思维的概念,但已经揭示出计算的思维属性及其规律。

2006年3月,周以真教授在CACM上发表《计算思维》,认为:计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解的思维活动[10];2010年,周以真在与他人合著的文章中给出了计算思维的另一种解释:计算思维是与形式化问题及其解决方案相关的一个思维过程,其解决问题的表示形式应该能有效地被信息处理代理执行[11]。

到稿日期:2013-04-25 返修日期:2013-09-02
史文崇(1965—),博士,副教授,主要研究方向为社会计算、社会多媒体计算,E-mail:mr_shi_pb@126.com。

这与她先前的表述完全不同。

或许周以真的计算思维“不便操作”，2011 年，国际教育技术协会（ISTE）和计算机科学教师协会（CSTA）给出了计算思维的“操作性定义”，将其阐释为一个 6 步骤的问题解决过程[12]。

综上所述，计算思维是一种学科思维，一种计算哲学。国内外学者关于计算思维有多种表述，而周的《计算思维》作为一篇“科技散文”[13]，更生动、更通俗、更具启迪性。爱因斯坦说，一个概念愈是普遍，愈是频繁地进入人的视野，我们要想理解它的意义也愈困难[14]。理解“什么是计算”、“什么是思维”尚且如此，理解计算思维更是如此。笔者认为，计算思维既然是从计算学科抽象而来，必然具有明显的计算学科特征；同时，既然“计算”已经升华为一种“思维”，也必然具有其思维属性。所以把握和研究计算思维，必须从思维的计算学科特征和计算的思维属性两方面入手，而不能拘泥于某一个简单的定义。研究计算思维，是研究计算学科惯用的思维方式，借鉴其积极、有益的成分，为解决社会实践问题服务。这才是让大众“像计算机专家一样思考”的根本用意。

2 计算思维的标志性概念

何以体现计算思维的计算学科特征？周以真提到“利用计算机科学的基础概念”。计算思维由计算学科抽象而来，而计算学科的基础概念是学科的重要组成部分，所以这些概念应该是计算学科特征的最好体现。但利用哪些基础概念？周没有明确，专门著述并不多见。国内除有学者给出了零散的叙述[15]外，另有学者以怀疑的态度提及九校联盟的两种截然不同的观点[16]，在某高校计算思维培养研究专题网站发布的问卷调查结果也折射出这两种观点[17]：一是“12 概念说”，认为包括绑定、大问题的复杂性、概念和形式模型、一致性和完备性、效率、演化、抽象层次、按空间排序、按时间排序、重用、安全性、折衷和结论；二是“25 概念说”，认为包括计算、通信、协调、记忆、自动化、评价、设计、约简、嵌入、转化、仿真、递归、并行、抽象、分解、保护、冗余、容错、纠错、系统恢复、启发式、规划 、学习、调度、折衷。稍作分析即可发现，“12 概念说”源自美国 ACM、IEEE 学会推出的计算学科课程体系 Computing Curricula 1991（CC1991）；而“25 概念说”的前 7 个概念源自美国计算机科学家、ACM 前主席 Peter Denning 的《伟大的计算原理》，后 18 个明显是从周以真原文筛选而来，毫无增减。

笔者认为，计算学科的基础概念很多，关键是搜寻标志性概念。Computing Curricula2004 已经承认其“很难再定义为一个单一的学科”、“涵盖了许多其他重要学科”[18]；CC1991 的 12 个核心概念实为 12 个核心问题，不具基础性，况且该提法已经过去 20 余年；一些权威著述提及的概念未必是专门针对计算思维提出的，可作参考，但不能简单照搬或挪用；即使是计算思维专门著述，也不能从中筛选词汇作为标志性概念。计算思维的标志性概念必须按照一定的原则或标准确立。它们应该具有以下特点：

①必须是计算学科术语，以便体现其计算特征。“效率”、“安全性”、“结论”等并非计算学科专用，不是标志性概念；“抽象”、“约简”、“绑定”、“转化”等动词根本不是术语，也不能作标志性概念；

②必须是基础术语，因为它们更便于理解，更便于计算思维的探究和推广。计算学科已经超越和突破计算机科学（CS）和计算机工程（CE）范畴[18]，而周以真仍强调概念的基础性，这恐怕是最根本的原因；

③必须有足够高的使用频率。使用频率高方可体现计算特征的共性，而计算思维势必蕴含在这些共性之中；

④体现一种学科理念或认知方式，这是其思维属性的根本体现，因而尤其重要；

⑤内涵和外延适度，不能处于计算学科知识树的根节点或叶节点。比如“计算”一词，太笼统，在“一切皆计算”的大背景下，很难把握其内涵，不能成为标志性概念。

当然，上述原则有其相对性，尤其是节点所在的层级直接影响到标志性概念的数量。关键是应该以计算理论或理论计算机科学为基础而不应涉及过多的分支。笔者尝试分出 6 个节点并列举了一些标志性概念（见表 1）。不难看出，结构化、模型等概念出现的频率很高。

表 1 几个节点下的标志性概念

节点	涉及科目	标志性概念举例
数据	数据结构、数据库	栈、树、队列、数据类型、完整/一致性、结构化、冗余、数据模型
算法	算法	穷举法、递推法、递归法、分治法、回溯法、查找、排序、复杂度
程序	程序设计	编码、范型、模块化、结构化、一致性、可移植性、可测试性、优化
软件	软件工程、软件过程	重用、建模、形式化方法、结构化方法、面向对象、可视化
系统	操作系统、信息系统等	进程、线程、并发控制、优先级
网络	互联网、网页设计、网络编程	协议、OSI 模型、拓扑结构、超链接

3 计算思维的典型思想

周以真在另一篇文章中谈到“计算是我们的抽象的自动化”。“抽象是我们的心智工具”；我们通过使我们的抽象、抽象层次及其联系机械化进行计算；由于计算学科具有精确、严格的记号和模型，机械化是可能的，自动化需要我们寻找某些“计算机”去解析抽象；这种计算机可能是人、机器、人机组合或网络。关键是“选择正确的抽象，选择正确的‘计算机’”[19]。由此，研究计算思维，重在寻找相应的“计算机”或抽象出相应的“机械化”。有学者指出了计算思维原理[20]，但其“能行性”有待验证；吴文虎等在归纳和阐释程序设计中常用的编程技巧时提出了“计算思维方式”[21]，其中，有些并无显著的计算学科特征，而“建立模型”、“分治”（算法范畴）的确是计算学科的典型做法；何明昕认为关注点分离是计算思维的重要原则之一，并指出分而治之、模块化是其另类表述[22]。赵岭忠等认为贪心策略、分治策略、动态规划、回溯和分枝限界策略很好地揭示了计算思维[23]；九校联盟编写的一些计算思维教材，多以算法知识作为重点内容。可见，许多学者已经注意到了算法在计算思维中的特殊意义和价值。

算法的确具有一些思维内涵，如递归法体现了“重复套用最初步骤的操作规律直至结束”的思想，分治法体现了“化整为零，各个击破”的思想。但是，穷举法、递推法、贪心算法一般都用于解决某类数学问题；随着算法难度加大，一些算法的应用反而愈发局限在更小的领域（如迭代法限于数值分析、分枝限界法用于组合爆炸问题），成为解决特定问题的唯一的方法或工具，思维属性愈发淡化。所以，“唯算法论”是狭隘的计

算思维观。计算思维是一种思维，而思维的结果是思想，其外在表现应该是谋略、对策、策略。研究计算思维的典型策略对于理解和把握计算思维的思维属性、培养和运用计算思维都有重要意义。笔者认为，计算思维的相关思想更广泛存在于一些标志性概念之中，根据表1中出现频率较高的一些标志性概念，至少可归结出以下几种典型思想。

3.1 结构化思想

1968年，高德纳著《计算机程序设计艺术》之第一卷《基本算法》，首次系统地阐述了数据的逻辑结构和存储结构及其操作技术，开创了数据结构的最初体系。这一成就不仅使之成为最年轻的图灵奖得主，而且直接催生了高德纳奖。此后，数据结构理论成为程序设计、编译程序、操作系统、数据库系统等系统程序的基础。结构化思想也成为计算学科的重要思想、最基本的策略之一。

结构化是一种格式化，也是一种规范化，有助于独立存在，便于访问，也有利于重用或共享。在硬件领域，二进制就是一种结构化，便于电子电路的实现，这是计算机科学的基础；在数据存储方面，数据类型是一种结构化；关系模型中的表也是一种结构化。数据项各有其名称、数据类型，数据类型的确立意味着其所占用的存储空间的确立，实现了数据存储的有序化，便于按名存取；而数据文件的结构化直接导致了从最初的文件系统向数据库系统的转变；在程序设计方面，Goto语句合理取舍便于保持各程序段的独立性，是结构化编码的基本要求；在软件开发领域，结构化是一种传统的软件开发方法，需要结构化分析、结构化设计、结构化实现；模块化是结构化的基本要求。而结构化查询语言(SQL)不仅上升为一种标准，可独立于各种程序设计语言和DBMS软件之外，而且统一了访问工具、简化了访问过程，极大地促进了数据的跨平台使用，可谓达到了数据重用或共享的最高境界。所以结构化是计算学科的最重要的思想，体现的是一种效率观念，是计算思维的重要组成部分。

3.2 形式化思想

大卫·希尔伯特在1900年提出的23个数学问题引领了数学和计算机科学的发展。20世纪30年代，为了从本质上说明什么是“可计算的”，邱奇(Alonzo Church)提出了λ算子，图灵(Alan Turing)提出了图灵机，克林(Stephen Kleenen)提出了一般递归函数[24]，这些奠定了形式化思想的基础。此后，先提出问题，而后用严谨的数学语言揭示事物的本质，就成为计算科学的最基本的方法——形式化方法(Formal Methods)。集合论、一阶谓词演算、时态逻辑是形式化的基本理论，模型、逻辑、代数、过程代数、网络是形式化的基础工具。形式化(Formality)的本意是“正式化”、“正规化”，体现了严谨的科学态度。

模型是形式化的基本手段，体现了抽象化、直观化(便于理解)，以及通用的思想。建模或建立模型的过程是抽象思维过程。其中，图模型应用最为广泛，如关系模型、ISO/OSI模型等。但也有许多文字模型，如网络协议等。其实，图灵机、元胞自动机、λ算子、一般递归函数都是模型的不同形式。如今，汉诺塔几乎成了递归问题的直观模型，五哲学家用餐问题几乎成了并发控制的模型。

形式化的过程是典型的思维过程，例如形式化方法中的形式规约(系统建模等)是关于一致性、完备性的考量过程；形式验证(软件模拟、测试)过程则是程序正确性的考量过程。它们都很好地体现了计算思维。

归结起来，程序设计语言是一种形式化，算法也是一种形式化。栈、队列、树、逻辑运算也都是形式化。这些已经成为计算学科的重要工具，而信息、学习、安全等词汇已经通过形式化的定义开辟了计算学科多个新的研究领域。可以说，形式化是计算学科重要的技术体系。形式化思想开创和发展了计算机科学。

3.3 最优化思想

冗余、复杂度、优化等概念突出体现了最优化思想。原本由人完成的工作通过程序或软件由计算机实现，以提高工作效率，这本身就是一种优化。不仅如此，优化思想进一步直接触及程序本身——提高程序运行效率。在复杂度概念并未普及时，在最早的程序设计中，人们已经通过减少循环次数来降低时间复杂度。后来优化思想发展到提高资源利用效率，在数据库理论中从第一范式到第五范式，都是为了降低数据冗余，节省存储空间。之后，最优化又从最初对人的要求逐渐变为系统的必备功能。例如降低复杂度、降低冗余、合理选择数据类型等原本都是对设计者的要求，后来随着优化规则和要求日益繁多，许多软硬件中已经能够自动实现一些相关性能，如自检、系统容错、纠错技术体现了系统自我完善的思想；可变长度字符串及变体数据类型体现的则是灵活、自适应思想，其本质也是最优化。再后来，优化还从内涵、本质逐渐延伸到形式、外观——软件界面从黑白字符到彩色GUI；如今，优化又使静态单调的数字演变出动态的可视化数据……。可见，优化思想不仅是计算学科的根本思想，也引领了学科的研究与发展。

3.4 联动化思想

HTML中的超链接体现的是文件、媒体、信息位置的联动；面向对象的编程体现的是对象与数据的联动；数据库中的事务管理体现的是事件联动，锁实现了联动的并发控制；参照完整性体现的是数据联动，有些软件中还提供了指针联动。

联动是一种整体性、一种相互关联性，可以提高访问速度，可以实现自动控制，保证数据的有效性。分析表明，绑定也是一种联动。从OOP、GUI的人机互动，到互联网的计算机软硬件的有线联动，再到物联网的人-机、人-物、物-物的无线联动……，联动化的形式和内涵正在不断翻新，联动化思想正在改变着世界。联动化思想的核心是寻找一种可靠的联动机制，实现一体化共赢。

结束语 理解计算思维的关键是把握其思维的计算学科特征和计算的思维属性。从理论计算机科学提取一些标志性概念的共性来领悟计算思维，可以发现结构化思想、形式化思想、最优化思想、联动化思想是计算思维的典型思想。一般而言，结构化、形式化、联动化是手段，最优化是目标。培养计算思维，应该从理解和运用标志性概念入手，学习和使用计算学科惯用的典型思想或策略。

参 考 文 献

[1] 百度百科. 莱布尼茨[OL]. http://baike.soso.com/v1957.htm, 2011-10-28

[2] Gardner H. The Disciplined Mind: Beyond Facts and Standardized Tests[M]. New York: Penguin Group (USA) Incorpora-

ted,2000:120-126
[3] Gardner H. Changing Minds: the Art and Science of changing our own and Other People's Minds[M]. New York: Penguin Group (USA) Incorporated,2006:89-96
[4] APA-CAP. The Digital Phoenix:How Computers Are Changing Philosophy[M]. New York: Springer Publishing Company, 1998:78-82
[5] 郝宁湘. 计算:一个新的哲学范畴[J]. 哲学动态,2000,18(11):32-36
[6] 李建会. 走向计算主义[J]. 自然辩证法通讯,2003,22(3):31-36
[7] Karp R. Understanding Science Through The Computational Lens[J]. Journal of Computer Science And Technology,2011,26(4):68-75
[8] 黄崇福. 信息扩散原理与计算思维机器在地震工程中的应用[M]. 北京:北京师范大学,1992:168-178
[9] 董荣胜,古天龙. 计算机科学与技术方法论[M]. 北京:人民邮电出版社,2002:98-110
[10] Wing J M. Computational Thinking[J]. Communications of the ACM,2006,22(9):45-49
[11] Cuny J, Snyder L, Wing J M. Demystifying CT For Non-Computer Scientists[J]. Work In Progress,2010,8(3):30-36
[12] ISTE. Operational Definition of Computational Thinking for k-12 Education[OL]. http://www. iste. /Libra ries/PDFs/Operational_Definition_of_Computational_Think ing. sflb. ashx,2012-04-15
[13] 王飞跃. 从计算思维到计算文化[J]. 中国计算机学会通讯,2007,17(11):10-15
[14] 爱因斯坦. 爱因斯坦文集[M]. 北京:科学出版社,1989:245
[15] 李廉. 计算思维——概念与挑战[J]. 中国大学教学,2012,7(1):7-12
[16] 谭浩强. 研究计算思维,坚持面向应用[J]. 计算机教育,2012,6(21):45-50
[17] 西安交通大学计算机教学实验中心. 国家级精品课《大学计算机基础》问卷调查统计数据[OL]. http://computer. xjtu. edu. cn/dcwj_l. htm,2012-5-17
[18] IEEE/ACM. Computing Curricula 2004[OL]. http://www. acmtyc. org,2005-6-21
[19] Wing J M. Computational Thinking and Thinking about Computing[J]. Philosophical Transactions of the Royal Society A, 2008,10(4),37-45
[20] 朱亚宗. 论计算思维——计算思维的科学定位、基本原理及创新路径[J]. 计算机科学,2009,12(4):53-56
[21] 吴文虎,王建德. 世界大学生程序设计竞赛高级教程—程序设计中常用的计算思维方式[M]. 北京:中国铁道出版社,2009,180-184
[22] 何明昕. 关注点分离在计算思维和软件工程中的方法论意义[J]. 计算机科学,2009,12(4):60-63
[23] 赵岭忠,钱俊彦,蔡国永. 算法设计策略与计算思维[J]. 企业科技与发展,2010,7(8):35-38
[24] 包云岗. 世纪图灵纪念[J]. 中国计算机学会通讯,2012,22(11):42-47

(上接第 6 页)

[28] Zhao Lei, Wang Li-na, Xiong Zuo-ting, et al. Execution-aware fault localization based on the control flow analysis[J]. Information Computing and Applications Lecture Notes in Computer Science,2010,6377:158-165
[29] Christ J, Ermis E, Schäf M, et al. Flow-sensitive fault localization[J]. Verification, Model Checking, and Abstract Interpretation Lecture Notes in Computer Science,2013,7737:189-208
[30] 许高阳,李必信,孙小兵,等. 一种基于层次切片的软件错误定位方法[J]. 东南大学学报:自然科学版,2010,40(4):692-698
[31] Zhang X, He H, Gupta N, et al. Experimental evaluation of using dynamic slices for fault location[C]// Proceedings of the 6th interna- tional symposium on Automated analysis-driven debugging. Monterey, California, USA,2005:33-42
[32] Zhang X, Gupta N, Guptaand R. Locating faulty code by multiple points slicing [J]. Software: Practice and Experience, 2007, 37:935-961
[33] Lei Yan, Mao Xiao-guang, Dai Zi-ying, et al. Effective fault localization approach using feedback[J]. IEICE Transactions on Information and Systems,2012,95(9):2247-2257
[34] Jiang Shu-juan, Li Wei, Li Hai-yang, et al. Fault localization for null pointer exception based on stack trace and program slicing [C]//Proceedings of the 12th International Conference on Quality Software. 2012:9-12
[35] Liu C, Fei L, Yan X, et al. Statistical debugging: a hypothesis testing-based approach[J]. IEEE Transactions on Software Engineering,2006,32(10):831-848
[36] Artzi S, Dolby J, Tip F, et al. Fault localization for dynamic web applications[C]//Proceedings of the 19th international symposium on Software testing and analysis. Trento, Italy, ACM,2012, 38:314-335
[37] http://sir. unl. edu/portal/index. html
[38] http://sourceforge. net
[39] 张云乾,郑征,季晓慧,等. 基于马尔可夫模型的软件错误定位方法[J]. 计算机学报,2013,36(2):445-456
[40] 衷璐洁,霍玮,李丰,等. 基于传播引擎的指针引用错误检测[J]. 计算机学报,2013,36(2):432-444
[41] Xie X Y, Chen T Y, Kuo F C, et al. A theoretical analysis of the risk evaluation formulas for spectrum-based fault localization [J]. ACM Transactions on Software Engineering and Methodology,2013
[42] Wong W E, Debroy V, Choi B. A family of code coverage-based heuri- stics for effective fault localization[J]. Journal of Systems and Software,2010,83(2):188-208
[43] 顾庆,唐宝,陈道蓄. 一种面向测试需求部分覆盖的测试用例集约简技术[J]. 计算机学报,2011,34(5):879-888
[44] Wen wan-zhi. Software fault localization based on program slicing Spectrum[C]// Proceedings of the 2012 International Conference on Software Engineering. 2012:1511-1514