

# 学科形态与核心概念典型方法与典型实例

# 计算科学导论

商建东 sjd@zzu. edu. cn 计算机与人工智能学院 国家超级计算郑州中心



- 每一个学科都有其自身的知识组织结构、学科形态、核心概念和基本工作流程方式。
- 所谓学科形态,是指从事该领域工作的文化方式。
- 对计算科学的深入研究使我们已知该学科存在三种主要的学科形态
  - 理论
  - 抽象
  - 设计



- 第一种形态: 理论
- 科学理论
  - 科学认识由**感性阶段**上升为**理性阶段**就形成了科学理论
  - 科学理论是经过实践检验的系统化了的**科学知识体系**, 它是由科学概念、科学原理,以及对这些概念原理的理论论证所组成的体系
  - **理论源于数学**,是从抽象到抽象的升华,它们已经完全脱离现实事物,不受现实事物的限制,具有精确的优美的特征, 因而更能把握事物的本质



- 计算学科的理论
  - 计算科学的数学基础和计算科学理论,广泛采用数学的研究方法.
  - 按统一的合理的理论发展过程,包含以下四个步骤:
  - (1) 对研究对象的概念抽象(定义和公理)
  - (2) 假设对象的基本性质和对象之间可能存在的关系(定理)
  - (3) 确定这些性质和关系是否正确(证明)
  - (4) 解释结果(与计算机系统或研究对象形成对应)
- 这个学科形态的基本特征是其研究内容的构造性数学特征,是区别于更广泛的数学科学学科形态的典型特征



#### 哲学与数学 (数学的作用)

- ◆ 数学的领域在扩大。哲学的地盘在缩小。
  - ▶ 哲学曾经把整个宇宙作为自己的研究对象。那时,它是包罗万象的,数学只不过是算术和几何而已。17世纪,自然科学的大发展使哲学退出了一系列研究领域,哲学的中心问题从"世界是什么样的"变成"人怎样认识世界"。这个时候,数学扩大了自己的领域,它开始研究运动与变化。
  - 今天,数学在向一切学科渗透,它的研究对象是一切抽象结构——所有可能的关系与形式。可是西方现代哲学此时却把注意力限于意义的分析,把问题缩小到"人能说出些什么"。
- ◆ 哲学应当是人类认识世界的先导,哲学关心的首先应当是科学的未知领域。
  - 哲学家谈论原子在物理学家研究原子之前,哲学家谈论元素在化学家研究元素之前,哲学家谈论无限与连续性在数学家说明无限与连续性之前。
  - 一旦科学真真实实地研究哲学家所谈论过的对象时,哲学沉默了。它倾听科学的发现,准备提出新的问题。



#### 哲学与数学(数学的作用)

模糊的哲学与精确的数学——人类的望远镜与显微镜。

- 哲学,在某种意义上是望远镜。当旅行者到达一个地方时,他不再用望远镜观察这个地方了,而是把它用于观察前方。
- 数学则相反,它是最容易进入成熟的科学,获得了足够丰富事实的科学,能够提出规律性的假设的科学。它好像是显微镜,只有把对象拿到手中,甚至切成薄片,经过处理,才能用显微镜观察它。
- ▶ <u>哲学从一门学科退出,意味着这门学科的诞生。数学渗入一门学科,甚至控制一门学科</u> 意味着这门学科达到成熟的阶段。(构建了该学科的理论体系)
- ▶ 哲学的地盘缩小,数学的领域扩大,这是科学发展的结果,是人类智慧的胜利。
- ▶ 但是,宇宙的奥秘无穷。向前看,望远镜的视野不受任何限制。新的学科将不断涌现,而在它们出现之前,哲学有许多事可做。面对着浩渺的宇宙,面对着人类的种种困难问题,哲学已经放弃的和数学已经占领的,都不过是沧海一粟。
- 哲学在任何具体学科领域都无法与该学科一争高下,但是它可以从事任何具体学科无法完成的工作,它为学科的诞生准备条件。
- 数学在任何具体学科领域都有可能出色地工作,但是它离开具体学科之后无法作出贡献。它必须利用具体学科为它创造条件。



### 学生选课问题

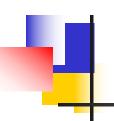
- 学生选课问题是一类数据处理问题的特例
  - 当然不能对特例建立一大套理论
  - 类似的数据处理问题涉及方方面面----值得建立理论,处理数据管理问题
- 数据库理论----为解决数据处理问题建立的
  - 关系模型----数据结构
  - 关系代数/关系演算----数据操作
  - 数据库设计理论
    - 数据依赖: 函数依赖, 多值依赖...
    - 范式: 六种范式: 第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、巴斯-科德范式(BCNF)、第四范式(4NF)和第五范式(5NF,又称完美范式)
    - 模式分解与规范化



- 第二种形态:抽象
- 科学抽象
  - 是指在思维中对同类事物去除其现象的次要的方面,抽取其共同的主要的方面,从而做到从个别中把握一般,从现象中把握本质的认知过程和思维方法
  - 抽象源于现实世界,源于经验,是对现实原形的理想化
  - 理想化后的现实原形与现实事物有了质的区别。但它们总是现实事物的概念化,有现实背景
  - 抽象是科学认识的基础和决定性环节
  - 学科中的抽象形态包含着具体的内容,它们是学科中所具有的科学概念、科学符号和思想模型



- 计算学科的抽象,或称模型化
  - 基于计算科学的实验科学方法,广泛采用实验物理学的研究方法。 按照对客观现象和规律的实验研究过程,包含以下四个步骤:
  - (1) 确定可能世界(环境)并形成假设
  - (2) 构造模型并做出预测;
  - (3) 设计实验并收集数据;
  - (4) 分析结果。
- 这个学科形态主要出现在计算科学中与硬件设计和实验有关的研究之中。当计算科学理论比较深奥,理解较为困难时,不少科研人员在大致了解理论、方法和技术的情况下,基于经验和技能常以这种学科形态方式开展工作



# 例: 四色问题





# 例: 四色问题(续)

- 任何规范的地图都可以至多用四种颜色着色,使得任何两个相邻的区域 都具有不同的颜色
  - 两个区域相邻是指它们有一条公共边
- 下面两个图都形状不完全相同,从着色角度是相同



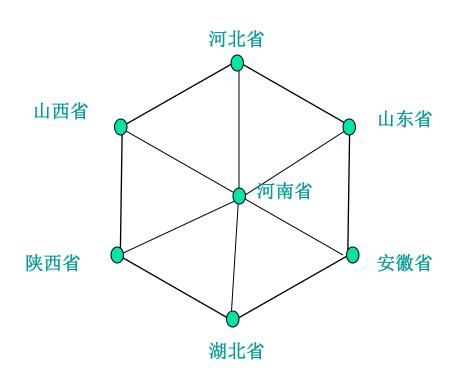




## 例: 四色问题(续)

- 抽象:将地图按如下方式转换
  - 顶点: 地图上每个区域
  - 边: 如果地图上的两个区域相邻,则对应的两个顶点之间有一条边
- 前面的例子转换成







## 例: 四色问题(续)

- 可以证明: 转换后的图没有交叉边----平面图
- 可以证明:任何平面图都可以至多用五种颜色着色,使得任意两个相邻的 顶点都具有不同的颜色
  - 可以用如下定理归纳地证明: 任何平面图都有小于5度的顶点
- 地图的四着色问题转换成
  - 任何平面图都可以至多用<mark>四种颜色</mark>着色, 使得任意两个相邻的顶点都 具有不同的颜色
- 转换后的平面图与原地图很不相同,但是从着色角度,两个问题是否有解 是等价的
- 抽象去除了问题的次要方面,如每个区域的形状,大小
- 抽象保留了问题的本质方面----区域的相邻性被顶点的相邻性捕获
- 抽象得到的平面图概念不仅可以解决地图着色问题,而且还有其他应用, 如电路板布线问题



- 第三种形态:设计
- 科学技术的设计问题
- 设计形态与抽象、理论两个形态存在的联系
  - 设计源于工程,并用于系统或设备的开发,以实现给定的任务
  - 设计必须以对自然规律的认识(科学形态的认识/经验形态的认识) 为 前提
  - 必须创造出相应的人工系统和人工条件,还必须认识自然规律在这些人工系统中和人工条件下的具体表现形式
- 设计形态的主要特征与抽象、理论两个形态的主要区别
  - 设计形态具有较强的实践性
  - 设计形态具有较强的社会性
  - 设计形态具有较强的综合性



- 计算学科的设计
  - 基于工程,广泛采用工程科学(如建筑工程)的研究方法。
  - 按照为解决某一个问题构建系统或装置的过程,包含以下四个步骤:
    - (1) 需求分析;
    - (2) 建立规格说明;
    - (3) 设计并实现该系统;
    - (4) 对系统进行测试和分析。
- 这个学科形态广泛出现在计算科学中与硬件、软件、应用有关的设计和实现之中。当计算科学理论(包括技术理论)已解决某一问题后,科研人员在正确理解理论、方法和技术的情况下,可以十分有效地以这种学科形态方式开展工作



- 3个学科形态的内在联系
  - 抽象源于现实世界,它的研究内容表现在两个方面
    - 建立对客观事物进行抽象描述的方法
    - 采用现有的抽象方法,建立具体问题的概念模型,从而实现对客观 世界的感性认识
  - 理论源于数学,它的研究内容也表现在两个方面
    - 建立完整的理论体系
    - 在现有理论的指导下,建立具体问题的数学模型,从而实现对客观 世界的理性认识
  - 设计源于工程,它的研究内容也表现在两个方面
    - 在对客观世界的感性认识和理性认识的基础上, 完成一个具体的任务
    - 对工程设计中所遇到的问题进行总结,提出问题,由理论界去解决它



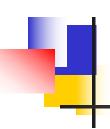
- 对现实世界中被研究的对象进行抽象,建立必要的基本概念,运用数学工具和方法对其进行基础和应用基础研究,研究(对象)概念的基本性质、概念与概念之间的关系,揭示对象发展变化的内在规律,为实验设计和工程设计实现提供方法和技术,并开展实验和工程设计与实现工作是计算科学学科的基本工作流程方式。其中,抽象(主要指抽象化过程)是理论、抽象和设计三个基本学科形态中最重要的一个形态,它是连接学科科学研究与工程应用开发研究的重要环节。
- 在计算科学的研究与开发中,三种学科形态并不是孤立出现的,它们常常交织在一起。经验告诉我们,没有抽象形态的支持,理论研究就失去了背景参照;没有理论的指导,工程开发往往会误入歧途。有许多例子分别表明,抽象和设计阶段出现了理论;理论和设计阶段需要模型化;而理论和抽象阶段离不开设计,它们必须考虑到现实是否能行。



在计算科学的发展中,有一批在各个分支学科中重复出现的概念。它们虽然在各学科中的具体解释在形式上有差异,但相互之间存在着重要的联系。核心概念是计算科学重要思想、原则、方法、技术过程的集中体现,有助于在学科的深层统一认识计算机科学。对核心概念的深入理解和正确拓展与应用的能力,是计算科学家和工程师成熟的标志之一。

核心概念是方法论的重要组成内容,一般具有如下特点:

- (1) 在本学科的不少分支学科中经常出现, 甚至在学科中普遍出现;
- (2) 在计算科学理论、抽象和设计这三个过程的各个层面上都有许多示例;
- (3) 在理论上具有可延展和变形的作用,在技术上具有高度的独立性。



下面我们筛选和分类给出计算科学的核心概念。当然,某一概念划入一个类并不说明它只在一些分支方向有效,学习中应注意这些概念在不同课程中的具体解释,最好能够通过思考,将这些概念串联起来,把自己所学的专业知识尽可能地系统化。

#### (1) 计算模型与能行性

计算模型 (Computational Model),可计算性 (Computability),计算复杂性 (Computational Complexity),最优性 (Optimum),相似性与对偶性 (Similarity and Duality);

#### (2) 抽象与构造性描述

论域与计算对象(Domain and Computing Object), 枚举(Enumeration)与有穷表示(Finite Representation), 分层与抽象的级(Hierarchy and Levels of Abstraction), 内涵与外延(Intension and Extension),递归(Recursion), 归纳(Induction),自由与约束(Freedom and Restriction);



#### (3) 系统特征

```
相容性(Consistency),完备性(Completeness),
单调性(Monotoneity),透明性(Transparence),
容错与安全性(Fault-Tolerant and Security),
开放性(Openness),稳定性(Stability),
健壮性(Robustness);
(4) 计算方法
折衷(Compromise),分解(Decomposition),
集成(Integration),类比(Analogy),
推导(Inference or Reasoning),变换(Transfomation),
扩展(Extension and Expansion);
```



#### (5) 实现技术

```
类型(Type),进程与线程(Process and Thread),顺序与并发(Sequence and Concurrent),软计算结点(Agent);现役(的)(Active),关联(Binding)与实例化(Instantiation),虚拟(的)(Virtual),编码(Coding),模式匹配(Pattern Matching),分权(Branching),合一(或通代)(Unification),协议(Protocol),循环与迭代(Loop and Iteration),重用(Reuse),规范与标准化(Standardization);
```



对于上面所讨论的计算科学中的三种学科形态和在学科中重复出现的核心概念,在将三个过程(学科形态)贯穿在整个学科各门课程教学的始终,用核心概念将各分支学科的内在联系串联起来时,事实上,学科形态与核心概念并不能完全解决问题,还需要于教学中介绍在这门学科中反复出现的具有哲学意义的典型方法,并通过一批典型实例给学生以较深刻的感性认识。

典型方法与典型实例是属于方法论的内容。就目前我们对学科的认识,已知有下列几种典型的方法和一大批典型实例。它们是:

- 1) 内涵与外延的方法
- 2) 以递归、归纳和迭代技术形式为代表的构造性方法
- 3) 公理化方法
- 4) 快速原型方法
- 5) 演化方法
- 6) 以及展开与规约方法



- 1) 内涵与外延是哲学的两个基本的概念。所谓内涵是指一个概念所反映的事物的本质属性的总和,也就是概念的内容。外延是指概念所界定的所有对象的集合,即所有满足概念定义属性的对象的集合。内涵与外延的方法广泛出现在计算科学的许多分支学科中。这是一个能够对无穷对象的集合作分类处理的方法。
- 2) 构造性方法是整个计算科学学科最本质的方法。这是一种能够对论域为无穷的客观事物按其有限构造特征进行处理的方法。
- 3)公理化方法也是计算科学的一种典型方法。它能帮助学生认识一个系统如何严格表述,认识到完备性和无矛盾性对一个公理系统的重要性,认识每一条公理深刻的背景,独立性和它的作用。可惜,其深刻的哲学意义、学术深度和理解上的困难性使得它在本科的课程中较少出现。



在学科的基础和应用基础的理论研究中,上述三种典型方法是比较常见的学科研究方法。下面,我们介绍在学科的应用研究的工程技术开发研究中使用较多的另外三种典型方法。当然,切不可由此认为某种典型方法仅仅适用于某一类研究而不适用于其他类研究,典型方法在学科的各个层面具有广泛的例示,只是在不同的类别中运用时有的多一些,有的少一些。

4)快速原型方法是计算科学的典型方法。最初,快速原型的思想出现在软件工程的研究之中,其主要内涵是:在软件的开发中,随着程序代码量的日渐庞大,开发费用和设计,自定义逻辑框架系统的研究,以及分布式代数系统的研究都采用了公理化方法或吸取了公理化方法的思想。随着学科发展的深化,预计这一方法还将在一些分支方向周期的不断增长,人们迫切需要对软件开发中引入的新思想、新原理和采用的新方法、新技术的可行性进行验证,通过验证过程提出改进意见,为实际产品的工程技术开发提供原理性的指导。此时,Prolog语言刚刚出现,研究人员惊喜地发现,该语言特别适合充当快速原型实验的验证工具。对于软件开发中系统设计的一种新的构思,



采用Prolog语言只对涉及新构思的内容"去粗取精"地开发原理性的验证程序,在证实了新构思确实可行之后,再按照规范的、工程化的要求进行软件的开发,并且在系统的开发中,基本上按照快速原型验证开发中设计的程序设计思想进行细化。这样做,不仅可以避免大型软件系统的开发出现原理性的错误,而且能够提高大型软件开发的速度和质量。由此不难看出,快速原型方法事实上是一种低成本系统原理验证性实验方法。

5)演化方法也是计算科学的典型方法。所谓演化(Evolutation)方法,也叫进化方法,是一种模拟事物演进过程进而求解问题的方法。其主要思想是,针对具体的问题,首先找到解决该问题的办法(或算法、程序、电路等),然后通过各种有效的技术方法改进解决问题的办法(或算法、程序、电路等)进而改进求解的结果。近年来在演化算法基础上发展起来的演化计算是该方法最具代表性的研究领域之一。例如,算法设计与分析中的代数化简方法,程序设计方法学中的程序变换方法,电路设计与分析中通过布尔函数的化简进而简化电路设计的方法,软件维护与升级等都是演化方法的具体应用实例。近年来,演化计算的内涵开始更多地指向以模拟大自然演化过程对一类复杂问题的计算。



演化方法在使用时常常与其他典型方法联系在一起。例如,与快速原型和展开方法合用,可以开发一类特殊软件的程序自动生成系统,如人机界面自动生成系统。

尽管演化方法实际上在计算科学的发展中早就出现了,其他学科也有类 似的方法,但是,学术界在相当长的一个时期内没有注意到该方法的方法学 意义。作者认为,演化方法是一种特别适合计算科学的技术方法。随着学科 研究与发展的不断深化,人们不可能在软硬件开发和应用中将所有的研究与 开发工作都做得十分圆满,万无一失。学科自身知识体系的复杂性决定了今 后在研究与开发中需要不断地对先前进行的工作进行改进,不断地完善,这 不仅仅是工程技术开发中一个具有普遍规律性的问题,而且,演化方法本身 还可能具有重大的、酝酿着学术研究突破价值的科学意义。从近年来演化计 算的研究(主要是算法演化计算研究)中出现广泛使用逻辑与代数方法的特 点考察和获得联想:如果在计算模型的研究中引入演化计算的方法改造传统 的计算模型,或设计新型的计算模型,那么,新计算模型有可能变得具有演 化计算能力,表现出某些健壮性的特征,这是否暗示了智能计算系统的研究 不必一定要沿着首先发展人工智能的逻辑基础,然后发展智能计算机、智能 计算系统的"演化"路线,而存在着一条崭新的、以前从未有人涉足过的研 究路线呢?



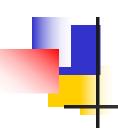
6)展开与归约是一对技术概念,是在处理实际事务的过程中对两个相向的处理活动所作的一般化的方法学概括。展开的内涵是从一个较为抽象的目标(对象)出发,通过一系列的过程操作或变换,将抽象的目标(对象)转换为具体的细节描述。例如,在程序设计方法学中,从一个程序的规范出发,运用程序推导技术,可以将一个程序一步一步地自动设计出来,这样就实现了对该程序的展开。理论上,一个具体的程序与该程序的规范指称的是同一个对象,语义完全相同,只是表现形式不同。从描述与理解的角度看,前者较为具体,后者较为抽象。规约方法可以视为展开过程的逆过程。

不难理解,基于程序规范的"自顶向下"的程序设计方法是一种展开方法,从一个电路系统布尔函数表达式出发到对应的电路系统的实际完成所进行的工作方法也是一种展开方法。"自底向上"的语法分析方法,语义约化方法都是规约方法的实例。



典型方法之所以重要,不仅在于它能深刻地揭示出学科的基本规律和各主科目之间内在的联系,而且在于它能启发人们更深入地思考学科的一些深层次的问题,培养学生的创造性能力和科学研究能力。

当然。计算科学中还有许多方法也是重要的,如类比推理、证伪发现、统计推断与分析、各种测试方法与测试数据筛选的方法等。但相对于上述 六种方法来说,这些方法应用范围比较单一、狭窄,或是哲学意义比较浅显。至于其它的典型方法,还有待人们去整理和总结。



#### 四、计算科学的典型实例

所谓典型实例是指那些反映学科某一方面内在规律和典型问题 本质内容的实例。典型实例也是方法论的重要组成内容,由于它们 往往以简化形式深入浅出地表达学科深奥的科学规律和学科典型问 题的本质内容,因此,在学科研究中常常被用来辅助说明思想、原 理、方法和技术,用来比较思想、理论、方法和技术的优劣。显然, 典型实例在计算科学发展中具有重要的作用。

典型实例在计算机软硬件的开发中具有重要的应用价值。我们知道,一个系统开发出来之后,面临着用户的确认和验收。对软硬件中的程序而言,测试程序只能发现程序有错,但不能证明程序无错。然而,采用形式化的方法证明程序的正确性又是一个十分困难的问题,不仅成本高、周期长,而且,对并发程序和并行程序的验证还有一些关键问题尚未解决,怎么办?显然,用长期积累的一批典型实例对程序进行测试是一个比较可行的折衷解决问题的办法,因为程序能够通过测试,意味着程序没有违背人类积累的本学科范围内的"常理",正确性在一定程度上比较可信。



#### 四、计算科学的典型实例

例 一组典型实例:停机问题,哲学家共餐问题,饮料问题,最小费用流问题,可满足性问题,货郎担问题,生产与消费问题,文法的二义性问题,上下文语义问题,最小生成树问题,阿克曼函数,鞍点问题,子集和数问题,确定性自动机与不确定性自动机的等价问题,八皇后问题,九宫排定问题,三个中国人算法,0/1背包问题,作业调度问题,路由选择问题,图的最小覆盖问题,荷兰国旗问题,等等。