

doi: 10.12052/gdutxb.160136

进化创新的绕行变换

陈文伟^{1,3}, 赵 侠², 黄金才³

(1. 海军陆战学院三系, 广东 广州 510430; 2. 国防科技大学 理学院;
3. 国防科技大学 信息系统与管理学院重点实验室, 湖南 长沙 410073)

摘要: 绕行变换是形象化的可拓变换。数学进化与计算机软件进化在创新中都采用了绕行变换。数学进化中的绕行变换有: 形式化方法、等价变换、变量数学、简化变换、求值变换。计算机软件的绕行变换是把复杂软件变换成基本运算, 使之适合于硬件完成。本文从软件的数值计算、数据存储和计算机程序3个方面进行了说明。
关键词: 绕行变换; 数学进化; 软件进化; 矛盾问题; 可拓变换
中图分类号: TP311.13 **文献标志码:** A **文章编号:** 1007-7162(2017)01-0001-05

Detour Transformation of Evolutionary Innovations

Chen Wen-wei^{1,3}, Zhao Xia², Huang Jin-cai³

(1. The Third Department, Naval Marine Academy, Guangzhou 510430, China; 2. School of Science, National University of Defense Technology; 3. Information System Engineering Key Laboratory, National University of Defense Technology, Changsha 410073, China)

Abstract: Detour transformation is visualization of Extension transformation, which is adopted in the innovation of evolution of mathematics and software. The forms of detour transformations in mathematics evolution are formalization, equivalent transformation and transforming transformation. The form of detour transformation in the computer software is to convert the complex software to the basic operations, making it suitable for hardware to accomplish. The software is described in three aspects, namely numerical calculation, data storage and computer program. The purpose of detour transformations is to change the complex contradiction problem into the easy solvable problem. The detour transformation is an effective extension transformation.
Key words: detour transformation; mathematics evolution; software evolution; contradiction problems; extension transformation

绕行变换是形象化的可拓变换^[1-3], 它把复杂的难解问题绕行到简单的可求解的问题。绕行变换在数学进化和软件进化中是一种典型的创新变换。目前, 对数学进化和软件进化的研究偏向于它们的历史, 而对于进化的规律研究太少。本文主要是探讨它们的进化规律。

1 数学进化中绕行变换

1.1 数学的形式化表示

形式化就是用数学符号体系, 去表达数学对象的结构和规律, 从而把对具体数学对象的研究转变

为对符号的研究^[4]。它更容易揭示事物的本质和规律性。形式化表示极大地帮助了数学的进化发展^[4-5]。欧洲古代数学的快速发展完全得益于采用了数学形式化方法。我国古代数学发展缓慢就在于没有采用数学形式化方法。

数学的进化包括: 数学符号的进化和数学理论的形成。

1.1.1 数学符号的进化

数学符号进化过程可以表述为

数字→算术→图形→微积分符号→表达式→方程式→程序。

数字是数学的原始表示,主要是印度-阿拉伯数字:0123456789. 数学符号还包括:变量符号(x, y, z)、运算符号($+$ $-$ $\times$ $\div$)、关系符号($=$ $<$ $>$ ∞)、微积分符号(d ∂ $\int$)、逻辑符号($\wedge$ $\vee$ $\sim$)等类型符号. 表达式和方程式是用数学符号组合而形成的,它是数学中最重要的表示形式;图形是空间的表现形式,主要用于解析几何和拓扑学中;程序是计算机解决数学问题的表示形式.

数学符号是数学的抽象语言,是文字的缩写,它是数学家们交流、传达和记录数学思维的简明话语. 它能精确、深刻地表达某种概念、方法和逻辑关系.

1.1.2 数学理论的形成

数学理论的形成过程可以概括为

形式化表示 $\rightarrow$ 推演求证 $\rightarrow$ 公式和定理 $\rightarrow$ 理论体系.

数学问题用表达式或方程式(符号的数学组合)进行表示,实质上完成了“从数学问题的自然语言描述到形式化表示的转换”,这是数学中最基本的一种绕行变换. 表示为

$T_{\text{form}}(\text{问题的自然语言描述}) = (\text{问题的形式化描述}).$

形式化表示便利了推演与求证,即利用形式化的等价变换,得到正确的公式或者证明了定理. 这种抽去了内容的正确结论,具有通用性,形成了数学理论. 数学中各门类的理论集合,形成了数学的理论体系.

1.2 等价变换

数学中一个重要的解题方法是用等价变换进行解析求解或者进行定理证明. 它是一种典型的绕行变换. 等价变换不是相同变换,等价变换的前后会发生变化,要么是数学结构元素的变化,要么是数值的变化. 等价变换是从已知条件或者是初始状态出发,利用保值的等价运算,通过一步一步的演算或者推导,最后得到所要的结果. 这一步一步的演算或者推导,就是在绕行变换,实现从已知到未知的变化,即解决已知和未知的矛盾.

1.2.1 数学结构中元素变化的等价变换

数学中绝大多数的运算都是数学结构的元素发生变化的等价变换,即相等运算的元素前后都发生了变化. 等价变换的目的是对问题求解,或者是进行定理证明.

例如,线代数方程组用矩阵表示为

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}.$$

线代数方程组利用消元法求解,通过对矩阵的元素进行选主元、主元归一和消元等步骤反复推演,最后得到具有单位矩阵的线性方程组,其中每一步相等运算前后的元素都发生了变化. 最后得到的具有单位矩阵的线性方程组,即

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \cdots \\ b'_n \end{pmatrix}.$$

它和原始的线性方程组在数学结构的元素上发生巨大的变化,这时已经求出方程组的解.

1.2.2 数值变化的等价变换

数学中选代法的等价变换,其运算的前后都不发生数学结构元素的变化,但是发生数值的变化. 经过成千上万次迭代,最后得到方程的解.

例如,BP神经网络中权值和阈值的求解的迭代法为

1) 网络权值的变换公式为 $w_{ij}^{(k+1)} = w_{ij}^{(k)} + \eta \delta_j x_i$;

2) 阈值的变换公式为 $\theta_j^{(k+1)} = \theta_j^{(k)} + \eta \delta_j$.

迭代公式每次迭代时公式形式不会发生变化,但每次迭代时元素的值在发生变化.

数学中的等价变换是解决“未知”与“已知”的问题. 等价变换是在数学原理(定义、定理等)或数学方法(消去法、迭代法等)的指导下进行的,通过等价推演或相等计算来完成从“未知”到“已知”的绕行变换.

1.3 从常量数学到变量数学的绕行变换

函数的出现以及微积分的发明就是从常量数学向变量数学的绕行变换结果.

1.3.1 函数和解析几何的出现

解析几何是笛卡儿开创的,他第一个建立了以 x, y 为轴的坐标系. 对于运动的点可以通过它与坐标的距离,来确定运动点的轨迹. 同时也出现了函数的概念,即一个变量 x 的变化引起另一个变量 y 的变化. 傅立叶发现函数可以用曲线表示,也可以用式子 $y=f(x)$ 表示. 从此数学由研究常量的初等数学,进入了研究变量的高等数学. 函数的进化过程表述为

函数 $\rightarrow$ 初等函数 $\rightarrow$ 复合函数 $\rightarrow$ 复变函数 $\rightarrow$ 函数方程 $\rightarrow$ 特殊函数.

函数的创造在数学中是具有重要意义的标志或里程碑.

1.3.2 微分和积分的发明

自变量的变化引起函数的变化,形成了微分. 微分学研究物体运动的瞬时速度、曲线的切线、函数的

极值等问题.微积分实质上是对函数的变换,即把一个函数变换成了另一个函数(导数),导数的数学表达式和函数变换分别表示分别为

$$\frac{d}{dx}f(x)=f'(x), \quad T_d(f(x))=f'(x).$$

随自变量的变化,对应变化的面积的积累,形成了积分.积分学解决计算曲线所围成的面积、曲面所围成的体积、曲线长、物体的中心等问题.积分实质是把一个函数变换成了另一个函数(原函数),积分的数学表达式和函数变换分别表示分别为

$$\int f(x)dx=g(x), \quad T_s(f(x))=g(x).$$

牛顿和莱布尼茨分别独立地建立了微积分.微积分的进化过程表述为

导数→微分→积分→常微分方程→偏微分方程→积分方程→变分方程.

导数表示当前变化的情况,积分表示长期积累的结果.微积分的创造是数学进化的重要里程碑,微积分的发展形成了“数学分析”的新领域.

微积分的价值在于用微分方程式表示自然现象和法则(多个变量变化时相互之间的关系)的内在本质.例如,流体力学中的纳维-斯托克方程和电磁学原理的麦克斯韦方程(历史上最伟大的10个方程之一)等都是用微分方程式表示.

1.3.3 无穷与极限的概念

无穷数有无穷大与无穷小,这是一个变化的概念,它困扰了数学界100多年,即“第二次数学危机”,直到柯西提出极限(limit)的观点,微积分的表示才确定下来,数学危机才得以解决.无穷数和极限自然引伸出的无限序列与无穷级数.这样,常量数学就绕行到了变量数学,开辟了新领域.

1.4 简化变换

简化变换是把复杂难于求解的函数或者是方程,转变成简单的可求解的函数或者是方程.这是一种有效解决问题的绕行变换.

1.4.1 傅里叶变换

傅立叶变换是将一个复杂函数变换成简单的正弦函数的累加.在物理学的信号处理中,傅立叶变换将随时间变化的复杂信号转换成了一系列正弦曲线之和,并计算其振幅和频率.这样就很容易对这些信号进行处理(如地震信号、JPEG数据压缩等处理).傅立叶变换是典型的绕行变换.傅立叶变换也被认为是主宰人类生活的7个方程之一.

1.4.2 拉普拉斯变换

当微分方程求不出解析解时,可以利用拉普拉斯变换(T_l)把它变换成代数方程,对代数方程求解,就容易求出它的解,再利用拉普拉斯逆变换(T_l^{-1}),把代数方程的解变换成微分方程的解,从而解决原微分方程求不出解析解的矛盾.用拉普拉斯变换表示为

$$T_l(\text{微分方程})=(\text{代数方程}), T_l^{-1}(\text{代数方程解析解})=(\text{微分方程解析解}).$$

拉普拉斯变换和逆变换都是典型的绕行变换.

1.4.3 微分方程的差分变换

所有不能进行解析求解的微分方程或积分方程,在变换成差分方程以后,都可以在计算机中进行数值求解.用绕行变换表示为

$$T_{\text{diff}}(\text{微分方程})=(\text{差分方程}).$$

差分方程经过整理后,就形成了线代数方程组,在计算机中就可以进行数值计算了,最后求得偏微分方程的数值计算结果.这也是一种简化的绕行变换.

1.4.4 数值计算的表达式变换

表达式有两种,一种进行算术运算,一种进行导数求解.这两种表达式计算,都对运算符有个优先顺序的规定,算术运算的优先顺序是“先乘除、后加减,括号优先”的原则;函数表达式求导数时,对运算符的优先顺序规定是,先对低级运算符(+、-)求导数,再对高级运算符($\times$ 、 $\div$)求导数.这种人为规定,不适合在计算机中编程序完成.这需要将表达式进行变换.把有优先顺序规定的表达式变换成只有前后顺序的表达式,这样才能编制程序在计算机中完成表达式的算术运算或求导数.

(1) 算术运算的逆波兰式变换

这种绕行变换实质上完成了表达式的中缀表达,变成了后缀表达,即逆波兰式.表示为

$$T_{\text{pola}}^{-1}(\text{中缀表达式})=(\text{后缀表达式}).$$

这样,编程序就按前后顺序进行算术运算.

(2) 求导数的波兰式变换

这种绕行变换实质上把函数表达式的中缀表达,变成了前缀表达,即波兰式.表示为

$$T_{\text{pola}}(\text{中缀表达式})=(\text{前缀表达式}).$$

这样,对函数的前缀表达式求导数时,程序按前后顺序进行求导数.

1.4.5 求值变换

数学主要是研究求解的方法(过程)或求值的

公式. 现代数学追求的目标是获得解析解(用公式表示的解). 对于具体的求值, 不深入讨论. 例如, 对于大型方程的求解, 现代数学不再过问了. 计算机的出现, 把问题的求解过程用程序来表达, 具体的计算交给计算机来做, 从而形成了计算数学. 计算数学不但把数学中的有效的求解方法加以实现, 而且把现代数学不能获得解析解的问题, 通过数值化, 求得满足精度要求的数值解. 例如, 有些偏微分方程得不出解析解, 把偏微分方程改变成差分方程, 再转化为线代数方程组, 这样就能够求出它的数值解.

对于现代数学的求解析解, 改变为计算数学的求数值解. 这种求值变换也是一种绕行变换, 它促成了数学史中最大规模的一次进化, 具体表示为

T_{means} (现代数学的解析解) = (计算数学的数值解).

计算机的发展推动了计算数学的发展, 计算数学属于数值计算. 数值计算的发展又推动了非数值计算的发展(数据库的数据处理和知识推理), 使计算机进入了个人和社会, 从而又推动了社会信息化进程.

2 软件的绕行变换

任何复杂的计算机软件, 都需要把它变换成计算机的基本运算上, 即适合计算机硬件能完成的基础上, 它才能在计算机上进行计算求解^[6]. 计算机硬件只会做加法运算, 其他任何运算, 只能通过变换到加法运算上来, 计算机才能计算. 这个变换用绕行变换来说明更合适. 绕行变换就是解决加法运算与其他数学运算之间的矛盾问题.

2.1 数值计算中矛盾问题的绕行变换

数值计算中主要是加减乘除运算(算术运算). 算术运算的加减乘除都要变换成加法运算. 加(+)是最基本运算; 减(-)是利用减数的补数(求反加1), 变减为加; 乘($\times$)是把乘变成累加; 除($\div$)是把除变成累减的次数.

但是, 有很多的数学计算不是加减乘除运算. 这时就要利用绕行变换把它变换成加减乘除运算, 解决算术运算和非算术运算的矛盾问题.

(1) 初等函数绕行变换成加减乘除运算

初等函数的定义不是算术运算, 为了让计算机计算, 需要利用初等函数的幂级数公式来计算, 即利用绕行变换成加减乘除运算.

(2) 微积分绕行变换成加减乘除运算

微分和积分的定义也不是算术运算, 是极限运

算. 为了让计算机计算, 利用绕行变换取消极限, 变换成加减乘除运算. 计算精度由自变量改变的大小(Δx)来控制.

2.2 数据存储的矛盾问题的绕行变换

数据分为结构化和非结构化两种. 数字、标点符号、英文字母称为结构化数据. 汉字、声音、图像、视频等称为非结构化数据.

非结构化数据和结构化数据是矛盾的, 绕行变换是把非结构化数据变为结构化数据^[7].

(1) 结构化数据进化的绕行变换

结构化数据的进化体现在数据存储量愈来愈大, 数据种类也愈来愈多, 这样使计算机处理问题的能力也愈来愈强. 它们都是要通过绕行变换为二进制数据. 表示为

T_{del} (结构化数据) = (二进制数据).

(2) 非结构化数据(多媒体)的绕行变换

非结构化数据(多媒体)本身是不能存入计算机的. 为了把它存入计算机, 需要对它作一个绕行变换, 必须把多媒体数据二值化(用于表示, 不进行运算).

最早把计算机程序存入计算机中, 就是把计算机程序用二进制程序来表示. 后来为解决汉字和多媒体存入计算机的问题, 再次采用二值数据表示汉字和多媒体, 从而进入了多媒体时代.

非结构化数据的绕行变换表示为

T_{del} (多媒体数据) = 二值数据.

2.3 计算机程序的矛盾问题的解决

(1) 计算机程序存储的绕行变换

计算机程序主要是描述对数据的操作过程. 如何把操作过程也存入计算机中, 这是冯·诺依曼的贡献. 他把操作过程称为程序, 由一串机器指令组成. 机器指令含操作码和地址码. 操作码表示对数据运算和对程序控制的动作. 地址码表示数据存放的地址或程序指令的存放地址, 它们均用二进制数据表示, 这样程序就很容易用二进制形式存入计算机中. 计算机称为存储程序式计算机, 原因在此. 二进制程序也称机器语言程序, 可直接在计算机上运行.

机器语言程序有两个重要特点: (1) 在数据地址码中只放数据, 不放运算符. 运算符都在操作码中, 即运算和数据是分开的; (2) 操作码中的指令是对变量的地址进行操作, 而不是直接对变量的操作, 这是一种间接操作. 可以说, 间接操作就是一种绕行变换. 表示为

T_{del} (计算机程序) = 二进制操作码对二进制地址码的操作.

(2) 高级语言程序的绕行变换

高级语言程序同样要经过绕行变换,回到二进制程序,这由编译程序来完成. 编译程序的技术原理相同于人工智能中的专家系统. 即利用文法(知识)对程序中的语句进行归约(反向推理)或推导(正向推理),既要检查语句是否符合文法,又要将语句编译成中间语言或机器语言.

高级语言程序绕行变换表示为

T_{del} (高级语言源程序) = 二进制程序.

高级语言程序设计了3种基本结构: 顺序、选择、循环,任何复杂的程序都是这3个基本结构的嵌套组合. 其绕行变换表示为

T_{del} (任何复杂的程序) = 顺序、选择、循环结构的嵌套组合.

这种程序设计方法保证了程序的正确性. 它克服了20世纪60年代的软件危机.

(3) 计算机程序的绕行变换的表示

计算机程序的绕行变换表示为

T_{del} (对数据的操作) = 对数据的地址操作;
 T_{del} (高级语言程序) = 二进制程序;
 T_{del} (任何复杂的程序) = 顺序、选择、循环的嵌套组合.

这些绕行变换都是把矛盾问题化繁为简、化难为易,最后解决这些矛盾问题.

3 结束语

在数学进化和计算机软件进化创新中,把复杂矛盾问题通过绕行变换成简单的、可运行的问题. 数学进化中的绕行变换是形式化方法、等价变换、变量数学、简化变换、求值变换等; 计算机软件的绕行变换是把非算术运算变换成算术运算(加减乘除);

任何多媒体数据变换成二值数据存储; 任何计算机程序变换成顺序、选择、循环的嵌套组合. 笔者在这里只总结了部分数学进化和计算机软件进化创新中的绕行变换内容^[4-6].

数学的辉煌成就和计算机软件的万能应用中,包含了很多绕行变换的创新需要去挖掘. 它可以帮助解决更复杂的数学问题,使计算机解决问题的能力进一步提升.

参考文献:

- [1] 陈文伟, 杨春燕, 黄金才. 可拓知识与可拓知识推理[J]. 哈尔滨工业大学学报, 2006, 38(7): 1094-1096.
CHEN W W, YANG C Y, HUANG J C. Extension knowledge and extension knowledge reasoning[J]. Journal of Harbin Institute of Technology, 2006, 38(7): 1094-1096.
- [2] 陈文伟, 黄金才, 毕季明. 适应变化环境的元知识的研究[J]. 智能系统学报, 2009, 4(4): 331-334.
CHEN W W, HUANG J C, BI J M. A study on meta-knowledge suitable for a changing environment[J]. Caai Transactions on Intelligent Systems, 2009, 4(4): 331-334.
- [3] 杨春燕, 蔡文. 可拓学[M]. 北京: 科学出版社, 2014.
- [4] CHEN W W, HUANG J C, ZHAO X. Evolutionary innovations of formalization and digitalization[M]//CAI W, YANG C Y, SMARANDACHE F, et al. Extenics and Innovation Methods. London: CRC Press, 2013, 89-94.
- [5] 陈文伟, 黄金才, 陈晟. 数学进化中的知识发现方法[J]. 智能系统学报, 2011, 6(5): 391-395.
CHEN W W, HUANG J C, CHEN S. A knowledge discovery approach in the evolution of mathematics[J]. Caai Transactions on Intelligent Systems, 2011, 6(5): 391-395.
- [6] 陈文伟, 陈晟. 计算机软件进化中创新变换和回归变换[J]. 广东工业大学学报, 2012, 29(4): 1-6.
CHEN W W, CHEN S. Innovation transformation and regression transformation in computer software evolution[J]. Journal of Guangdong University of Technology, 2012, 29(4): 1-6.
- [7] 陈文伟, 陈晟. 知识工程与知识管理[M]. 2版. 北京: 清华大学出版社, 2016.