软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(6):2208−2223 [doi: 10.13328/j.cnki.jos.006578]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 Coq 的杨忠道定理形式化证明*

严 升, 郁文生, 付尧顺

(天地互联与融合北京市重点实验室(北京邮电大学 电子工程学院), 北京 100876)

通信作者: 郁文生, E-mail: wsyu@bupt.edu.cn

摘 要: 实现拓扑学定理的机器证明, 是吴文俊院士生前的宿愿. 杨忠道定理涉及一般拓扑学中的诸多基本概念, 对深刻理解拓扑空间的本质有重要意义. 该定理表明, 拓扑空间中每一个子集的导集为闭集当且仅当此空间中的每一个单点集的导集为闭集, 是一般拓扑学中的一个重要定理. 基于定理证明辅助工具 Coq, 从公理化集合论机器证明系统出发, 对一般拓扑学中的开集、闭集、邻域、凝聚点和导集等拓扑基本概念进行形式化描述, 给出这些概念基本性质的形式化验证, 建立了拓扑空间的形式化框架. 在此基础上, 实现基于 Coq 的杨忠道定理形式化证明. 全部引理、定理和推论均完整给出 Coq 的形式化描述和机器证明代码, 并在计算机上运行通过, 体现了基于 Coq 的数学定理机器证明具有可读性、交互性和智能性的特点, 其证明过程规范、严谨、可靠. 杨忠道定理的形式化证明是一般拓扑学形式化内容的一个深刻体现.

关键词: Coq; 形式化证明; 公理化集合论; 一般拓扑; 拓扑空间; 杨忠道定理

中图法分类号: TP311

中文引用格式: 严升, 郁文生, 付尧顺. 基于 Coq 的杨忠道定理形式化证明. 软件学报, 2022, 33(6): 2208−2223. http://www.jos.org.cn/1000-9825/6578.htm

英文引用格式: Yan S, Yu WS, Fu YS. Formalization of C.T.Yang's Theorem in Coq. Ruan Jian Xue Bao/Journal of Software, 2022, 33(6): 2208−2223 (in Chinese). http://www.jos.org.cn/1000-9825/6578.htm

Formalization of C.T.Yang's Theorem in Coq

YAN Sheng, YU Wen-Sheng, FU Yao-Shun

(Beijing Key Laboratory of Space-ground Interconnection and Convergence (School of Electronic Engineering, Beijing University of Posts and Telecommunications), Beijing 100876, China)

Abstract: It was a wish for Academician Wu Wen-tsun to mechanically prove a class theorem in topology. The C.T.Yang's Theorem includes many basic concepts in general topology, which has great significance for understanding essential content of topological space. The C.T.Yang's Theorem is an important theorem in general topology, which states that in any topological space, if the derived set of a singleton set is closed, then the derived set of any subset is also closed. Based on the interactive theorem prover Coq, this paper presents a formalization of the basic concepts in general topology from mechanized axiomatic set theory, including open sets, closed sets, neighborhoods, condensation point, derived sets, and gives a formal verification of the corresponding properties. Furthermore, a formal framework of topological space is proposed and the formal proof of C.T.Yang's Theorem is realized in general topology. The proof code of all the theorems is given without exception, the formalization process has been verified, which reflects that the formal proof of mathematics theorem has the characteristics of readability and interactivity in Coq. The proof process is standardized, rigorous, and reliable, and the formal proof of C.T.Yang's Theorem is a profound embodiment of general topology formalization.

Key words: Coq; formal proof; axiomatic set theory; general topology; topological space; C.T.Yang's Theorem

人工智能研究是当前科技领域的前沿和热点, 也是我国科技战略发展的重要方向之一. 夯实人工智能基

* 基金项目: 国家自然科学基金(61936008)
本文由"定理证明理论与应用"专题特约编辑曹钦翔副教授、詹博华副研究员、赵永望教授推荐.
收稿时间: 2021-09-05; 修改时间: 2021-10-14; 采用时间: 2022-01-04; jos 在线出版时间: 2022-01-28

础理论尤为重要, 数学定理机器证明是人工智能基础理论的深刻体现, 受到学者的广泛关注[1−3].

数学定理的计算机形式化证明, 近年来随着计算机科学的迅猛发展, 特别是证明辅助工具 Coq[4−6]、Isabelle/HOL[7]、HOL Light[8,9]及 Mizar 等[10]的出现, 取得了长足的进展. 2005 年, 国际著名计算机专家 Gonthier 和 Werner 成功地基于 Coq 给出了著名的“四色定理”的形式化证明[11]. 进而, Gonthier 等人又经过 6 年努力, 于 2012 年完成对“有限单群分类定理”的形式化验证(该证明过程约有 4 300 个定义和 15 000 条定理, 约 170 000 行 Coq 代码)[12,13]. 2015 年, Hales 等人基于 HOL Light 和 Isabelle/HOL 等定理证明器完成了对“Kepler 猜想”的形式化验证[14]. 上述一系列著名数学难题形式化证明的实现, 使得各种证明辅助工具在学术界的影响日益增强[15].

Wiedijk 在文献[16]中指出, “数学历史上发生过 3 次革命: 第 1 次是公元前 3 世纪, 古希腊数学家欧几里德的《几何原本》引入数学证明方法; 第 2 次是 19 世纪柯西等人引入‘严格’数学方法以及后来的数理逻辑和集合论; 第 3 次就是当前正在进行的形式化数学”[17]. 全球各相关研究团队已经或计划完成包括 Gödel 不完备性定理、Jordan 曲线定理、素数定理以及 Fermat 大定理等在内的 100 个著名数学定理的计算机形式化证明. 相关网站[18]详细记录了 Wiedijk 所列出的 100 个著名数学定理在各类定理证明器下的完成情况, 目前共形式化验证 97 个, 其中, Coq 完成的有 75 个, Isabelle 完成的有 86 个, Mizar 完成的有 69 个. 大多数定理都是在多个定理证明器下进行了验证. 2002 年, 菲尔兹奖获得者 Voevodsky[19]和 Gowers[20]; 2010 年菲尔兹奖获得者 Villani 都大力倡导发展可信数学, 他们认为: 当今数学论证变得如此复杂, 而计算机软件能够检查卷帙浩繁的数学证明的正确性, 人类的大脑无法跟上数学不断增长的复杂性, 计算机检验将是唯一的解决方案. 英国帝国理工学院的数学教授 Buzzard 在剑桥举办的一次研讨会上表示: 证明是一种很高的标准, 我们不需要数学家像机器一样工作, 而是可以要他们去使用机器. Gowers 在文献[20]中指出: “21 世纪, 计算机在证明定理的过程中会起到巨大作用, 理论数学研究的模式将会彻底改观, 计算机的作用有可能超出我们现在的想象”. 甚至预测: “2099 年之前, 电脑或可完成所有重要的数学. 电脑会提出猜想、找到证明. 而数学家的工作是试着去理解和运用其中的一些结果”. 1987 年沃尔夫奖和 2005 年阿贝尔奖获得者 Lax[21]认为, “高速计算机对于应用数学和纯粹数学的影响可以与望远镜对天文学和显微镜对生物学的影响相比拟”. 今后, 每一本严谨的数学专著, 甚至每一篇数学论文, 都可由计算机检验其正确性, 这正发展为一种趋势.

实现拓扑学定理的机器证明, 是吴文俊院士生前的宿愿[22−24]. 杨忠道定理涉及一般拓扑学中的诸多基本概念, 对深刻理解拓扑空间本质有重要意义. 该定理表明, 拓扑空间中每一个子集的导集为闭集当且仅当此空间中的每一个单点集的导集为闭集, 是一般拓扑学中的一个重要定理. 杨忠道定理最早出现在 Kelley 的名著《General Topology》[25]中, 是由著名数学家杨忠道于 20 世纪 50 年代追随 Kelley 学习一般拓扑学时得到的[26]. 类似于数学分析中逐点收敛和一致收敛的关系, 拓扑空间中的集合可以看作是由单点集构成的, 如果将“单点集的导集为闭集”理解为局部的性质, “一个集合的导集为闭集”理解为整体的性质, 那么杨忠道定理很好地阐释了拓扑空间中局部与整体的关系. 该定理丰富了拓扑空间理论, 为拓扑空间中的收敛理论奠定了基础[26]. 此外, 数学家蒲保明、刘应明等人在模糊拓扑空间方面的突出贡献[27], 使得一般拓扑空间也不断向模糊拓扑空间、不分明拓扑空间及双模糊拓扑空间演变与发展[28,29]. 在这个过程中, 杨忠道定理得到了进一步的推广[30−32].

对杨忠道定理的机器证明, 曾振柄教授 2012 年指导博士生王建林在其学位论文中, 基于 Isabelle 首次实现了其形式化验证[33]. 2021 年, 他们将相关内容加以整理、改进, 正式发表于《中国科学》[24]. 本文基于交互式定理证明辅助工具 Coq, 实现杨忠道定理的机器证明. 本文作者研究团队前期已基于 Coq 定理证明器给出了 Morse-Kelley 公理化集合论机器证明系统[22]和分析基础的机器证明系统[23], 在此基础上, 可方便、快捷地实现现代数学中代数结构和拓扑结构的形式化构建.

本文在 Coq 定理证明器中对一般拓扑学中的杨忠道定理进行机器证明, 主要内容包括如下 3 个方面.

(1) 对集合论中集、空类、单点类、幂类、并、交、差、包含等基本定义、性质以及集合的基本运算进行形式化定义与验证;

(2) 对开集、闭集、邻域、凝聚点、导集等拓扑概念进行形式化描述, 完成拓扑空间的形式化构建, 并验证相关性质;

(3) 形式化构建杨忠道定理的相关引理, 实现杨忠道定理的机器证明.

其中, 上述第(1)方面、第(2)方面的形式化构建是杨忠道定理形式化证明的基础. 事实上, 为了实现杨忠道定理在计算机上的形式化描述与机器证明, 需要在机器上建立一般拓扑学中开集、闭集、邻域、凝聚点、导集等基本概念, 从而完成拓扑空间的形式化构建. 因此, 前两方面也是一般拓扑学形式化的基础, 利用 Coq 的推理规则和证明策略, 进而可以完成一批独立、有趣的拓扑学定理的机器证明. 从这个角度来看, 杨忠道定理机器证明的实现, 也标志着一般拓扑学系统形式化的一个开始[24].

本文实现基于 Coq 的杨忠道定理形式化证明, 全部引理、定理和推论均完整给出 Coq 的形式化描述和机器证明代码, 并通过了 Coq 定理证明器的检查验证, 体现了基于 Coq 的数学定理机器证明具有可读性、交互性和智能性的特点, 其证明过程规范、严谨、可靠. 杨忠道定理的形式化证明是一般拓扑学形式化内容的一个深刻体现, 该工作也为今后系统地实现代数拓扑和微分拓扑理论的形式化构建奠定了基础.

本文第 1 节预备知识介绍定理证明器 Coq 及一般拓扑学形式化相关工作. 第 2 节形式化定义并验证基础逻辑、集合论中的基本概念和性质. 第 3 节通过对拓扑基本概念的形式化描述, 完成拓扑空间的形式化构建, 并验证相关性质. 第 4 节完整实现杨忠道定理的形式化证明. 第 5 节总结全文并给出相关注记.

1 预备知识

本节介绍定理证明器 Coq 及一般拓扑学形式化的相关工作.

1.1 定理证明器Coq

Coq 是一个交互式的编译环境, 支持自动推理程序. Coq 通过命令式程序进行逻辑推导, 可以利用已证命题进行自动推理. Coq 中的归纳类型扩展了传统程序设计语言中有关类型定义的概念, 类似于大多数函数式程序设计语言中的递归类型定义[4−6]. Coq 有一支强大的全职研发队伍, 支持开源.

定理证明辅助工具 Coq 在推理和编程方面具有强大的表达能力, 从构造简单的项, 执行简单的证明, 到建立完整的理论, 学习复杂的算法, 对使用者的能力有着不同层次的需求[4,5]. Coq 的规范语言是 Gallina 语言, 该语言可以对程序设计中常用的类型和程序进行描述.

Coq 的基本原理是归纳构造演算, 用户对 Coq 中已证明定理的信心即来自于归纳构造演算的性质, 极大地扩展了传统程序设计语言中有关类型定义的概念. 在使用 Coq 的过程中, 必须遵守它的命令和语法习惯. 因为 Coq 是一个交互式的编译环境, 用户与机器之间是以人机交互的方式一问一答进行程序开发的, 用户在使用过程中可以边设计边修改, 以使得证明过程中出现的错误得到及时的修正[4,5].

在定理证明过程中, 需要用到一定的证明策略, 而证明策略是一条可以用在 Proof(证明)和 Qed(证毕)之间的指令, 它利用上下文中的声明、定义、公理、引理和一些已被证明的定理, 将待证目标分解为更为简单的子目标或直接证明该目标. 此外, Coq 允许用户使用 Ltac 命令定义新的证明策略, 这些扩展的策略可以使得后面的证明过程被更简洁地表达, 提高了系统的自动化程度, 这也体现了定理证明器 Coq 的智能性. 本文所用到的指令清单见表 1, 更详细和更多指令的功能可见文献[4−6].

表 1 Coq 常用指令简表

Coq 指令	用法
intro/intros	引入目标中的一个或多个条件
destruct	拆分条件中的析取式或实例化存在量词
assumption	遍历上下文寻找类型可转换为目标的假设
elim	用于归纳类型的消去证明策略
apply	应用一个假设或定理
eapply	apply 的变式, 可避免用 with 指令手动为 apply 指定变元
split	拆分目标中间的合取式
exfalso	将当前目标变成 False 类型

表 1　Coq 常用指令简表(续)

Coq 指令	用法
auto	自动重复执行 assumption、intro、apply 等策略
eauto	自动重复执行 assumption、intro、eapply 等策略
tauto	直觉命题演算自动证明策略
firstorder	一阶逻辑自动证明策略
left/right	当目标是析取式时, 得到析取式左边/右边的目标
rewrite	利用等式进行重写
exists	指定存在量词
pose proof	将已经证明的命题引入到条件中
generalize	引入上下文中的命题到目标中
assert/cut	生成新的子目标
subst	寻找条件中指定参数的等式, 等式替换出现该参数的地方
unfold	展开一个定义
repeat	无限重复指定策略直到失败或完全成功

1.2 一般拓扑学形式化研究现状

为完成杨忠道定理的形式化证明, 我们首先需要对拓扑空间进行形式化构建. 本节简要介绍一般拓扑学形式化的研究现状.

对于一般拓扑学的形式化研究, 国内外已有一些相关工作. Schepler 在 Coq 官网的贡献库中给出了一般拓扑学基本内容的形式化构建, 包括拓扑基本概念(开集、邻域、基与子基)和基本性质(连续性、分离性、紧致性)的形式化描述, 以及 Urysohn 引理和 Tietze 扩张定理等著名定理的机器证明[34], 是基于 Coq 的一般拓扑学形式化的代表性成果. 最近, Princeton 大学王盛颐博士也给出了一般拓扑学形式化框架的实现, 包括拓扑空间、拓扑子空间及度量空间等概念的形式化描述[35].

在使用定理证明器 Isabelle 的拓扑学形式化方面, Hölzl 等人在文献[36]中涉及了一些拓扑概念和拓扑空间的形式化描述. Gammie 和 Gioiosa 于 2017 年对一般拓扑学中的“Kuratowski 十四集定理”进行了机器证明[37]. Kuratowski 十四集定理与杨忠道定理一样, 都是拓扑空间中著名的定理[26]. 此外, Friedrich 也实现了一般拓扑学的形式化构建, 形式化证明了拓扑空间分离性相关性质, 并发展了 LList 拓扑的形式化[38].

杨忠道定理对深刻理解拓扑空间的本质有重要意义, 是一般拓扑学中的一个重要定理. 上述拓扑学形式化的工作均未涉及杨忠道定理, 目前仅在文献[24,33]中给出了基于 Isabelle 的杨忠道定理形式化证明. 现有的基于 Isabelle 的杨忠道定理形式化证明对一般拓扑学中拓扑空间基本概念的形式化具有较为重要的意义, 是对杨忠道定理形式化验证的一种有益尝试. 由于不同定理证明器的底层原理不同, 并且证明结果也不能相互共享, 因此, 基于定理器 Coq 对杨忠道定理进行形式化证明是十分必要的.

2　基础逻辑和集合论的形式化

集合论是现代数学的基础, 一般拓扑学的形式化需首先实现集合论的形式化. 事实上, Morse-Kelley 公理化集合论体系就是在 Kelley 的名著《General Topology》中以附录的形式正式发表的. Morse-Kelley 公理化集合论承认存在比集合更广的类, 采用无限的公理体系, 可“用来迅速而又自然地给出一个数学基础, 其中摆脱了明显的悖论”[25], 较通常的公理化集合论[39,40]应用起来更为便利.

本文从 Morse-Kelley 公理化集合论机器证明系统出发, 以熊金城所发表的文献[41]中的一般拓扑学理论体系为指导, 对有关集合论的一些基本知识进行形式化. 我们仅给出集合论形式化系统的部分内容, 以对杨忠道定理形式化证明的实现够用为限, 体现本文 Coq 代码的独立性和完整性.

集合论形式化系统中, 通过 Notation 指令添加了一些数学符号, 以使得形式化描述与证明过程更加接近数学语言, 增强代码可读性. 包括量词符号“$\forall$”表示任意, “$\exists$”表示存在; 逻辑符号“$\vee$”表示析取式, “$\wedge$”表示合取式, “$\neg$”表示逻辑非; “$\rightarrow$”表示蕴含, “$\leftrightarrow$”表示等价; 基本数学常项符号“=”表示等于, “$\neq$”表示不等于, “$\in$”表示属于, “\{P\}”表示满足性质 P 的类.

此外, 集合论形式化系统中需要一定的初等逻辑. Coq 系统中采用的是直觉主义逻辑, 即构造性逻辑, 在此逻辑中, 排中律是不成立的. 但是本文的工作需要排中律, 一种方法是调用 Coq 中 Classical 库实现. 为了本文代码的自封闭, 我们以公理的形式引入排中律, 其 Coq 代码如下:

Axiom classic: $\forall P$: *Prop*, $P\vee\neg P$.

利用上述公理, 可动态地根据给定命题的“真”或“假”返回需要的值, 为后续表达, 如分段函数的 Coq 代码描述或机器证明提供便捷. 因此, 本文采用的逻辑基础是直觉主义逻辑加上排中律. 此外, 从排中律还可推出一些常用的基本逻辑性质, 文中所涉及的初等逻辑命题 Coq 代码如下:

Proposition *NNPP*: $\forall P$, $(\neg(\neg P)\leftrightarrow P)$.

Proposition *inp*: $\forall P\ Q$: *Prop*, $(P\leftrightarrow Q)\rightarrow(\neg P\rightarrow\neg Q)$.

本文与现有基于 Coq 的拓扑学形式化工作最大的区别就是系统中对集合和元素的类型表示, 本系统中集合和元素的类型都是 *Class*, 在 Coq 形式化描述中用 *Type* 类型表示, 形式化描述如下:

Parameter *Class*: *Type*.

系统除了“=”与其他逻辑常项之外, 还有两个基本常项.

- 第 1 个常项是“∈”, 它读作“属于”. 由于该系统不区分集合和元素的类型, 故统一都用 *Class* 类型表示;
- 第 2 个常项是“{…:…}”, 它读作“{所有…的集使得…}”, 所以它是一种分类.

这两个常项对应数学符号的代码描述如下:

Parameter *In*: $Class\rightarrow Class\rightarrow Prop$.

Notation “$a\in A$”: $=(In\ a\ A)$ (at level 70).

Notation “$a\notin A$”: $=(\neg(a\in A))$ (at level 70).

Parameter *Classifier*: $\forall P$: $Class\rightarrow Prop$, *Class*.

Notation “\{P\}”: $=(Classifier\ P)$ (at level 0).

下面我们给出 Morse-Kelley 公理化集合论完整的公理体系及基本定义(集、非空类、空类、全域、单点类、包含、幂类、并、交、差、类的元的并、类的元的交、关系、函数)的形式化描述.

Definition *Ensemble* x: *Prop*: $=\exists y, x\in y$.

Axiom *ExtAx*: $\forall A\ B$: *Class*, $A=B\leftrightarrow(\forall x, x\in A\leftrightarrow x\in B)$.

Axiom *ClaAx*: $\forall x\ P$, $x\in$\{P\}$\leftrightarrow Ensemble\ x\wedge(P\ x)$.

Definition *NoEmpty* A: $=\exists x, x\in A$.

Notation “$\odot A$”: $=(NoEmpty\ A)$ (at level 45).

Definition *Empty*: =\{$\lambda x, x\neq x$\}.

Notation “$\varnothing$”: $=Empty$.

Definition μ: =\{$\lambda x, x=x$\}.

Definition *Singleton* x: =\{$\lambda z, x\in\mu\rightarrow z=x$\}.

Notation “$[x]$”: $=(Singleton\ x)$ (at level 0, right associativity).

Definition *Included* $A\ B$: $=\forall x, x\in A\rightarrow x\in B$.

Notation “$A\subset B$”: $=(Included\ A\ B)$ (at level 70).

Axiom *SubAx*: $\forall x$, $Ensemble\ x\rightarrow\exists y, Ensemble\ y\wedge(\forall z, z\subset x\rightarrow z\in y)$.

Definition *PowerSet X*: $=\backslash\{\lambda A, A\subset X\backslash\}$.
Notation "$\mathscr{P}(X)$": =(*PowerSet X*) (at level 9, right associativity).

Definition *Union A B*: $=\backslash\{\lambda x, x\in A\vee x\in B\backslash\}$.
Notation "$A\cup B$": =(*Union A B*) (at level 65, right associativity).
Axiom *UnionAx*: $\forall X\ Y$, *Ensemble X*→*Ensemble Y*→*Ensemble*$(X\cup Y)$.

Definition *Inter A B*: $=\backslash\{\lambda x, x\in A\wedge x\in B\backslash\}$.
Notation "$A\cap B$": =(*Inter A B*) (at level 60, right associativity).

Definition *Setmin A B*: $=\backslash\{\lambda x, x\in A\wedge x\notin B\backslash\}$.
Notation "$A-B$": =(*Setmin A B*).

Definition *EleU* $\mathscr{A}$: $=\backslash\{\lambda z, \exists y, z\in y\wedge y\in\mathscr{A}\backslash\}$.
Notation "$\cup\mathscr{A}$": =(*EleU* $\mathscr{A}$) (at level 66).
Axiom *EleUAx*: $\forall A$, *Ensemble A*→*Ensemble*$(\cup A)$.

Definition *EleI* $\mathscr{A}$: $=\backslash\{\lambda z, \forall y, y\in\mathscr{A}\to z\in y\backslash\}$.
Notation "$\cap\mathscr{A}$": =(*EleI* $\mathscr{A}$) (at level 66).

函数的相关定义及代换公理、正则性公理、无限性公理、选择公理对杨忠道定理的机器证明是非必需的, 为了完整形式化 Morse-Kelley 公理体系, 我们给出对应的形式化代码.

Definition *Unordered x y*: $=[x]\cup[y]$.
Notation "$[x|y]$": =(*Unordered x y*) (at level 0).

Definition *Ordered x y*: $=[[x]|[x|y]]$.
Notation "$[x,y]$":= (*Ordered x y*) (at level 0).

Definition *Relation r*: $=\forall z, z\in r\to\exists x\ y, z=[x,y]$.
Definition *Function f*: =*Relation f*$\wedge\forall x\ y\ z, [x,y]\in f\wedge[x,z]\in f\to y=z$.

Definition *Value f x*: $=\cap\backslash\{\lambda y, [x,y]\in f\backslash\}$.
Notation "$f[x]$": =(*Value f x*) (at level 5).

Definition *Domain f*: $=\backslash\{\lambda x, \exists y, [x,y]\in f\backslash\}$.
Notation "*dom*(f)": =(*Domain f*) (at level 5).

Definition *Range f*: $=\backslash\{\lambda y, \exists x, [x,y]\in f\backslash\}$.
Notation "*ran*(f)": =(*Range f*) (at level 5).

Axiom *RepAx*: $\forall f$, *Function f*→*Ensemble dom*(f)→*Ensemble ran*(f).
Axiom *RegAx*: $\forall x, x\neq\varnothing\to\exists y, y\in x\wedge x\cap y=\varnothing$.

Axiom *InfAx*: $\exists y$, *Ensemble* $y\wedge\varnothing\in y\wedge(\forall x, x\in y\rightarrow(x\cup[x])\in y)$.

Definition *ChoiceFunction* c: =*Function* $c\wedge\forall x, x\in dom(c)\rightarrow c[x]\in x$.

Axiom *ChoAx*: $\exists c$, *ChoiceFunction* $c\wedge dom(c)=\mu-[\varnothing]$.

需要指出的是: 该公理体系可迅速而自然地给出一个数学基础, 摆脱了较为明显的悖论[25]. 集的定义 *Ensemble* 和分类公理 *ClaAx* 对于消除朴素集合论中明显的逻辑悖论是关键的[22]. 上述公理对文中涉及到的一些“类”是“集”的证明是必不可少的.

集合论中基本定义所对应的性质是容易验证的, 这些性质在杨忠道定理的机器证明过程中会反复使用, 在 Coq 中是以 *Fact* 的形式呈现, 我们已完成这些性质的形式化验证, 完整的代码源文件可见 https://github.com/BalanceYan/CT.Yang.

下面给出集合论中一些基本性质的 Coq 代码描述.

- 空类的性质: 任何类不是空类的成员; 任何等于空类的类与该类是非空类的否命题等价; 任何不等于空类的类是非空类.

  Fact *EmptyNI*: $\forall x, x\notin\varnothing$;

  Fact *EmptyEq*: $\forall x, x=\varnothing\leftrightarrow\neg(\circledcirc x)$;

  Fact *EmptyNE*: $\forall x, x\neq\varnothing\leftrightarrow\circledcirc x$;

- 单点类的性质: 任何集是其单点类的成员; 若任意类 y 是集 x 的单点类的成员, 则类 x 与类 y 相等.

  Fact *SingI*: $\forall x$, *Ensemble* $x\rightarrow x\in[x]$;

  Fact *SingE*: $\forall x\ y$, *Ensemble* $x\rightarrow y\in[x]\rightarrow y=x$;

- 包含关系具有自反性、反对称性及传递性.

  Fact *ReSyTrP*: $\forall A\ B\ C, (A\subset A)\wedge(A\subset B\rightarrow B\subset A\rightarrow A=B)\wedge(A\subset B\rightarrow B\subset C\rightarrow A\subset C)$;

- 并、交、差这 3 种基本运算有幂等律、交换律、分配律及 De Morgan 律等性质.

  Fact *Idem*: $\forall A, A\cup A=A$;

  Fact *Idem'*: $\forall A, A\cap A=A$;

  Fact *Commu*: $\forall A\ B, A\cup B=B\cup A$;

  Fact *Commu'*: $\forall A\ B, A\cap B=B\cap A$;

  Fact *Distribu*: $\forall A\ B\ C, (A\cup B)\cap C=(A\cap C)\cup(B\cap C)$;

  Fact *DistribuLI*: $\forall A\ B\ C, A\cap(B\cup C)=A\cap B\cup A\cap C$;

  Fact *TwDeMorgan*: $\forall A\ B\ C, A-(B\cap C)=(A-B)\cup(A-C)$;

- 并、交、差这 3 种运算与空类有着以下基本性质.

  Fact *EmUnion*: $\forall A, A\cup\varnothing=A$;

  Fact *EmInter*: $\forall A, A\cap\varnothing=\varnothing$;

  Fact *SetminId*: $\forall X, X-X=\varnothing$;

  Fact *SetminEm*: $\forall X, X-\varnothing=X$;

- 并、交、差这 3 种运算与包含关系有着以下基本性质.

  Fact *TwSetmin*: $\forall A\ X, A\subset X\rightarrow X-(X-A)=A$;

  Fact *IncludP*: $\forall A\ B\ X, A\subset X\rightarrow A-B\subset X$;

  Fact *IncludP1*: $\forall A\ B\ C, A\subset B\rightarrow A-C\subset B-C$;

  Fact *IncludP2*: $\forall A\ X, X-A\subset X$;

- 由类的元的交与类的元的并推广的 De Morgan 律.

  Definition *AAr* $A\ \mathscr{A}$: $=\backslash\{\lambda z, \exists Ar, Ar\in\mathscr{A}\wedge z=A-Ar\backslash\}$;

  Fact *DeMorganUI*: $\forall A\ \mathscr{A}$, *Ensemble* $A\rightarrow\mathscr{A}\neq\varnothing\rightarrow(A-\cup\mathscr{A})=\cap(AAr\ A\ \mathscr{A})$.

表 2 列出了本文所涉及的集合论中一些重要概念的 Coq 描述、数学含义及性质.

表 2　集合论系统重要概念

Coq 描述	数学含义	基本性质
Empty	空类	$x\notin\varnothing$; $x=\varnothing\leftrightarrow\neg(\circledcirc x)$; $x\neq\varnothing\leftrightarrow\circledcirc x$
Singleton x	x 的单点类	$x\in[x]$; $y\in[x]\rightarrow y=x$
Included A B	A 包含于 B	$A\subset A$; $A\subset B\rightarrow B\subset A\rightarrow A=B$; $A\subset B\rightarrow B\subset C\rightarrow A\subset C$
PowerSet X	X 的幂类	$Y\in P(X)\leftrightarrow Y\subset X$
Union A B	A 与 B 的并	$A\cup A=A$; $A\cup B=B\cup A$; $A\cup\varnothing=A$
Inter A B	A 与 B 的交	$A\cap A=A$; $A\cap B=B\cap A$; $A\cap\varnothing=\varnothing$
Setmin A B	A 与 B 的差	$X-X=\varnothing$; $X-\varnothing=X$; $A\subset X\rightarrow X-(X-A)=A$
EleU/EleI $\mathscr{A}$	$\mathscr{A}$的元的并/交	$\mathscr{A}\neq\varnothing\rightarrow(A-\cup\mathscr{A})=\cap(AAr\ A\ \mathscr{A})$

3　拓扑基本概念的形式化

拓扑基本概念的形式化是杨忠道定理机器证明的基础, 这一节将基于定理证明器 Coq 给出拓扑基本概念及性质的形式化, 包括开集、邻域、导集及闭集等基本概念及对应性质的形式化描述与证明. 为了更加完整地理解拓扑空间形式化内容, 也给出了如平庸空间、离散空间等特殊拓扑空间的例子, 这些工作对需要学习拓扑学基本内容及 Coq 基本语法的读者是有帮助的.

为了便于阅读杨忠道定理形式化证明涉及的定理及引理, 本文首先给出相关定理的人工描述, 并以定理 3.1、定理 3.2、…的格式标记; 再给出其 Coq 代码的描述, 用 Theorem3_1、Theorem3_2,…这样的形式来表示; 定理的 Coq 证明代码可见源文件. 定理机器证明过程中涉及到的引理及推论不单独进行标号, 以其性质或对应定理的编号进行扩展命名, 如 LeTh3_3 表示的是定理 3.3 的第 1 个引理.

本节是杨忠道定理机器证明的预备工作, 文中给出了全部定义、推论、引理及定理的形式化描述. 形式化证明过程代码量较大, 简明起见, 未在文中列出, 详细的证明过程可见代码源文件.

3.1　拓扑空间与开集的形式化

拓扑空间的定义有很多种方式, 包括开集、闭集、导集、闭包、邻域、边界等方法, 都可以进行拓扑空间的定义. 在拓扑学的早期发展阶段, 德国数学家 Hausdorff 利用邻域公理定义了拓扑空间、波兰数学家 Kuratowski 利用闭包公理定义了拓扑空间. 随着拓扑学的不断发展, 人们发现上述不同定义出发的拓扑空间之间彼此都是等价的, 并且开集是探讨拓扑空间最简单、最便捷的工具[26]. 因此, 从开集公理进行拓扑空间的定义被广泛接受, 当下主流的拓扑学文献也是倾向于用开集公理进行拓扑空间的定义[25,41]. 本文与文献[41]中拓扑空间的定义一致, 也是通过开集对拓扑空间进行定义.

定义 3.1. 设 X 是一个集合, $\mathscr{T}$是 X 的一个子集族. 如果 $\mathscr{T}$满足如下条件:

(1)　$X, \varnothing\in\mathscr{T}$;

(2)　若 $A, B\in\mathscr{T}$, 则 $A\cap B\in\mathscr{T}$;

(3)　若 $\mathscr{T}_1\subset\mathscr{T}$, 则$\cup\mathscr{T}_1\in\mathscr{T}$,

则称 $\mathscr{T}$是 X 的一个拓扑.

如果 $\mathscr{T}$是 X 的一个拓扑, 则称偶对$(X,\mathscr{T})$是一个拓扑空间, 或称集合 X 是一个相对于拓扑 $\mathscr{T}$而言的拓扑空间; 或者当拓扑 $\mathscr{T}$早有约定或在行文中已有说明而无须指出时, 称集合 X 是一个拓扑空间. 此外, $\mathscr{T}$的每一个元素都叫作拓扑空间$(X,\mathscr{T})$ (或 X)中的一个开集.

Definition *Topology* $X\ \mathscr{T}$: $=\mathscr{T}\subset\mathscr{P}(X)\wedge X\in\mathscr{T}\wedge\varnothing\in\mathscr{T}\wedge$

$(\forall A\ B, A\in\mathscr{T}\rightarrow B\in\mathscr{T}\rightarrow A\cap B\in\mathscr{T})\wedge(\forall\mathscr{T}_1, \mathscr{T}_1\subset\mathscr{T}\rightarrow\cup\mathscr{T}_1\in\mathscr{T})$.

从拓扑空间的定义可以得出如下结论: (1) 拓扑空间 X 本身是开集; (2) 空集是开集; (3) 拓扑空间中任意两个开集的交是开集, 还可进一步推广成拓扑空间中任意有限个开集的交是开集; (4) 拓扑空间中任意开集族的并是开集.

为了更好地理解基于 Coq 的拓扑空间理论, 我们给出一些特殊拓扑空间的例子. 对于任意集合 X, 都有两

个特殊的拓扑空间, 一个是拓扑 $\mathscr{T}$ 只包含 X 本身和$\varnothing$的平庸拓扑, 另一个是拓扑 $\mathscr{T}$ 包含 X 的所有子集的离散拓扑. 在这里, 我们分别给出平庸拓扑和离散拓扑的定义及它们也是拓扑空间的形式化描述过程.

定义 3.2. 设 X 是一个集合, 若 X 的子集族 $\mathscr{T}=\{X,\varnothing\}$, 则称 $\mathscr{T}$ 是 X 的一个平庸拓扑.

Definition *Ordinary X*: =[*X*]∪[∅].

定理 3.1. 设 $\mathscr{T}$ 是 X 的一个平庸拓扑, 则$(X,\mathscr{T})$是一个拓扑空间.

Theorem *Theorem*3_1: ∀*X*, *Ensemble X*→*Topology X*(*Ordinary X*).

定义 3.3. 设 X 是一个集合, 若 X 的子集族 $\mathscr{T}=\mathscr{P}(X)$, 则称 $\mathscr{T}$ 是 X 的一个离散拓扑.

Definition *Discrete X*: =𝒫(*X*).

定理 3.2. 设 $\mathscr{T}$ 是 X 的一个离散拓扑, 则$(X,\mathscr{T})$是一个拓扑空间.

Theorem *Theorem*3_2: ∀*X*, *Ensemble X*→*Topology X*(*Discrete X*).

关于平庸拓扑空间和离散拓扑空间有许多有趣的性质. 集合 X 的离散拓扑和平庸拓扑分别是关于 X 的最大和最小拓扑, 因此集合 X 的每一个拓扑均包含平庸拓扑且包含于离散拓扑; 离散拓扑空间中的每一个子集都是开集. 此外, 还有如既不是平庸空间又不是离散空间、有限补空间及可数补空间等特殊的拓扑空间, 受篇幅所限, 这里不一一列举, 感兴趣的读者可基于本系统在 Coq 中对上述性质作进一步的探索与验证.

3.2 邻域的形式化

邻域是拓扑空间中十分重要的一个概念, 可以用来定义拓扑空间, 也可以用来定义拓扑空间中映射在某一点处的连续性, 接下来我们按照文献[41]中的理论体系给出拓扑空间中邻域、邻域系及开邻域的定义及基本性质, 并给出对应的 Coq 代码描述.

定义 3.4. 设$(X,\mathscr{T})$是一个拓扑空间, $x\in X$. 如果 U 是 X 的一个子集, 满足条件: 存在一个开集 $V\in\mathscr{T}$, 使得 $x\in V\subset U$, 则称 U 是点 x 的一个邻域. 点 x 的所有邻域构成的 X 的子集族称为点 x 的邻域系. 易见: 如果 U 是包含着点 x 的一个开集, 那么它一定是 x 的一个邻域. 于是, 我们称 U 是点 x 的一个开邻域.

Definition *TNeigh x U X* 𝒯: =*Topology X* 𝒯∧*x*∈*X*∧*U*⊂*X*∧∃*V*, *V*∈𝒯∧*x*∈*V*∧*V*⊂*U*.

Definition *TNeighS x X* 𝒯: =\{λ *U*, *TNeigh x U X* 𝒯\}.

Definition *TONeigh x U X* 𝒯: =*TNeigh x U X* 𝒯∧*x*∈*U*∧*U*∈𝒯.

从定义 3.4 可以得出邻域的两个基本性质, 我们以推论的形式在 Coq 中加以呈现.

推论 1. 拓扑空间中包含着点 x 的开集, 一定是点 x 的一个邻域.

Corollary *TNeighP*: ∀*x U X* 𝒯, *Ensemble X*→*Topology X* 𝒯→*x*∈*U*→*U*∈𝒯→*TNeigh x U X* 𝒯.

推论 2. 任意一个邻域都存在一个包含于它的开邻域.

Corollary *TNeighP*1: ∀*x U X* 𝒯, *Ensemble X*→*TNeigh x U X* 𝒯→∃*V*, *TONeigh x V X* 𝒯∧*V*⊂*U*.

定理 3.3. 拓扑空间 X 的一个子集 U 是开集的充分必要条件是 U 是它的每一点的邻域, 即只要 $x\in U$, U 便是 x 的一个邻域.

该定理在证明过程中需要补充一个引理: 任意一个集合等于集合中每个元的单点集的并组成的集合:

Lemma *LeTh*3_3: ∀*U*, *U*=∪(\{λ *t*, ∃*x*, *x*∈*U*∧*t*=[*x*]\}).

证明定理 3.3 的过程中需要构造每一个 x 存在一个开集 U_x, 通过如下定义进行构造:

Definition U_x *x U* 𝒯: =∪\{λ *V*, *x*∈*U*∧*V*∈𝒯∧*x*∈*V*∧*V*⊂*U*\}.

定理 3.3 的形式化描述如下所示, 通过引理 LeTh3_3 和定义 U_x, 可以完成该定理的证明.

Theorem *Theorem*3_3: ∀*U X* 𝒯, *Ensemble X*→*Topology X* 𝒯→*U*⊂*X*→
(*U*∈𝒯↔∀*x*, *x*∈*U*→*U*∈*TNeighS x X* 𝒯).

定理 3.4 概括了邻域系的基本性质.

定理 3.4. 设 X 是一个拓扑空间. 记 U_x 为点 $x\in X$ 的邻域系, 则有:

(1) 对于任何 $x\in X$, $\mathscr{U}_x\neq\varnothing$, 并且如果 $U\in\mathscr{U}_x$, 则 $x\in U$;

(2) 如果 $U,V\in\mathscr{U}_x$, 则 $U\cap V\in\mathscr{U}_x$;

(3) 如果 $U\in\mathscr{U}_x$, 并且 $U\subset V$, 则 $V\in\mathscr{U}_x$;

(4) 如果 $U\in\mathscr{U}_x$, 则存在 $V\in\mathscr{U}_x$ 满足条件: (i) $V\subset U$; (ii) 对于任何 $y\in V$, 有 $V\in\mathscr{U}_y$.

Theorem *Theorem*3_4a: $\forall x\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to x\in X\to$
$TNeighS\ x\ X\ \mathscr{T}\neq\varnothing\wedge(\forall U, U\in TNeighS\ x\ X\ \mathscr{T}\to x\in U)$.

Theorem *Theorem*3_4b: $\forall x\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to x\in X\to$
$(\forall U\ V, U\in TNeighS\ x\ X\ \mathscr{T}\to V\in TNeighS\ x\ X\ \mathscr{T}\to U\cap V\in TNeighS\ x\ X\ \mathscr{T})$.

Theorem *Theorem*3_4c: $\forall x\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to x\in X\to$
$\forall U\ V, U\in TNeighS\ x\ X\ \mathscr{T}\to V\subset X\to U\subset V\to V\in TNeighS\ x\ X\ \mathscr{T}$.

Theorem *Theorem*3_4d: $\forall x\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to x\in X\to$
$\forall U, U\in TNeighS\ x\ X\ \mathscr{T}\to\exists V, V\in TNeighS\ x\ X\ \mathscr{T}\wedge V\subset U\wedge$
$(\forall y, y\in V\to V\in TNeighS\ y\ X\ \mathscr{T})$.

3.3 导集的形式化

本节通过凝聚点的定义给出导集的定义.

定义 3.5. 设 X 是一个拓扑空间, $A\subset X$. 如果点 $x\in X$ 的每个邻域 U 中都有 A 中异于 x 的点, 即 $U\cap(A-\{x\})\neq\varnothing$, 则称点 x 是集合 A 的一个凝聚点或极限点. 集合 A 的所有凝聚点构成的集合称为 A 的导集, 记为 $d(A)$. 如果 $x\in A$ 并且 x 不是 A 的凝聚点, 即: 存在 x 的一个邻域 U, 使得 $U\cap(A-\{x\})=\varnothing$, 则称 x 为 A 的一个孤立点.

Definition *Condensa* $x\ A\ X\ \mathscr{T}$: $=Topology\ X\ \mathscr{T}\wedge A\subset X\wedge x\in X\wedge\forall U, TNeigh\ x\ U\ X\ \mathscr{T}\to U\cap(A-[x])\neq\varnothing$.

Definition *Derivaed* $A\ X\ \mathscr{T}$: $=\backslash\{\lambda\ x, Condensa\ x\ A\ X\ \mathscr{T}\backslash\}$.

从定义 3.5 可以得出导集的两个基本性质, 我们以推论的形式在 Coq 中加以呈现.

推论 1. 拓扑空间中任意一个子集的导集包含于这个拓扑空间.

Corollary *DerivaedP*: $\forall A\ X\ \mathscr{T}$, $Derivaed\ A\ X\ \mathscr{T}\subset X$.

推论 2. 设 X 是一个拓扑空间, $C\subset X$, 若 $x\in X$ 且 x 不是 C 的凝聚点, 则 x 有一个邻域 U, 使得 $U\cap(C-\{x\})=\varnothing$.

Corollary *DerivaedP*1: $\forall x\ C\ X\ \mathscr{T}$, $Topology\ X\ \mathscr{T}\to C\subset X\to x\in X\to$
$x\notin Derivaed\ C\ X\ \mathscr{T}\to\exists U, TNeigh\ x\ U\ X\ \mathscr{T}\wedge U\cap(C-[x])=\varnothing$.

定理 3.5 概括了导集的基本性质.

定理 3.5. 设 X 是一个拓扑空间, $A\subset X$, 则有:

(1) $d(\varnothing)=\varnothing$;

(2) $A\subset B$ 蕴含 $d(A)\subset d(B)$;

(3) $d(A\cup B)=d(A)\cup d(B)$;

(4) $d(d(A))\subset A\cup d(A)$.

Theorem *Theorem*3_5a: $\forall X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to Derivaed\ \varnothing\ X\ \mathscr{T}=\varnothing$.

Theorem *Theorem*3_5b: $\forall A\ B\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to$
$A\subset X\to B\subset X\to A\subset B\to Derivaed\ A\ X\ \mathscr{T}\subset Derivaed\ B\ X\ \mathscr{T}$.

Theorem *Theorem*3_5c: $\forall A\ B\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to A\subset X\to B\subset X\to$
$Derivaed(A\cup B)\ X\ \mathscr{T}=Derivaed\ A\ X\ \mathscr{T}\cup Derivaed\ B\ X\ \mathscr{T}$.

Theorem *Theorem*3_5d: $\forall A\ X\ \mathscr{T}$, $Ensemble\ X\to Topology\ X\ \mathscr{T}\to A\subset X\to$
$Derivaed(Derivaed\ A\ X\ \mathscr{T})\ X\ \mathscr{T}\subset A\cup Derivaed\ A\ X\ \mathscr{T}$.

此外, 凝聚点和导集还有许多其他的性质, 比如: 在离散空间中, 任何一个子集都没有凝聚点, 从而离散空间中任何一个子集的导集都是空集. 这些性质都可以在 Coq 中进一步作形式化验证, 以提高对拓扑空间的理解与认识.

3.4 闭集的形式化

闭集可以从导集的角度进行定义, 也可以直接从开集出发进行定义. 本节是按照文献[41]中闭集的概念

进行定义.

定义 3.6. 设 X 是一个拓扑空间, $A\subset X$. 如果 A 的每一个凝聚点都属于 A, 即 $d(A)\subset A$, 则称 A 是拓扑空间 X 中的一个闭集.

Definition *Closed* $A\ X\ \mathscr{T}$: $=$*Topology* $X\ \mathscr{T}\wedge A\subset X\wedge$*Derivaed* $A\ X\ \mathscr{T}\subset A$.

根据上述定义易知: 离散空间中的任何一个子集既是开集也是闭集, 而平庸空间中 X 本身和$\varnothing$既是开集也是闭集.

接下来是关于闭集的两个重要定理的形式化描述.

定理 3.6. 设 X 是一个拓扑空间, $A\subset X$, 则 A 是一个闭集当且仅当 A 的补集 A'是一个开集.

Theorem *Theorem*3_6: $\forall A\ X\ \mathscr{T}$, *Ensemble* $X\rightarrow$*Topology* $X\ \mathscr{T}\rightarrow A\subset X\rightarrow$*Closed* $A\ X\ \mathscr{T}\leftrightarrow X-A\in\mathscr{T}$.

定理 3.7 概括了闭集的基本性质.

定理 3.7. 设 X 是一个拓扑空间, 记 $\mathscr{F}$为所有闭集构成的族, 则有:

(1) $X, \varnothing\in\mathscr{F}$;

(2) 如果 $A, B\in\mathscr{F}$, 则 $A\cup B\in\mathscr{F}$(从而, 如果 $A_1,A_2,\dots,A_n\in\mathscr{F}$, $n\geqslant 1$, 则 $A_1\cup A_2\cup\dots\cup A_n\in\mathscr{F}$);

(3) 如果$\varnothing\neq\mathscr{F}_1\subset\mathscr{F}$, 则$\cap\mathscr{F}_1\in\mathscr{F}$.

证明定理 3.7 的过程中, 需要先定义拓扑空间 X 中所有闭集组成的集族 $\mathscr{F}$, 通过如下定义进行构造:

Definition $\mathscr{F}X\ \mathscr{T}$: $=\backslash\{\lambda\ U, U\subset X\wedge X-U\in\mathscr{T}\backslash\}$.

易得, $\mathscr{F}$也是拓扑空间 X 的一个子族.

Corollary *cFP*: $\forall X\ \mathscr{T}$, *Ensemble* $X\rightarrow\mathscr{F}X\ \mathscr{T}\subset\mathscr{P}(X)$.

定理 3.7 的形式化描述如下所示, 通过已证引理和定义 $\mathscr{F}$可完成该定理的证明.

Theorem *Theorem*3_7a: $\forall X\ \mathscr{T}$, *Ensemble* $X\rightarrow$*Topology* $X\ \mathscr{T}\rightarrow X\in\mathscr{F}\ X\ \mathscr{T}\wedge\varnothing\in\mathscr{F}\ X\ \mathscr{T}$.

Theorem *Theorem*3_7b: $\forall A\ B\ X\ \mathscr{T}$, *Ensemble* $X\rightarrow$*Topology* $X\ \mathscr{T}\rightarrow A\in\mathscr{F}\ X\ \mathscr{T}\rightarrow B\in\mathscr{F}X\ \mathscr{T}\rightarrow A\cup B\in\mathscr{F}\ X\ \mathscr{T}$.

Theorem *Theorem*3_7c: $\forall\mathscr{F}_1\ X\ \mathscr{T}$, *Ensemble* $X\rightarrow$*Topology* $X\ \mathscr{T}\rightarrow\mathscr{F}_1\neq\varnothing\rightarrow\mathscr{F}_1\subset\mathscr{F}X\ \mathscr{T}\rightarrow\cap\mathscr{F}_1\in\mathscr{F}X\ \mathscr{T}$.

4 杨忠道定理的形式化证明

本节将在拓扑基本概念形式化的基础上给出杨忠道定理的机器证明, 从而更加深入地理解拓扑空间. 由于杨忠道定理的机器证明是本文的核心, 本节给出详细的机器证明过程.

首先给出杨忠道定理的一个引理: 拓扑空间中, 任意一个开集与闭集的差集为开集.

该引理可通过集合的基本运算及定理 3.6 完成证明, 其形式化证明过程如下.

Lemma *OpenClosedP*: $\forall A\ B\ X\ \mathscr{T}$, *Ensemble* $X\rightarrow$*Topology* $X\ \mathscr{T}\rightarrow A\in\mathscr{T}\rightarrow$*Closed* $B\ X\ \mathscr{T}\rightarrow A-B\in\mathscr{T}$.

```
Proof with eauto.
  intros * Hxe Ht Ha Hb. pose proof Hb as [_[Hbx_]].apply Theorem3_6 in Hb...
  assert(A∩X−B∈𝒯).apply Ht... assert(A∩X−B=A−B).
  {AppE.apply InterIE in H0 as []; apply SetminIE.apply SetminIE in H1 as [].tauto.
    apply SetminIE in H0 as [].apply InterIE.split... apply SetminIE.split...
    apply Ht, PowerIE in Ha...} rewrite←H0...
Qed.
```

现在给出杨忠道定理的人工证明, 并对证明过程中的每个断言进行编号, 机器证明过程将遵循人工证明思路进行.

杨忠道定理. 拓扑空间中, 每一个子集的导集为闭集当且仅当此空间中的每一个单点集的导集为闭集.

证明思路:

- 必要性: 设拓扑空间中, 每一个子集的导集为闭集, 因为拓扑空间中每一个单点集也是拓扑空间中的子集, 所以易证此空间中的每一个单点集的导集为闭集;

- 充分性: 设拓扑空间$(X,\mathscr{T})$中每一个单点集的导集为闭集. 对任意的集合$A\subset X$, 由定义 3.5 导集的第 1 个推论可得 $d(A)\subset X$, 根据定义 3.6 可知, 接下来只需证明 $d(d(A))\subset d(A)$即可.
  (1) 设 $x\in d(d(A))$, 则由定义 3.5 可知: 对任意的邻域 U, 有 $U\cap(d(A)-\{x\})\neq\varnothing$. 因为每一个开邻域也是一个邻域, 故上述命题可推广为: 对任意开邻域 U, 有 $U\cap(d(A)-\{x\})\neq\varnothing$;
  (2) 虽然一个点的邻域不一定为开集, 但每一个开集一定是其内每一点的邻域, 又一个点的每一个邻域均包含该点的一个开邻域, 故对点 x 的任意一个邻域 U', 存在一个包含于 U'的开邻域 U(定义 3.4 的第 2 个推论);
  (3) 由假设条件可知, $d(\{x\})$是闭集;
  (4) 由已知条件易得, $x\notin d(\{x\})$;
  (5) 令 $V=U-d(\{x\})$. 因 U 是开集, $d(\{x\})$是闭集, 由引理 *OpenClosedP* 可进一步推出 V 是 x 的开邻域;
  (6) 将步骤(1)作用在步骤(5)上, 得 $V\cap(d(A)-\{x\})\neq\varnothing$;
  (7) 因此, 存在 $y\in V\cap(d(A)-\{x\})$, 进一步可得 $y\in V$, $y\notin d(\{x\})$, 且 $y\neq x$;
  (8) 将定义 3.5 的第 2 个推论作用在 $y\notin d(\{x\})$上, 则存在 y 的邻域 W 且 $x\notin W$;
  (9) 由引理 *OpenClosedP* 可进一步推出 V 是 y 的邻域;
  (10) 由于 W 和 V 都是 y 的邻域, 令 $K=W\cap V$, 由定理 3.4 可知, K 也是 y 的邻域;
  (11) 由步骤(7)得 $y\in d(A)$, 由定义 3.5 凝聚点的定义与步骤(10)可得, 存在 $z\in K\cap(A-\{y\})\neq\varnothing$;
  (12) 由 $z\in K\subset W$ 可知 $z\neq x$, 这是因为 $z\in W$ 而 $x\notin W$;
  (13) 因此, $z\in U\cap(A-\{x\})$. 这一步是因为 $z\in K\cap A=(W\cap V)\cap A\subset V\cap A=(U-d(\{x\}))\cap A$ 和 $z\neq x$;
  (14) 故 $U\cap(A-\{x\})\neq\varnothing$. 因为 U 是 x 的任意一个邻域, 必有 $x\in d(A)$. 所以 $d(d(A))\subset d(A)$, $d(A)$为闭集.

基于上述工作, 可以给出基于 Coq 的杨忠道定理的形式化证明过程.

```
Theorem CTYang: ∀X 𝒯, Ensemble X→Topology X 𝒯→
  (∀A, A⊂X→Closed(Derivaed A X 𝒯) X 𝒯)↔(∀x, x∈X→Closed(Derivaed([x]) X 𝒯) X 𝒯).
Proof with eauto.
  intros * Hxe Ht.split; intros Hp.
  - intros.apply Hp.intros z Hz.apply SingE in Hz.subst... Ens.
  - intros * Ha.split... split.apply DerivaedP.intros x Hx.apply DerivaedIE in Hx as [_[_[Hx Hxp]]].
    assert(Hop: ∀U, TONeigh x U X 𝒯→U∩Derivaed A X 𝒯–[x]≠∅).
    {intros.apply Hxp.apply H.} clear Hxp.apply DerivaedIE.split... split... split... intros U′ Hu′.
    red in Hu′.apply TNeighP1 in Hu′ as [U [Huo Huu]]... apply EmptyNE.pose proof Hx as Hx′;
    apply Hp in Hx; clear Hp.assert(Hxn: x∉Derivaed([x]) X 𝒯).
    {intro.apply DerivaedIE in H as [_[_[_H]]].pose proof Ht as [_[HX_]].
      pose proof TNeighP _ _ _ _ Hxe Ht Hx′ HX.apply H in H0.elim H0.AppE; [|exfalso0].
      apply InterIE in H1 as [].apply SetminIE in H2 as [].tauto.}
    pose proof Huo as [[_[_[Hxu_]]]_].set(V: =U–Derivaed([x]) X 𝒯).
    assert (Huv: V⊂U).{apply IncludP.intros z...} assert(Hv: TONeigh x V X 𝒯).
    {destruct Huo as [[_[_[_[Z [Hz [Hxz Huz]]]]]] [_Huo]].split.split... split... split.eapply ReSyTrP...
      exists(Z–Derivaed([x]) X 𝒯).split.eapply OpenClosedP... split.apply ClaI.Ens.tauto.
      eapply IncludP1... split.apply SetminIE.split... eapply OpenClosedP...}
    pose proof Hv as Hv′.apply Hop, EmptyNE in Hv as [y Hv].apply InterIE in Hv as [Hyv Hyp].
    apply SetminIE in Hyv as [Hyu Hyd].apply SetminIE in Hyp as [Hya Hxy].
    assert(Hnq: y≠x).{intro.elim Hxy.apply ClaI; Ens.}
    assert(Hw: ∃W, W∈TNeighS y X 𝒯∧x∉W).
```

{apply *DerivaedP*1 in *Hyd* as [*W* [*Hw He*]]... exists *W*.split.apply *TNeighSIE*...

apply *InterEqEmI* in *He*... intro.assert(*W*∩[*x*]≠∅).

{apply *EmptyNE*.exists *x*.apply *InterIE*.split... apply *SingI*.Ens.}

tauto.*Ens*.intros *z Hz*.apply *SingE* in *Hz*.subst... Ens.}

destruct *Hw* as [*W* [*Hw Hxw*]].assert(*Hv*: *V*∈*TNeighS y X* 𝒯).

{apply *TNeighSIE*... split... split... split.apply *IncludP*... exists(*V*−*Derivaed*([*x*]) *X* 𝒯).split.

eapply *OpenClosedP*; try apply *Hv*′... split.apply *SetminIE*.split... apply *ClaI*; Ens.apply *IncludP*2.}

set(*K*: =*W*∩*V*).assert(*Hk*: *K*∈*TNeighS y X* 𝒯).apply Theorem3_4b...

apply *TNeighSIE* in *Hk*... apply *DerivaedIE* in *Hya*.apply *Hya*, *EmptyNE* in *Hk* as [*z Hz*].

apply *InterIE* in *Hz* as [*Hz Hzp*].apply *SetminIE* in *Hzp* as [*Hza Hzy*].exists *z*.apply *InterIE*.

assert(*Hnq*1: *z*≠*x*).{apply *InterIE* in *Hz* as [*Hz*_].intro.subst; tauto.}

split.apply *InterIE* in *Hz* as []... apply *SetminIE*.split... intro.apply *SingE* in *H*.tauto.Ens.

Qed.

图 1 是杨忠道定理在证明辅助工具 Coq 中的交互式证明窗口. 至此, 完成了基于定理证明器 Coq 的杨忠道定理的形式化证明.

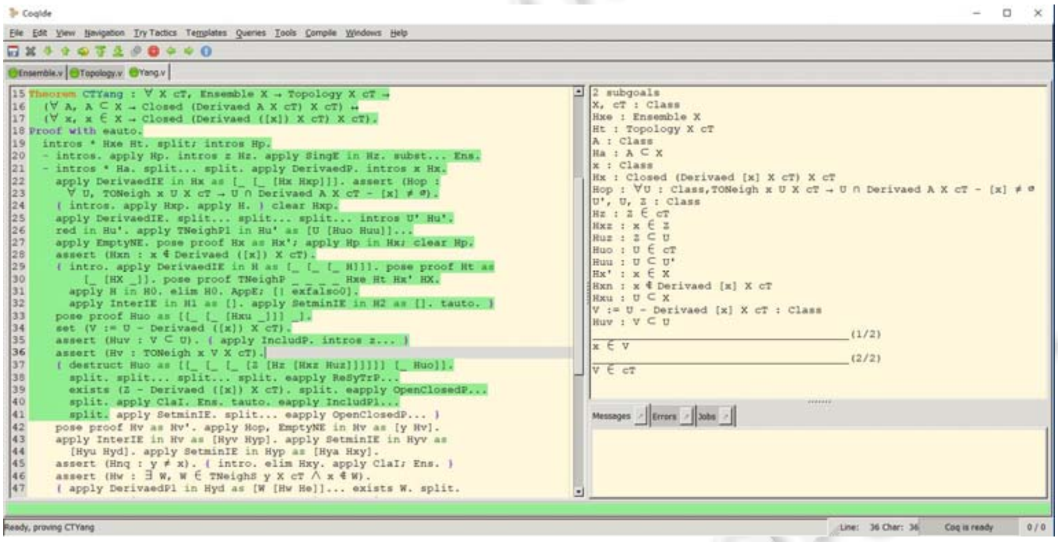

图 1　杨忠道定理的交互式证明窗口

5　结　论

本文使用定理证明器 Coq, 以文献[41]中的理论体系为基础, 在给出集合论的一些基础知识后, 对一般拓扑学中拓扑基本概念进行了 Coq 代码描述, 包括开集、邻域、导集、闭集等定义的形式化描述, 并对它们的基本性质进行了验证; 基于此, 实现了点集拓扑学中杨忠道定理的形式化证明, 这些工作是一般拓扑学理论形式化的一个深刻体现.

本文所形式化的是一个比较具体的数学定理, 比较抽象. 但这个过程需要先实现一般拓扑学中一些基本概念的形式化, 可以在此基础上对平庸空间、离散空间、有限补空间、可数补空间的基本性质进一步进行探索与形式化, 对后续一般拓扑学、代数拓扑学、微分拓扑的形式化也具有一定的启发性.

本文对一般拓扑学中杨忠道定理的形式化证明是在 Coq 8.9.1 版本上实现的, 全部的定义及定理均给出了对应的形式化描述与证明, 并且编译运行通过. 完整的代码源文件可见 https://github.com/BalanceYan/CT.Yang.

在使用Coq进行杨忠道定理形式化证明的过程中, 主要有以下几个注记.

- Coq具有智能性的特点. 可以通过Ltac指令集成, 如化简、重写、引入等证明策略提高定理证明的自动化程度, 还有一些泛证明策略及自动证明策略相对于人工证明在证明过程会更加高效一些. 这个过程要求使用者熟悉所使用的定理证明工具;
- Coq具有可靠性的特点. Coq的代码语言与数学语言十分相似, 可以理解为一种忠实的“翻译”, 它可以发现人工证明过程中的一些漏洞. 这个过程也要求使用者对要验证的理论较为熟悉, 具有一定的抽象思维与逻辑推理能力;
- Coq具有交互性的特点. 在定理证明过程中, Coq代码是以交互式的方式进行开发, 编译界面会实时显示目前的假设条件与目标, 使用者可以边证明、边修改. 这个过程工作量比较大, 耗时长, 但也正是定理机器证明的魅力之所在.

References:

[1] Wu WT. Mechanical Theorem Proving in Geometries: Basic Principles. Berlin: Springer-Verlag, 1994.

[2] Wu WT. Mathematics Mechanization: Mechanical Geometry Theorem-proving, Mechanical Geometry Problem-solving, and Polynomial Equations-solving. In: Hazewinkel M, ed. Mathematics and Its Applications, Vol.489. Beijing and Dordrecht: Science Press and Kluwer Academic Publishers, 2000.

[3] Xia BC, Yang L. Automated Inequality Proving and Discovering. Singapore: World Scientific, 2016.

[4] Bertot Y, Castéran P. Interactive Theorem Proving and Program Development—Coq’ Art: The Calculus of Inductive Constructions. Berlin: Spring-Verlag, 2004.

[5] Pierce BC, de Amorim AA, Casinghino C, Gaboardi M, Greenberg M, Hriţcu C, Sjöberg V, Yorgey B. Software foundations. 2017. https://softwarefoundations.cis.upenn.edu/

[6] The Coq Development Team. The Coq reference manual. Version 8.9.1. 2019. https://coq.inria.fr/distrib/V8.9.1/refman/

[7] Nipow T, Paulson LC, Wenzel M. Isabelle/HOL: A Proof Assistant for Higher-order Logic. Berlin: Springer-Verlag, 2002.

[8] Hales TC. Formal proof. Notices of the American Mathematical Society, 2008, 55(11): 1370–1380.

[9] Harrision J. Formal proof—Theory and practice. Notices of the American Mathematical Society, 2008, 55(11): 1395–1406.

[10] Bancerek G, Bylinski C, Grabowski A, Kornilowicz A, Matuszewski R, Naumowicz A, Pak K, Urban J. Mizar: State-of-the-art and beyond. In: Proc. of the Int’l Conf. on Intelligent Computer Mathematics, Vol.9150. 2015. 261–279.

[11] Gonthier G. Formal proof—The four color theorem. Notices of the American Mathematical Society, 2008, 55(11): 1382–1393.

[12] Beeson M. Mixing computations and proofs. Journal of Formalized Reasoning, 2016, 9(1): 71–99.

[13] Gonthier G, Asperti A, Avigad J, Bertot Y, Cohen C, Garillot F, Roux SL, Mahboubi A, O’Connor R, Biha SO, Pasca I, Rideau L, Solovyev A, Tassi E, Thery L. A machine-checked proof of the odd order theorem. In: Blazy S, Paulin-Mohring C, Picharidie D, eds. Proc. of the Interactive Theorem Proving. LNCS 7998, Rennes: Springer-Verlag, 2013. 163–179.

[14] Hales T, Adams M, Bauer G, Harrison J, Hoang TL, Kaliszyk C, Magron V, Mclaughlin S, Nguyen TT, Nguyen TQ, Nipkow T, Obua S, Pleso J, Rute J, Solovyev A, Ta AHT, Tran TN, Trieu DT, Urban J, Vu K, Zumkeller R. A formal proof of the Kepler conjecture. Forum of Mathematics, Pi, 5, 2017, e2. https://doi.org/10.1017/fmp.2017.1

[15] Guo LQ, Fu YS, Yu WS. A mechanized proof system of the third generation calculus in Coq. Scientia Sinica Mathematica, 2021, 51(1): 115–136 (in Chinese with English abstract).

[16] Wiedijk F. Formal proof—Getting started. Notices of the American Mathematical Society, 2008, 55(11): 1408–1414.

[17] Chen G. Formalized mathematics and proof engineering. Communications of CCF, 2016, 12(9): 40–44 (in Chinese with English abstract).

[18] Formalizing 100 theorems. http://www.cs.ru.nl/~freek/100/

[19] Voevodsky V. Univalent foundations of mathematics. In: Beklemishev LD, de Queiroz R, eds. Proc. of the 18th Int’l Workshop on Logic, Language, Information and Computation. LNAI 6642, Berlin, Heidelberg: Springer-Verlag, 2011. https://doi.org/10.1007/978-3-642-20920-8_4

[20] Gowers WT. Rough structure and classification. In: Alon N, Bourgain J, Connes A, Gromov M, Milman V, eds. Visions in Mathematics (Part of the Modern Birkhäuser Classics). Basel: Birkhäuser Verlag, 2010. [doi: 10.1007/978-3-0346-0422-2_4]

[21] Holden H, Piene R. The Abel Prize 2003−2007: The First Five Years. New York: Springer-Verlag, 2010.

[22] Yu WS, Sun TY, Fu YS. Machine Proof System of Axiomatic Set Theory. Beijing: Science Press, 2020 (in Chinese).

[23] Yu WS, Fu YS, Guo LQ. Machine Proof System of Foundations of Analysis. Beijing: Science Press, 2022 (in Chinese).

[24] Zeng ZB, Wang JL, Yang ZF, Xiao LYH. A mechanical proof of the C.T. Yang's Theorem related to property of derived sets in general topology. Scientia Sinica Mathematica, 2021, 51(1): 257−288 (in Chinese with English abstract).

[25] Kelley JL. General Topology. New York: Springer-Verlag, 1955.

[26] Han G. The historical study of two important theories in topology [Ph.D. Thesis]. Huhehaote: Inner Mongolia Normal University, 2016 (in Chinese with English abstract).

[27] Pu BM, Liu YM. Fuzzy topology. I. Neighborhood structure of a fuzzy point and Moore-Smith convergence. Journal of Mathematical Analysis and Applications, 1980, 76(2): 571−599. https://doi.org/10.1016/0022-247X(80)90048-7

[28] Liu YM, Luo MK. Fuzzy topology. In: Advances in Fuzzy Systems—Applications and Theory, 9. World Scientific, 1998. https://doi.org/10.1142/3281

[29] Wang GJ. Theory of *L*-fuzzy Topological Space. Xi'an: Shaanxi Normal University Press, 1988 (in Chinese).

[30] Huang Y, Luo MK. The *l*-accumulation point and *w*-accumulation point in *L*-fuzzy topological spaces. Journal of Sichuan University (Natural Science Edition), 2004(5): 940−947 (in Chinese with English abstract).

[31] Ji ZF, Wang RY, Shuang L. C.T.Yang's Theorem in *L*-fuzzy topology spaces. Journal of Mathematical Research and Exposition, 2003(1): 182−184 (in Chinese with English abstract).

[32] Zou XF. The Generalization s of C.T.Yang's Theorem in *L*-fuzzifying topological spaces. Fuzzy Systems and Mathematics, 2009, 23(2): 59−63 (in Chinese with English abstract).

[33] Wang JL. Mechanization of general topology and related problems in automatic deduction based on Isabelle [Ph.D. Thesis]. Shanghai: East China Normal University, 2012 (in Chinese with English abstract).

[34] Schepler D. Topology: General topology in Coq. 2011. https://github.com/coq-community/topology

[35] Wang SY. FormalMath: A side project about formalization of mathematics (Topology). 2021. https://github.com/txyyss/FormalMath/tree/master/Topology

[36] Hölzl J, Immler F, Huffman B. Type classes and filters for mathematical analysis in Isabelle/HOL. In: Blazy S, Paulin-Mohring C, Picharidie D, eds. Proc. of the Interactive Theorem Proving. LNCS 7998, Rennes: Springer-Verlag, 2013. 279−294.

[37] Gammie P, Gioiosa G. The Kuratowski closure-complement theorem. Archive of Formal Proofs. 2017. https://www.isa-afp.org/entries/Kuratowski_Closure_Complement.html

[38] Friedrich S. The topology of lazy lists. In: Archive of Formal Proofs. 2004. https://www.isa-afp.org/entries/Topology.html

[39] Enderton HB. Elements of Set Theory. New York: Spring-Verlag, 1977.

[40] Zhang QP. Set-theory: Coq encoding of ZFC and formalization of the textbook elements of set theory. https://github.com/choukh/Set-Theory

[41] Xiong JC. Set-theoretical Topology Lecture. 4th ed., Beijing: Higher Education Press, 2011 (in Chinese).

附中文参考文献:

[15] 郭礼权, 付尧顺, 郁文生. 基于 Coq 的第三代微积分机器证明系统. 中国科学: 数学, 2021, 51(1): 115−136.

[17] 陈钢. 形式化数学和证明工程. 中国计算机学会通讯, 2016, 12(9): 40−44.

[22] 郁文生, 孙天宇, 付尧顺. 公理化集合论机器证明系统. 北京: 科学出版社, 2020.

[23] 郁文生, 付尧顺, 郭礼权. 分析基础机器证明系统. 北京: 科学出版社, 2022.

[24] 曾振柄, 王建林, 杨争峰, 小林英恒. 点集拓扑学之杨忠道定理的一个机械化证明. 中国科学: 数学, 2021, 51(1): 257−288.

[26] 韩刚. 拓扑学中两个重要定理的历史研究 [博士学位论文]. 呼和浩特: 内蒙古师范大学, 2016.

[29] 王国俊. L-fuzzy 拓扑空间论. 西安: 陕西师范大学出版社, 1988.

[30] 黄勇, 罗懋康. *L*-fuzzy 拓扑空间中的 *l*-聚点和 *w*-聚点. 四川大学学报(自然科学版), 2004(5): 940−947.

[31] 吉智方, 王瑞英, 双龙. *L*-fuzzy 拓扑空间中的杨忠道定理. 数学研究与评论, 2003(1): 182−184.

[32] 邹祥福. 杨忠道定理在 L-不分明化拓扑中的推广. 模糊系统与数学, 2009, 23(2): 59–63.
[33] 王建林. 基于 Isabelle 平台的一般拓扑学机械化及自动定理证明研究 [博士学位论文]. 上海: 华东师范大学, 2012.
[41] 熊金城. 点集拓扑讲义. 第 4 版, 北京: 高等教育出版社, 2011.


严升(1993－), 男, 博士生, 主要研究领域为形式化验证, 数学定理机器证明.


付尧顺(1992－), 男, 博士生, 主要研究领域为形式化验证, 数学定理机器证明.


郁文生(1967－), 男, 博士, 教授, 博士生导师, 主要研究领域为形式化方法, 数学定理机器证明, 人工智能与自动推理, 控制理论, 控制工程.