基于Coq的有标集族相关定理的机器证明

刘 佳[1],吕红伟[1],付尧顺[2],郁文生[1,2*]

(1.伊犁师范大学 数学与统计学院,新疆 伊宁 835000;2.北京邮电大学 电子工程学院,北京 100876)

摘 要:基于计算机证明辅助工具Coq,参考"公理化集合论"形式化系统,实现朴素集合论形式化系统,并在此基础上给出有标集族及其交和并的形式化,完成有标集族相关定理的Coq描述及机器证明代码,所有形式化过程已被Coq验证,并在计算机上运行通过,体现了Coq的规范、严谨、可靠.为构建完整的点集拓扑理论打下基础,在此系统下,有望实现拓扑空间相关性质的形式化系统.

关键词: Coq;机器证明;公理化集合论;有标集族

中图分类号:O212.1 **文献标识码:**A **文章编号:**1673-999X(2020)02-0013-10

0 引言

人工智能研究是国家当前重大科技发展战略之一,夯实人工智能基础理论尤为重要,数学定理机器证明是人工智能基础理论的深刻体现[1,2].

Wiedijk认为,数学历史上发生过三次革命,第一次是公元前3世纪,古希腊数学家欧几里得《几何原本》引入数学证明方法;第二次是19世纪柯西等人引入"严格"数学方法,以及后来的数学逻辑和集合论;第三次就是当前正在进行的形式化数学[3].形式化数学是一个有着30年历史的基础科学研究领域,其目标是用计算机对数学理论进行形式化描述,对数学证明进行验证和检查,并且建立一个包含数学定义、定理和证明的形式化数学库.形式化数学对数学的影响有两个方面:第一,检查数学证明的正确性;第二,促进数学基础的研究[3].

近年来,随着计算机科学的迅猛发展,一些证明工具Coq、Isabelle及HOL等[4-7]相继出现,数学定理的计算机形式化证明取得了长足进展[8].Coq是目前国际上交互式定理证明领域的主流工具,它基于归纳构造演算,有着强大的数学模型基础和很好的扩展性.在编程和推理方面,Coq拥有足够强大的表达能力,从构造简单的项,执行简单的证明,直至建立完整的理论,学习复杂的算法等,对学习者的能力有着不同层次的需求[5-6].Coq是一个交互式的编译环境,用户以人机对话的方式一问一答,用户可以边设计、边修改,使证明过程中的错误及时得到补正[5-6].Coq支持自动推导程序,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.

2005年,国际著名计算机专家Gonthier和Werner基于Coq成功给出了"四色定理"的计算机证明[2],进而,Gonthier又经过6年努力,于2012年完成对有限单群分类定理的机器验证(该证明过程约有4 200个定义

收稿日期:2020-04-20

基金项目:新疆维吾尔自治区自然科学基金项目(2018D01C008).

作者简介:刘佳(1994—),女,硕士,研究方向:数学定理机器证明.

*通信作者:郁文生(1967—),男,教授,研究方向:数学机械化、形式化数学、形式化验证.

和15 000条定理,约170 000行代码)[8],使得证明辅助工具Coq在学术界得到广泛认可. Wiedijk在文献[4]中指出,全球各相关研究团队已经或者计划完成包括Gödel第一不完全定理、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. 这些成果促使证明辅助工具Coq在学术界的影响逐渐增强.

2015年以来,我们研究团队分别在北京邮电大学和伊犁师范大学开始了基于Coq的"公理化集合论""近世代数基础"和"一般拓扑学"形式化系统的研发,对布尔巴基学派强调的现代数学三大母结构的机器证明系统实现进行了有意义的探索尝试[9-11].

点集拓扑学又称一般拓扑学,它是一门研究具有拓扑结构的集合及其在拓扑变换下的不变性质,即拓扑空间及其拓扑性质,是在欧式几何、解析几何、射影几何与微分几何之后发展起来的高度抽象的一门几何学. 作为十分重要的基础性的数学分支,点集拓扑学的许多概念、理论与方法在数学的其他分支中有着广泛的应用.

本文参考了"公理化集合论"形式化系统,调用并补充该系统中的一些基本公理、定义,给出有标集族相关定理机器证明. 本文的新意在于利用交互式定理证明工具Coq,给出纯计算机形式化证明,体现了基于Coq的数学定理机器证明所具有的可读性、交互性和智能性等特点,机器证明过程规范、严谨、可靠.

1 预备知识

集合论是数学一个基本的分支学科,基本概念已渗透到数学的多个领域,包含了集合、元素和成员关系等最基本的数学概念. 19世纪70年代,德国数学家康托尔给出了一个比较完整的集合论,即朴素集合论,对无穷集合的序数和基数进行了研究. 20世纪初,罗素指出了朴素集合论的矛盾,提出了罗素悖论,为了克服悖论,只能把集合论公理化,用公理对集合加以限制,形成了公理化集合论[12].

公理化集合论是对朴素集合论的严格处理,它既保留了朴素集合论有价值的成果,又消除了其可能存在的悖论. 公理化集合论把一些符号组成的表达式称为集合,是一种纯粹形式化的理论,摆脱了集合直观语言的束缚. 公理化集合论建立在若干公理组成的公理系统之上,集合论公理系统普遍采用的公理体系是ZFC,它由Zermelo-Fraenkel(ZF)集合论公理化体系加上选择公理(AC)构成,一部分公理规定集合应当具有的几个简明的性质,另一部分公理定义了可称为集合的表达式. 集合论里其他著名的公理化集合理论有von Neumann-Bernays-Gödel(NBG)集合论和Morse-Kelley(MK)集合论[12].

笔者导师所在团队开发的"公理化集合论"形式化系统是基于MK集合论公理化体系搭建的,"公理化集合论"形式化系统已经完成对MK公理化集合论体系的Coq形式化[9]. MK集合论公理化体系最早在1949年由王浩提出[13],1955年在Kelley的《一般拓扑学》中正式发表[14],此后在1965年由Morse完善. 该公理系统构造了序数和基数,定义了非负整数,并把Peano公设当作定理给予了证明,该公理化系统可用来迅速而又自然地给出一个数学基础,摆脱了集合论中较明显的悖论.

本文在MK公理化集合论的基础上做了精简以及改动,具体改动如下:本文没有对集进行定义,默认集就是类. 本文的元素类型都是集,不存在不是集的情况,同时对于MK公理化集合论系统的全域而言,本文没有涉及. MK公理化集合论系统的分类公理图示说明了属于某一大括号的类必须是集,本文已经默认所有的元素都是集,因此分类公理图示中元素必须是集的限制条件将取消.

下面介绍一些"公理化集合论"系统中的基本概念. 首先定义一个在系统中描述元素和集合概念的"类",在Coq形式化中定义为"Class",其类型是"Type". 通过"Notation"在Coq中添加相应数学符号,增强代码可读性. 形式化定义如下:

```
Parameter Class:Type.
```

本系统除了“=”和基本逻辑概念之外,定义两个基本的常项.第一个是“∈”,它读作“属于”.因为在本系统中不区分集合与元素的类型,统一用Class来表示.因此“∈”的形式化定义如下:

Parameter In:Class –> Class –> Prop.

第二个常项是分类“{ … :… }”读作“{ 所有…的类使得… }”,其形式化定义如下:

Parameter Classifier:(Class–>Prop)–>Class.

通过“Notation”命令可以在Coq中添加数学符号,增强代码可读性.

Notation "\{ P \}":=(Classifier P)(at level 0).

接下来,在上述描述定义基础上,引入分类公理图示及外延公理,MK公理化集合论系统的分类公理图示可以避免明显的悖论,上文已经强调过元素是集的情况可以取消,因此其定义及Coq的形式化描述如下:

分类公理图式　对于每一个 β,$\beta \in \{\alpha:P(\alpha)\}$的充分必要条件是$P(\beta)$,这里 $P(\cdot)$ 是适定的公式,这里“适定的公式”是通过原始公式及构造规则所组合而成的公式,构造过程如下[14]:

(a)原始公式要求满足如下形式,其中“α”和“β”代表变元:

$$\alpha = \beta, \alpha \in \beta.$$

(b)公式的构造要求满足如下形式,下面命题中的“α”和“β”代表变元,“A”和“B”代表任一公式:

$$A \to B; A \leftrightarrow B; \sim A$$

$$A \wedge B; A \vee B$$

$$\forall \alpha, A; \exists \alpha, A$$

$$\beta \in \{\alpha:A\}; \{\alpha:A\} \in \beta; \{\alpha:A\} \in \{\beta:B\}$$

从(a)中的原始公式开始,按(b)中给出的规则递归地构造出来的就是分类公理图式中的“适定的公式”.本系统的公式都是适定的.

Axiom Axiom_Classifier:forall(x:Class)(P:Class –> Prop), x ∈ \{ P \} <–>(P x).

本系统中,我们对相等的类进行补充,引入外延公理,外延公理可以被用来代替相等的定义,省略了关于相等逻辑的定义:

外延公理　对于每个A与B,A=B成立之充分条件就是对每一个x当且仅当$x \in A$时,$x \in B$.

Definition Class_Equal(A B:Class) := forall x , x ∈ A <–> x ∈ B.

Axiom Extensionality_Class:forall A B,Class_Equal A B –> A = B.

有时需要讨论以集合作为元素的类,我们把这样的类称为集族.本文对元素、集合、集族的类型不加以区分,统一用“Class”来表示.特别指出,下文所构造的“类”,我们习惯性地称为“集”,但“类”的概念更加广泛,因此“类”所具有的性质,“集”也会满足,在本系统下“集”和“类”是相容的.在定义有标集族前需要给出几个相关的基本概念,其数学定义及Coq形式化定义如下:

定义 1.1(单点集)　$\{x\} = \{z:z = x\}$.

Definition Singleton x:Class:= \{ λ z, z=x \}.

Notation "[x]" :=(Singleton x)(at level 0,right associativity).

为方便集族的交和并运算,引进类的元的交与并运算,以下涉及的Λ是一个类.

定义 1.2(类的元的交)　$\bigcap \Lambda = \{x:\forall A,$如果$A \in \Lambda,$则$x \in \Lambda\}$.

Definition Element_I CA :Class:= \{ λ x, forall A, A ∈ CA –> x ∈ A \}.

定义 1.3(类的元的并)　$\bigcup \Lambda = \{x:\exists A, x \in \Lambda \wedge A \in \Lambda\}$.

Definition Element_U CA:Class:= \{ λ x, exists A, x ∈ A ∧ A ∈ CA \}.

定义 1.4(无序偶) $\{xy\}=\{x\}\cup\{y\}$.

Definition Unordered x y :Class:= [x]∪[y].

Notation "[x | y]":=(Unordered x y)(at level 0).

进而定义有序偶:

定义 1.5(有序偶) $(x,y)=\{\{x\}\{xy\}\}$.

Definition Ordered x y:Class:= [[x] | [x|y]].

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

定义 1.6(笛卡尔积) $X\times Y=\{(x,y)|x\in X,y\in Y\}$.

Definition Cartesian X Y:Class:= \{\ λ x y,x ∈ X ∧ y ∈ Y \}\.

Notation "X × Y":=(Cartesian X Y)(at level 2,right associativity).

本系统中,有些类是由有序偶组成的,Coq中我们需要对这样的类进行描述,描述如下:

Parameter Classifier_P:(Class –> Class –> Prop) –> Class.

Notation "\{\ P \}\":=(Classifier_P P)(at level 0).

Axiom Axiom_Classifier':forall x y P,[x,y] ∈ \{\ P \}\ <–> P x y.

Axiom Property_P:forall z P,z ∈ \{\ P \}\ –>(exists x y,z = [x,y]) ∧ z ∈ \{\ P \}\.

定义 1.7(关系) R是从X到Y的一个关系当且仅当$R\subset X\times Y$.

Definition Relation R X Y:Prop:= R ⊂ X × Y

定义 1.8(R相关) 如果$(x,y)\in R$,则我们称x与y是R相关的,记作xRy.

Definition Rrelation x R y:Prop:= [x,y] ∈ R.

定义 1.9(像集) 设R是从集合X到集合Y的一个关系,对于任意一个集合A,如果$A\subset X$,那么$\{y|\exists x\in A$使得$xRy\}$称为集合A对于关系R而言的像集,记作$R(A)$. 若$A=X$,则$R(A)$是关系R的值域,记作$R(X)$.

Definition Image R A:= \{ λ y,exists x,x ∈ A ∧ [x,y] ∈ R \}.

Notation " R [A] I ":=(Image R A)(at level 5).

Definition Range R:= \{ λ y,exists x,x ∈ A ∧ [x,y] ∈ R \}.

定义 1.10(映射) 设F是从集合X到Y的一个关系. 如果对于每一个$x\in X$,满足

(1)存在$y\in Y$,使得xFy;

(2)如果对于$y_1,y_2\in Y$有xFy_1和xFy_2,则$y_1=y_2$.

则称关系F是从集合X到集合Y的一个映射,记作$F:X\to Y$.

Definition Function F X Y:Prop:= Relation F X Y ∧(forall x,x ∈ X –> exists y,[x,y]∈F)

∧(forall x y z,[x,y] ∈ F ∧ [x,z] ∈ F –> y=z).

定义 1.11(满射) 设X和Y是两个集合,$f:X\to Y$. 如果f的值域为Y,即$f(X)=Y$,那么称f是一个满射.

Definition FullF f X Y:= Function f X Y ∧ f[X]I = Y.

上述预备知识对有标集族的相关定义已足够. 为方便理解,给出定理证明时Coq常用的基本指令.

表 1 Coq常用的基本指令

指令	用法
intros/intro	引入目标中的条件
unfold	将定义展开
rewrite H	将 H 的右边当作左边代入目标

rewrite <- H	将H的左边当作右边代入目标
elim	消去,对条件进行分解
apply H	应用条件H到证明目标
destruct H	拆分条件H中或、与
split	拆分目标中间的与
generalize	引入假设条件
auto	重复执行apply、assumption等简单策略
assert	指定一个新的目标并证明

2　基本定义

本节给出有标集族及有标集族并和交的定义,在相关定义人工描述之后,相应给出其精确的Coq描述代码.

定义 2.1　设Γ是一个集合,Λ是一个集族,每一个满射$\phi:\Gamma\to\Lambda$称为一个以Γ为指标集的集族.如果将$\phi(\gamma)$改记为A_γ,则按照映射的定义我们有

$$\phi=\{(\gamma,A_\gamma)|\gamma\in\Gamma\}\subset\Gamma\times\Lambda.$$

通常把ϕ记作$\{A_\gamma\}_{\gamma\in\Gamma}$.

```
Definition MarkFamilySet φ Γ CA(1:FullF φ Γ CA ):=\{\ λ γ y,γ ∈ Γ ∧ y = φ[γ] \}\.
```

注:我们先了解到的集族是以集合作为元素的类,即无标集族;进而我们接触到有标集族,而有标集族$\{A_\gamma\}_{\gamma\in\Gamma}$中涉及的所有集合$A_\gamma$又构成一个通常意义下的集族,标准记法为$\{A_\gamma|\gamma\in\Gamma\}$,Coq定义如下:

```
Definition UsFamily φ Γ CA(1:FullF φ Γ CA ):=\{ λ x,exists γ,γ ∈ Γ ∧ x = φ[γ] \}.
```

特别地,由于有标集族与通常意义下的集族在记号上有明显的区别,所以在不至于引起概念上的混淆的情形下,我们通常将有标集族简称为集族.将指标集非空的有标集族称为非空集族;通常意义下的集族是非空集族,指的是这个集族是非空的,而不是指这个集族是由非空集构成的.

定义 2.2　设给定了一个集族$\{A_\gamma\}_{\gamma\in\Gamma}$,集合$\{x|\exists\gamma\in\Gamma$使得$x\in A_\gamma\}$称为$\{A_\gamma\}_{\gamma\in\Gamma}$的并,记作$\bigcup_{\gamma\in\Gamma}A_\gamma$;当指标集$\Gamma$非空时,集合$\{x|\forall\gamma\in\Gamma$有$x\in A_\gamma\}$称为$\{A_\gamma\}_{\gamma\in\Gamma}$的交,记作$\bigcap_{\gamma\in\Gamma}A_\gamma$.

```
Definition Un_FamilySet φ Γ CA(1:FullF φ Γ CA ):=\{ λ x,exists γ,γ ∈ Γ ∧ x ∈φ[γ] \}.
Definition In_FamilySet φ Γ CA(1:FullF φ Γ CA ):=\{ λ x,forall γ,γ ∈ Γ -> x ∈ φ[γ] \}.
```

特别地,当指标集Γ是空集时,$\bigcup_{\gamma\in\Gamma}A_\gamma$是一个空集;而$\bigcap_{\gamma\in\Gamma}A_\gamma$是没有意义的.假如我们仍按照原式给予定义,将会导致$\bigcap_{\gamma\in\Gamma}A_\gamma$是一个包含一切事物的集合,而这种情况是不能容许的.

为方便和完整起见,表2列出了文中一些重要概念的Coq定义、数学定义和数学符号.

表2　文中涉及的Coq定义、数学定义和数学符号

Coq定义	数学定义	数学符号
Included A B	A包含B	$A\subset B$
Union A B	A与B的并	$A\cup B$
Intersection A B	A与B的交	$A\cap B$
Setminus A B	A与B的差	$A-B$
Ordered x y	有序偶	(x,y)

Cartesian X Y	笛卡尔积	$X \times Y$
Rrelation x R y	R相关	xRy
Image R A	像集	$R(A)$
Inverse R	R的逆	R^{-1}
Function F X Y	F是X到Y的一个映射	$F:X \to Y$
MarkFamilySet φ Γ CA	有标集族	$\{A_\gamma\}_{\gamma\in\Gamma}$
UsFamily φ Γ CA	通常意义下的集族	$\{A_\gamma \mid \gamma\in\Gamma\}$
Un_FamilySet φ Γ CA	有标集族的并	$\bigcup_{\gamma\in\Gamma} A_\gamma$
In_FamilySet φ Γ CA	有标集族的交	$\bigcap_{\gamma\in\Gamma} A_\gamma$

3 有标集族相关定理的机器证明

本节在有标集族定义的基础上,给出有标集族相关定理的Coq描述代码,并实现所有定理的完整Coq形式化证明.

定理 3.1 设 $\{A_\gamma\}_{\gamma\in\Gamma}$ 和 $\{B_\delta\}_{\delta\in\Delta}$ 是两个非空集族. 如果 $\{A_\gamma|\gamma\in\Gamma\}=\{B_\delta|\delta\in\Delta\}$, 则有 $\bigcup_{\gamma\in\Gamma}A_\gamma=\bigcup_{\delta\in\Delta}B_\delta$, $\bigcap_{\gamma\in\Gamma}A_\gamma=\bigcap_{\delta\in\Delta}B_\delta$;特别地,如果 $\Lambda=\{A_\gamma|\gamma\in\Gamma\}$,则有 $\bigcup_{\gamma\in\Gamma}A_\gamma=\bigcup_{A\in\Lambda}A$, $\bigcap_{\gamma\in\Gamma}A_\gamma=\bigcap_{A\in\Lambda}A$.

Theorem Theorem3_1_1:forall f g Γ Δ CA CB(l:FullF f Γ CA)(l1:FullF g Δ CB),

(UsFamily f Γ CA l = UsFamily g Δ CB l1 ->Un_FamilySet f Γ CA l = Un_FamilySet g Δ

CB l1∧ In_FamilySet f Γ CA l = In_FamilySet g Δ CB l1.

Theorem Theorem3_1_1':forall f Γ CA(l:FullF f Γ CA),

CA = UsFamily f Γ CA l-> Un_FamilySet f Γ CA l=∪CA/\In_FamilySet f Γ CA l=∩CA.

由于上面4个等式是对称的,因此证明方法类似,证明过程可参考文献[15]. 图1仅展示了定理3.1的部分证明代码.

```
Theorem Theorem4_1_1a : forall f g Γ Δ CA CB (l:FullF f Γ CA )(l1:FullF g Δ CB),
  (UsFamily f Γ CA l = UsFamily g Δ CB l1 ->
   Un_FamilySet f Γ CA l = Un_FamilySet g Δ CB l1).
Proof.
  intros. apply Extensionality_Class. split; intros.
  - apply -> Axiom_Classifier in H0. simpl in H0. destruct H0,H0.
    assert (f [x0] ∈ (UsFamily f Γ CA l)).
    { apply Axiom_Classifier. exists x0. split; auto. }
    rewrite H in H2. apply -> Axiom_Classifier in H2. simpl in H2. destruct H2,H2.
    apply Axiom_Classifier. exists x1. split; auto. rewrite <- H3; auto.
  - apply -> Axiom_Classifier in H0. simpl in H0. destruct H0,H0.
    assert (g [x0] ∈ (UsFamily g Δ CB l1)).
    { apply Axiom_Classifier. exists x0. split; auto. }
    rewrite <- H in H2. apply -> Axiom_Classifier in H2. simpl in H2. destruct H2,H2.
    apply Axiom_Classifier. exists x1. split; auto. rewrite <- H3; auto.
Qed.

Theorem Theorem4_1_1b : forall f g Γ Δ CA CB (l:FullF f Γ CA )(l1:FullF g Δ CB),
  (UsFamily f Γ CA l = UsFamily g Δ CB l1 ->
   In_FamilySet f Γ CA l = In_FamilySet g Δ CB l1).
Proof.
  intros. apply Extensionality_Class. split; intros.
  - apply -> Axiom_Classifier in H0. simpl in H0. apply Axiom_Classifier. intros.
    assert (g [y] ∈ (UsFamily g Δ CB l1)).
    { apply Axiom_Classifier. exists y. split; auto. }
    rewrite <- H in H2. apply -> Axiom_Classifier in H2. simpl in H2. destruct H2,H2.
    apply H0 in H2. rewrite H3; auto.
  - apply -> Axiom_Classifier in H0. simpl in H0. apply Axiom_Classifier. intros.
    assert (f [y] ∈ (UsFamily f Γ CA l)).
    { apply Axiom_Classifier. exists y. split; auto. }
    rewrite -> H in H2. apply -> Axiom_Classifier in H2. simpl in H2. destruct H2,H2.
    apply H0 in H2. rewrite H3; auto.
Qed.
```

图 1 定理3.1部分Coq证明代码

定理3.2 设 $\{A_\gamma\}_{\gamma\in\Gamma}$ 是一个非空的有标集族,A 是一个集合,则

(1)对于任何 $\gamma_0\in\Gamma$, $\bigcap_{\gamma\in\Gamma}A_\gamma\subset A_{\gamma_0}\subset\bigcup_{\gamma\in\Gamma}A_\gamma$;

(2)(分配律)$A\cap\left(\bigcup_{\gamma\in\Gamma}A_\gamma\right)=\bigcup_{\gamma\in\Gamma}(A\cap A_\gamma),A\cup\left(\bigcap_{\gamma\in\Gamma}A_\gamma\right)=\bigcap_{\gamma\in\Gamma}(A\cup A_\gamma)$;

(3)(De Morgan律)$A-\left(\bigcup_{\gamma\in\Gamma}A_\gamma\right)=\bigcap_{\gamma\in\Gamma}(A-A_\gamma),A-\left(\bigcap_{\gamma\in\Gamma}A_\gamma\right)=\bigcup_{\gamma\in\Gamma}(A-A_\gamma)$.

Theorem Theorem3_2_2a:forall φ Γ CA(l:FullF φ Γ CA),Γ ≠ Φ

–>(forall γ0,γ0 ∈ Γ –> In_FamilySet φ Γ CA l ⊂ φ[γ0]/\ φ[γ0] ⊂ Un_FamilySet φ Γ CA l).

Theorem Theorem3_2_2b :forall A φ Γ CA l l1,Γ ≠ Φ –>

(A ∩(Un_FamilySet φ Γ CA l))=Un_FamilySet(Functionφ' φ A Γ) Γ(Family_CA' A CA) l1/\

(A ∩(In_FamilySet φ Γ CA l))=In_FamilySet(Functionφ1' φ A Γ) Γ(Family_CA1' A CA) l1/\

Theorem Theorem3_2_2c :forall A φ Γ CA l l1,Γ ≠ Φ –>

(A –(Un_FamilySet φ Γ CA l))=In_FamilySet(Functionφ2' φ A Γ) Γ(Family_CA2' A CA) l1/\(A –(In_FamilySet φ Γ CA l)) = Un_FamilySet(Functionφ2' φ A Γ) Γ(Family_CA2' A CA) l1.

定理3.2的(1)证明过程简单,故不再进行详细叙述. 为证定理3.2的(2)及(3),需引进一些辅助定义和引理,具体过程如下:

由于$\{A_\gamma\}_{\gamma\in\Gamma}$是一个有标集族,根据有标集族的定义,我们有一个Γ → Λ的满射φ,首先根据φ构造出φ',φ1',φ2',即

$\varphi'=\{(\gamma,y)|\gamma\in\Gamma\wedge y=A\cap\varphi(\gamma)\}$;

$\varphi1'=\{(\gamma,y)|\gamma\in\Gamma\wedge y=A\cup\varphi(\gamma)\}$;

$\varphi2'=\{(\gamma,y)|\gamma\in\Gamma\wedge y=A-\varphi(\gamma)\}$.

根据Λ构造Λ',Λ1',Λ2',即

$\Lambda'=\{B|\exists C,C\in\Lambda\wedge B=A\cap C\}$;

$\Lambda1'=\{B|\exists C,C\in\Lambda\wedge B=A\cup C\}$;

$\Lambda2'=\{B|\exists C,C\in\Lambda\wedge B=A-C\}$.

最后我们只需证明φ'是Γ→Λ'的满射、φ1'是Γ→Λ1'的满射、φ2'是Γ→Λ2'的满射即可. Coq完整描述如下:

Definition Functionφ' φ A Γ:= \{\ λ γ y,γ ∈ Γ /\ y = A ∩ φ[γ] \}\.

Definition Family_CA' A CA:= \{λ B,exists C,C ∈ CA /\ B = A ∩ C \}.

Definition Functionφ1' φ A Γ:= \{\ λ γ y,γ ∈ Γ /\ y = A ∪ φ[γ] \}\.

Definition Family_CA1' A CA:= \{λ B,exists C,C ∈ CA /\ B = A ∪ C \}.

Definition Functionφ2' φ A Γ:= \{\ λ γ y,γ ∈ Γ /\ y = A – φ[γ] \}\.

Definition Family_CA2' A CA:= \{λ B,exists C,C ∈ CA /\ B = A – C \}.

Lemma Lemma3_2_2b :forall φ A Γ CA,FullF φ Γ CA –>

FullF(Functionφ' φ A Γ) Γ(Family_CA' A CA).

Lemma Lemma3_2_2b':forall φ A Γ CA,FullF φ Γ CA –>

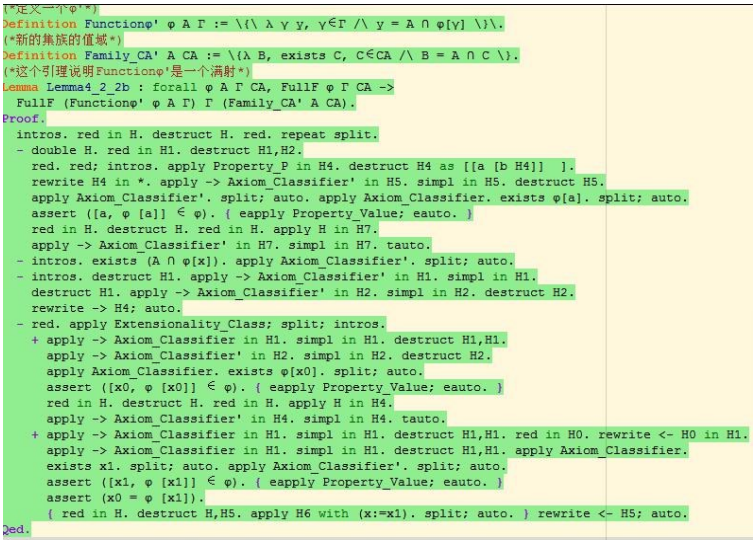
FullF(Functionφ1' φ A Γ) Γ(Family_CA1' A CA).

Lemma Lemma3_2_2c:forall φ A Γ CA,FullF φ Γ CA –>

FullF(Functionφ2' φ A Γ) Γ(Family_CA2' A CA).

由于3个引理的证明过程类似,严格按照满射的定义证明即可,因此图2只展示Lemma3_2_2b的Coq证

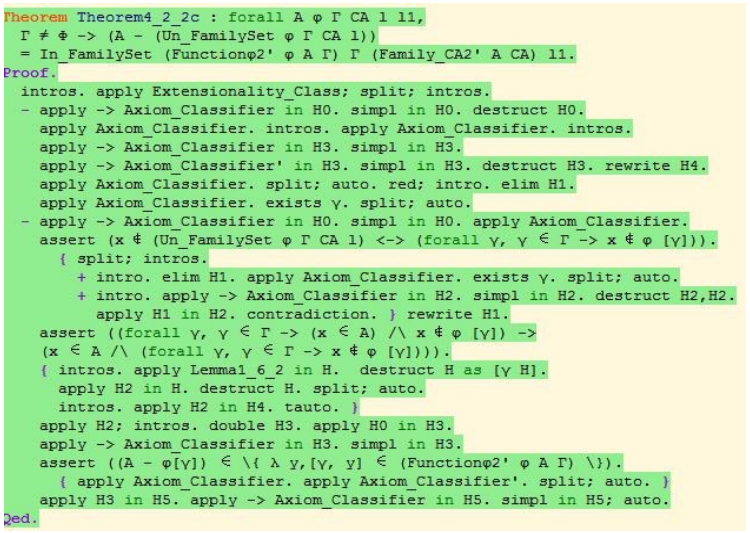
明代码.



```
(*定义一个φ'*)
Definition Functionφ' φ A Γ := \{\ λ γ y, γ∈Γ /\ y = A ∩ φ[γ] \}\.
(*新的集族的值域*)
Definition Family_CA' A CA := \{λ B, exists C, C∈CA /\ B = A ∩ C \}.
(*这个引理说明Functionφ'是一个满射*)
Lemma Lemma4_2_2b : forall φ A Γ CA, FullF φ Γ CA ->
  FullF (Functionφ' φ A Γ) Γ (Family_CA' A CA).
Proof.
  intros. red in H. destruct H. red. repeat split.
  - double H. red in H1. destruct H1,H2.
    red. red; intros. apply Property_P in H4. destruct H4 as [[a [b H4]]  ].
    rewrite H4 in *. apply -> Axiom_Classifier' in H5. simpl in H5. destruct H5.
    apply Axiom_Classifier'. split; auto. apply Axiom_Classifier. exists φ[a]. split; auto.
    assert ([a, φ [a]] ∈ φ). { eapply Property_Value; eauto. }
    red in H. destruct H. red in H. apply H in H7.
    apply -> Axiom_Classifier' in H7. simpl in H7. tauto.
  - intros. exists (A ∩ φ[x]). apply Axiom_Classifier'. split; auto.
  - intros. destruct H1. apply -> Axiom_Classifier' in H1. simpl in H1.
    destruct H1. apply -> Axiom_Classifier' in H2. simpl in H2. destruct H2.
    rewrite -> H4; auto.
  - red. apply Extensionality_Class; split; intros.
    + apply -> Axiom_Classifier in H1. simpl in H1. destruct H1,H1.
      apply -> Axiom_Classifier' in H2. simpl in H2. destruct H2.
      apply Axiom_Classifier. exists φ[x0]. split; auto.
      assert ([x0, φ [x0]] ∈ φ). { eapply Property_Value; eauto. }
      red in H. destruct H. red in H. apply H in H4.
      apply -> Axiom_Classifier' in H4. simpl in H4. tauto.
    + apply -> Axiom_Classifier in H1. simpl in H1. destruct H1,H1. red in H0. rewrite <- H0 in H1.
      apply -> Axiom_Classifier in H1. simpl in H1. destruct H1,H1. apply Axiom_Classifier.
      exists x1. split; auto. apply Axiom_Classifier'. split; auto.
      assert ([x1, φ [x1]] ∈ φ). { eapply Property_Value; eauto. }
      assert (x0 = φ [x1]).
      { red in H. destruct H,H5. apply H6 with (x:=x1). split; auto. } rewrite <- H5; auto.
Qed.
```

图2　Lemma3_2_2b的Coq证明代码

基于上述辅助引理,对定理3.2的(2)及(3)进行证明.由于定理3.2的(2)及(3)中4个等式都是对称的,证明方法类似.图3展示了(3)第一个等式的Coq证明代码.



```
Theorem Theorem4_2_2c : forall A φ Γ CA I I1,
  Γ ≠ Φ -> (A - (Un_FamilySet φ Γ CA I))
  = In_FamilySet (Functionφ2' φ A Γ) Γ (Family_CA2' A CA) I1.
Proof.
  intros. apply Extensionality_Class; split; intros.
  - apply -> Axiom_Classifier in H0. simpl in H0. destruct H0.
    apply Axiom_Classifier. intros. apply Axiom_Classifier. intros.
    apply -> Axiom_Classifier in H3. simpl in H3.
    apply -> Axiom_Classifier' in H3. simpl in H3. destruct H3. rewrite H4.
    apply Axiom_Classifier. split; auto. red; intro. elim H1.
    apply Axiom_Classifier. exists γ. split; auto.
  - apply -> Axiom_Classifier in H0. simpl in H0. apply Axiom_Classifier.
    assert (x ∉ (Un_FamilySet φ Γ CA I) <-> (forall γ, γ ∈ Γ -> x ∉ φ [γ])).
      { split; intros.
        + intro. elim H1. apply Axiom_Classifier. exists γ. split; auto.
        + intro. apply -> Axiom_Classifier in H2. simpl in H2. destruct H2,H2.
          apply H1 in H2. contradiction. } rewrite H1.
    assert ((forall γ, γ ∈ Γ -> (x ∈ A) /\ x ∉ φ [γ]) ->
    (x ∈ A /\ (forall γ, γ ∈ Γ -> x ∉ φ [γ]))).
    { intros. apply Lemma1_6_2 in H.  destruct H as [γ H].
      apply H2 in H. destruct H. split; auto.
      intros. apply H2 in H4. tauto. }
    apply H2; intros. double H3. apply H0 in H3.
    apply -> Axiom_Classifier in H3. simpl in H3.
    assert ((A - φ[γ]) ∈ \{ λ y,[γ, y] ∈ (Functionφ2' φ A Γ) \}).
      { apply Axiom_Classifier. apply Axiom_Classifier'. split; auto. }
    apply H3 in H5. apply -> Axiom_Classifier in H5. simpl in H5; auto.
Qed.
```

图3　定理3.2.2c的Coq证明代码

定理 3.3　设R是集合X到集合Y的一个关系,则对于集合X的任何一个非空子集族$\{A_\gamma\}_{\gamma\in\Gamma}$,有

$$R\left(\bigcup_{\gamma\in\Gamma}A_\gamma\right)=\bigcup_{\gamma\in\Gamma}R\left(A_\gamma\right),R\left(\bigcap_{\gamma\in\Gamma}A_\gamma\right)\subset\bigcap_{\gamma\in\Gamma}R\left(A_\gamma\right).$$

Theorem Theorem3_3:forall R X Y φ Γ CA I I1,R ⊂ X × Y /\ ∪(UsFamily φ Γ CA I) ⊂X

–> R[(Un_FamilySet φ Γ CA I)]I=Un_FamilySet(Functionφ3' R φ Γ) Γ(Family_CA3' R CA) I1

/\ R[(In_FamilySet φ Γ CA I)]I⊂In_FamilySet(Functionφ3' R φ Γ) Γ(Family_CA3' R CA) I1.

若$y\in R\left(\bigcup_{\gamma\in\Gamma}A_\gamma\right)$<=>存在$x\in\bigcup_{\gamma\in\Gamma}A_\gamma$使得$(x,y)\in R$<=>存在$\gamma\in\Gamma$使得$x\in A_\gamma$且$(x,y)\in R$<=>存在$\gamma\in\Gamma$且$y\in R\left(A_\gamma\right)$

($R\left(A_\gamma\right)$是集合A_γ的像集)<=>$y\in\bigcup_{\gamma\in\Gamma}R\left(A_\gamma\right)$.即$R\left(\bigcup_{\gamma\in\Gamma}A_\gamma\right)=\bigcup_{\gamma\in\Gamma}R\left(A_\gamma\right)$.该定理完整证明代码如图4所示.

```
Theorem Theorem4_3a : forall R X Y φ Γ CA l ll,
  R ⊂ X × Y /\ ⋃ (UsFamily φ Γ CA l) ⊂ X
  -> R[(Un_FamilySet φ Γ CA l)]I
  = Un_FamilySet (Functionφ3' R φ Γ) Γ (Family_CA3' R CA) ll.
Proof.
  intros. apply Extensionality_Class; split; intros.
  - apply -> Axiom_Classifier in H0; simpl in H0. destruct H0,H0.
    apply -> Axiom_Classifier in H0; simpl in H0. destruct H0,H0.
    apply Axiom_Classifier. exists x1. split; auto.
    apply Axiom_Classifier. intros. apply -> Axiom_Classifier in H3; simpl in H3.
    apply -> Axiom_Classifier' in H3. simpl in H3.
    destruct H3. rewrite H4. apply Axiom_Classifier. exists x0. split; auto.
  - apply -> Axiom_Classifier in H0; simpl in H0. destruct H0,H0.
    apply -> Axiom_Classifier in H1; simpl in H1.
    assert ((R [φ[x0]]I) ∈ \{ λ y,[x0, y] ∈ (Functionφ3' R φ Γ) \}).
    { apply Axiom_Classifier. apply Axiom_Classifier'. split; auto. }
    apply H1 in H2. apply -> Axiom_Classifier in H2; simpl in H2. destruct H2,H2.
    apply Axiom_Classifier. exists x1. split; auto.
    apply Axiom_Classifier. exists x0. split; auto.
Qed.

Theorem Theorem4_3b : forall R X Y φ Γ CA l ll,
  R ⊂ X × Y /\ ⋃ (UsFamily φ Γ CA l) ⊂ X
  -> R[(In_FamilySet φ Γ CA l)]I
  ⊂ In_FamilySet (Functionφ3' R φ Γ) Γ (Family_CA3' R CA) ll.
Proof.
  intros. red; intros.
  apply -> Axiom_Classifier in H0; simpl in H0. destruct H0,H0.
  apply -> Axiom_Classifier in H0; simpl in H0.
  apply Axiom_Classifier. intros. apply Axiom_Classifier.
  intros. apply -> Axiom_Classifier in H3; simpl in H3.
  apply -> Axiom_Classifier' in H3. simpl in H3. destruct H3. rewrite H4.
  apply Axiom_Classifier. exists x0. apply H0 in H2. split; auto.
Qed.
```

图 4　定理 3.3 的 Coq 证明代码

定理 3.4　设 X 和 Y 是两个集合，$f:X \to Y$，则对于集合 Y 的任何一个非空子集族 $\{B_\gamma\}_{\gamma\in\Gamma}$，有

$$f^{-1}\left(\bigcup_{\gamma\in\Gamma}B_\gamma\right)=\bigcup_{\gamma\in\Gamma}f^{-1}\left(B_\gamma\right),f^{-1}\left(\bigcap_{\gamma\in\Gamma}B_\gamma\right)=\bigcap_{\gamma\in\Gamma}f^{-1}\left(B_\gamma\right).$$

Theorem Theorem4_4：forall f X Y φ Γ CB l ll，Function f X Y/\⋃(UsFamily φ Γ CB l)⊂Y

–> f⁻¹[(Un_FamilySet φ Γ CB l)]I=Un_FamilySet(Functionφ4' f φ Γ) Γ(Family_CA4' f CB) ll

/\ f⁻¹[(In_FamilySet φ Γ CB l)]I=In_FamilySet(Functionφ4' f φ Γ) Γ(Family_CA4' f CB) ll.

根据定理 3.3 立即可得第一个等式成立. 对于第二个等式，我们只需补充证明 $\bigcap_{\gamma\in\Gamma}f^{-1}\left(B_\gamma\right)\subset f^{-1}\left(\bigcap_{\gamma\in\Gamma}B_\gamma\right)$. 证明如下：若 $x\in\bigcap_{\gamma\in\Gamma}f^{-1}\left(B_\gamma\right)$，则对于任何一个 $\gamma\in\Gamma$ 有 $x\in f^{-1}\left(B_\gamma\right)$，即 $f(x)\in B_\gamma$. 于是 $f(x)\in\bigcap_{\gamma\in\Gamma}B_\gamma$，从而 $x\in f^{-1}\left(\bigcap_{\gamma\in\Gamma}B_\gamma\right)$. 即 $\bigcap_{\gamma\in\Gamma}f^{-1}\left(B_\gamma\right)\subset f^{-1}\left(\bigcap_{\gamma\in\Gamma}B_\gamma\right)$. 该定理完整证明代码如图 5 所示.

```
Theorem Theorem4_4a : forall f X Y φ Γ CB l ll,
 Function f X Y /\ ⋃ (UsFamily φ Γ CB l) ⊂ Y
  -> f⁻¹[(Un_FamilySet φ Γ CB l)]I
  = Un_FamilySet (Functionφ4' f φ Γ) Γ (Family_CA4' f CB) ll.
Proof.
  intros.
  eapply Theorem4_3a. destruct H. apply lemma1_5_2b in f0. eauto.
Qed.

Theorem Theorem4_4b : forall f X Y φ Γ CB l ll,
 Function f X Y /\ Γ ≠ Φ /\ ⋃ (UsFamily φ Γ CB l) ⊂ Y
  -> f⁻¹[(In_FamilySet φ Γ CB l)]I
  = In_FamilySet (Functionφ4' f φ Γ) Γ (Family_CA4' f CB) ll.
Proof.
  intros. apply Extensionality_Class; split; auto.
  - eapply Theorem4_3b. destruct H,a. apply lemma1_5_2b in f0. eauto.
  - intros. apply -> Axiom_Classifier in H0; simpl in H0. destruct H,H1.
    apply Lemma1_6_2 in H1. destruct H1. double H1. apply H0 in H1.
    apply -> Axiom_Classifier in H1; simpl in H1. double H3. apply H0 in H3.
    apply -> Axiom_Classifier in H3; simpl in H3.
    assert ((f⁻¹ [φ[x0]]I) ∈ \{ λ y,[x0, y] ∈ (Functionφ4' f φ Γ) \}).
    { apply Axiom_Classifier. apply Axiom_Classifier'. split; auto. }
    apply H3 in H5. apply -> Axiom_Classifier in H5; simpl in H5. destruct H5,H5.
    apply Axiom_Classifier. exists x1. split; auto.
    apply Axiom_Classifier. apply -> Axiom_Classifier' in H6; simpl in H6.
    intros. double H7. apply H0 in H7. apply -> Axiom_Classifier in H7; simpl in H7.
    assert ((f⁻¹ [φ[y]]I) ∈ \{ λ y,[y, y] ∈ (Functionφ4' f φ Γ) \}).
    { apply Axiom_Classifier. apply Axiom_Classifier'. split; auto. }
    apply H7 in H9. apply -> Axiom_Classifier in H9; simpl in H9.
    destruct H9,H9. apply -> Axiom_Classifier' in H10; simpl in H10.
    assert (x1 = x2). { red in H. destruct H,H11. eapply H12; eauto. }
    rewrite H11; auto.
Qed.
```

图 5　定理 3.4 的 Coq 证明代码

4 结论

本文是“公理化集合论”形式化系统的一个应用,利用交互式定理证明工具Coq,在该系统上,建立了点集拓扑理论的机器证明框架系统,完成了有标集族的定义及相关定理的Coq描述,全部定理均给出了人工证明及Coq的机器证明代码.本系统为构建完整的点集拓扑理论打下了基础,在此系统下,有望结合代数学与分析学,探究代数拓扑与微分拓扑形式化方面的工作.

参考文献:

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

[2] GONTHIER G. Formal proof - the Four Color Theorem[J]. Notices of the American Mathematical Society,2008,55(11):1382–1393.

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

[4] WIEDIJK F. Formal proof - getting started[J]. Notices of the American Mathematical Society,2008,55(11):1408–1414.

[5] BERTOT Y, CASTERAN P. Interactive Theorem Proving and Program Development - Coq' Art: The Calculus of Inductive Constructions[M]. Berlin:Spring-Verlag,2004:1–11.

[6] The Coq Development Team. The Coq Proof Assistant Reference Manual(Version 8.9.0)[ED/OL].(2019-04-17)[2019-05-23]. https://coq.inria.fr/distrib/current/refman.

[7] HALES T C. Formal proof[J]. Notices of the American Mathematical Society,2008,55(11):1370–1380.

[8] GONTHIER G, ASPERTI A, et al. Machine-checked proof of the Odd Order Theorem[C]. Proceedings of the Interactive Theorem Proving 2013(Blazy S,Paulin C,Picharidie D,Eds),LNCS,2013,7998:163–179.

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

[10] SUN T Y, YU W S. Machine proving system for mathematical theorems based on Coq - machine proving of Hausdorff maximal principle and Zermelo postulate[C]. Proceedings of the 36th Chinese Control Conference,2017:9871–9878.

[11] ZAO X Y, SUN T Y, et al. Formalization of General Topology in Coq - A Formal Proof of Tychonoff’s Theorem[C]. Proceedings of the 38th Chinese Control Conference,2019:2685–2691.

[12] BEMAYS P,FRAENKEL A A. Axiomatic set theory[M]. Amsterdam:North Holland Publishing Company,1958:1–73.

[13] WANG H. On Zermelo's and Von Neumann's axioms for set theory[J]. Proc. Natl. Acad. Sci. USA,1949,35(3):150–155.

[14] KELLEY J L. General topology[M]. New York:Springer-Verlag,1955:230–241.

[15] 熊金城.点集拓扑讲义[M].4版.北京:高等教育出版社,2011:36–40.

【责任编辑:张建国】

Machine Proving of Correlation Theorem of Standard Set Family Based on Coq

Liu Jia[1], Lv Hongwei[1], Fu Yaoshun[2], Yu Wensheng[1,2*]

(1.College of Mathematics and Statistics,Yili Normal University, Yining, Xinjiang 835000, China; 2.School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Based on the proof assistant Coq, referring to the axiomatic set theory formal system, we realize a naive set theory formal system. Furthermore a set of labeled sets and their intersections are formalized. The Coq description and machine proof code of the related theorem of the standard set family have been verified by Coq, and run on the computer, which reflects Coq's standard, rigorous and reliable. This article lays the foundation for constructing a complete point set topology theory. Under this system, it is expected to realize a formal system of topological space related properties.

Key words: Coq; machine proof; axiomatic set theory; labeled set family