IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 7, 2020, accepted January 20, 2020, date of publication January 27, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969486

A Formal System of Axiomatic Set Theory in Coq

TIANYU SUN, (Member, IEEE), AND WENSHENG YU
Beijing Key Laboratory of Space-Ground Interconnection and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Tianyu Sun (stycyj@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grant 61571064 and Grant 61936008.

ABSTRACT Formal verification technology has been widely applied in the fields of mathematics and computer science. The formalization of fundamental mathematical theories is particularly essential. Axiomatic set theory is a foundational system of mathematics and has important applications in computer science. Most of the basic concepts and theories in computer science are described and demonstrated in terms of set theory. In this paper, we present a formal system of axiomatic set theory based on the Coq proof assistant. The axiomatic system used in the formal system refers to Morse-Kelley set theory which is a relatively complete and concise axiomatic set theory. In this formal system, we complete the formalization of the basic definitions of sets, functions, ordinal numbers, and cardinal numbers and prove the most commonly used theorems in Coq. Moreover, the non-negative integers are defined, and Peano's postulates are proved as theorems. According to the axiom of choice, we also present formal proofs of the Hausdorff maximal principle and Schröeder-Bernstein theorem. The whole formalization of the system includes eight axioms, one axiom schema, 62 definitions, and 148 corollaries or theorems. The ''axiomatic set theory'' formal system is free from the more apparent paradoxes, and a complete axiomatic system is constructed through it. It is designed to give a foundation for mathematics quickly and naturally. On the basis of the system, we can prove many famous mathematical theorems and quickly formalize the theories of topology, modern algebra, data structure, database, artificial intelligence, and so on. It will become an essential theoretical basis for mathematics, computer science, philosophy, and other disciplines.

INDEX TERMS Axiomatic set theory, Coq proof assistant, formalized mathematics, formal system.

I. INTRODUCTION

With the rapid development of computer science, especially the emergence of interactive theorem proving tools Coq [1], [2], Isabelle/HOL [3] and so on, the formal verification technology has made excellent progress [4], [5]. Formal verification technology has achieved remarkable achievements in both formalizations of mathematics and certification of properties of programming languages. In 2005, the international computer experts Gonthier and Werner offered the machine proving of the famous ''four-color theorem'' using Coq successfully [6]. After six years of hard work, Gonthier completed the machine verification of the ''odd order theorem'' in 2012 [7], which made Coq more and more popular in academia. Wiedijk pointed out that relevant research groups around the world have completed or plan to complete the formal proof of 100 famous mathematical theorems including Gödel incompleteness theorem, Prime number theorem, Fermat last theorem, and so on [8]. In the aspect of program verification, Xavier Leroy developed CompCert C in 2008, which is a high-assurance compiler for almost all of the C language, programmed and proved correct in Coq [9]. In 2016, Zhong Shao and Ronghui Gu had successfully developed a practical concurrent OS kernel and verified its functional correctness [10]. Andrew W. Appel, Benjamin C. Pierce, Zhong Shao, and others launched the Deep Specification project which focuses on the specification and verification of full functional correctness of software and hardware in 2016 [11]. These results show that formal verification technology plays an essential role in the fields of mathematics and computer science. However, whether it is the formal proof of complex mathematical theorems or the application of formal technology in engineering, it is necessary to formalize fundamental mathematical theories.

Set theory was founded in the 1870s by the German mathematician G. Cantor [12]. He developed the interest of set theory from the proof that the uniqueness of function expansion as trigonometric series, and gives a relatively complete

The associate editor coordinating the review of this manuscript and approving it for publication was Engang Tian.

theoretical system [13]. Moreover, Cantor also studies the ordinal numbers and cardinal numbers of infinite sets [14]. At the beginning of the twentieth century, Russell's paradox pointed out the contradiction in Cantor's set theory, which caused a great shock in the mathematics world [15]. People who believed that the foundations of mathematics had already established begin to shake their convictions. In order to overcome the paradox, people try to axiomatize the set theory and restrict it with axioms. In 1908, Zermelo proposed the first axiomatic system of set theory. Through the improvement of Fraenkel and Skolem, the famous Zermelo-Fraenkel set theory (ZFC) which includes the axiom of choice (AC) was formed [16]. In 1920, Von Neumann proposed his axiomatic system. It revised by Bernays in 1937 and further simplified by Gödel in 1940 [17]. This is the von Neumann-Bernays-Gödel set theory (NBG).

Morse-Kelley set theory (MK) is also an essential axiomatic set theory. It was first proposed by Wang [18] in 1949 and was formally published in Kelley's *General Topology* [19] in 1955. Morse [20] later presented his version in 1965. MK is a variant of the ZF system which improved by Skolem and Morse, and it is more close to NBG. Meanwhile, it is designed to give quickly and naturally a foundation for mathematics, which is free from the more apparent paradoxes. The ordinal, cardinal numbers,and non-negative integers are constructed. Peano's postulates are proved as theorems. Furthermore, the real numbers can be constructed from integers by use of the axiom of infinity and two facts: "the class of integers is a set" and "it is possible to define a function on the integers by induction" [21]. The axiom system admits classes as fundamental objects, like NBG. In addition, a finite axiom system is abandoned and the development is based on eight axioms and one axiom scheme (that is, all statements of a specific prescribed form are accepted as axioms) [19]. Thus MK cannot be finitely axiomatized and it is strictly stronger than both NBG and ZFC. In fact, NBG and ZFC can be proved consistent in MK. Monk, Rubin, and Mendelson submit that MK does what is expected of a set theory while being less cumbersome than ZFC and NBG [15]–[17].

Axiomatic set theory is the language and foundation of mathematics. It involves almost all branches of mathematics and occupies a significant position in mathematics. Meanwhile, the axiomatic set theory also has critical applications in many fields, such as computer science, artificial intelligence, logic, economics, linguistics, and psychology. Therefore, it is particularly vital to construct a formal system of axiomatic set theory. Most formalizations of axiomatic set theory are based on ZFC. MK is more concise and stronger than ZFC and is more conducive to the initial construction of complex systems. However, there is currently no formal system of axiomatic set theory based on MK. In this paper, we have completed the establishment of the "axiomatic set theory" formal system based on the Coq proof assistant. The formal system is built on the basis of the Morse-Kelley set theory. It includes eight axioms, one axiom schema, and 181 definitions or theorems in the Morse-Kelley set theory which is an appendix to Kelley's "*General Topology*" [19]. Moreover, we add one definition and 28 supplementary corollaries or theorems in the formal system. The complete source of the formal system is available online:
https://github.com/styzystyzy/Axiomatic_Set_Theory/

The formal system can quickly build a foundation for mathematics and present a concise and relatively complete axiomatic set theory in Coq. Compared with naive set theory[1] in Coq standard library, the system has many advantages. First of all, it avoids the more apparent paradoxes of naive set theory. Secondly, there is no type difference between sets and members in this system. The universe of discourse consists of classes. Classes that are members of other classes are called sets. A class that is not a set is a proper class. These help the system avoid the problem of nested type mismatch in Coq formalization. Thirdly, the system is complete for the Coq formalization of set theory. We formalize some basic definitions of sets, functions, ordinal numbers, integers, and cardinal numbers and prove the most commonly used theorems in this formal system. According to the axiom of choice, we also present formal proofs of the Hausdorff maximal principle and Schroeder-Bernstein theorem. Moreover, we present a formal proof of the mathematical induction by the definition of integers and propose some properties related to finiteness. These make the system more complete.

The "axiomatic set theory" formal system has many applications, on the basis of which we can directly study the axioms of choice and continuum hypothesis. Topology can be easily and quickly formalized by it. In fact, Kelley also published his axiomatic set theory in "*General Topology*" [19]. In addition, on the basis of the system, we can formally define the basic concepts in data structure and build a completely reliable basic theory of computer science. The underlying mathematical foundation of artificial intelligence algorithms can be quickly established through this system. In the formal verification of the blockchain smart contract, we can build a safe math library based on this formal system.

The paper is organized as follows. In Section II, we discuss related work. In Section III, we introduce some elementary logic knowledge and some primitive constants of the system, which are preliminary knowledge of the whole system. In Section IV, we present our formalization of the whole definitions in the system, including sets, functions, well ordering, ordinal numbers, integers, cardinal numbers, and so on. Section V introduces the formalization of the classification axiom scheme and eight axioms. The development of the axiom system is based on eight axioms and one axiom scheme. In Section VI, we describe the formalization and the proof of some crucial theorems in the system. In Sction VII, we discuss the application of the formal system through

[1] The naive set theory in Coq standard library are available at https://coq.inria.fr/distrib/current/stdlib/Coq.Sets.Ensembles.html/

```
1 Lemma NNPP : forall p: Prop, ~ ~ p -> p.
2 Lemma imply_to_or : forall P Q: Prop, (P -> Q) -> ~ P \/ Q.
3 Lemma not_and_or : forall P Q: Prop, ~ (P /\ Q) -> ~ P \/ ~ Q.
4 Lemma or_not_and : forall P Q: Prop, ~ P \/ ~ Q -> ~ (P /\ Q).
5 Lemma not_all_ex_not : forall P: U -> Prop, ~ (forall n: U, P n) -> exists n: U, ~ P n.
6 Lemma not_ex_all_not : forall P: U -> Prop, ~ (exists n: U, P n) -> forall n: U, ~ P n.
```

FIGURE 1. Basic logic properties in Logic.Classical library.

```
 1 Proposition Lemma_x : forall x: Prop, x -> x /\ x.
 2 Proof. intros; split; auto. Qed.
 3 Ltac double H := apply Lemma_x in H; destruct H.
 4 
 5 Proposition Lemma_xy : forall (x y: Prop), x -> y -> x /\ y.
 6 Proof. intros; split; auto. Qed.
 7 Ltac add y H:= apply (Lemma_xy _ y) in H; auto.
 8 
 9 Proposition definition_not : forall (A B: Prop), (A <-> B) -> (~ A) -> (~ B).
10 Proof. intros; destruct H; apply imply_to_or in H1; destruct H1; tauto. Qed.
```

FIGURE 2. Extra logic properties and `Ltac` commands.

some examples. In Section VIII, we draw conclusions and discuss some potential further work. Finally, Appendix lists the formal description of essential theorems in the system.

II. RELATED WORK

There already exist several formalizations of axiomatic set theory. For instance, Werner's work [22] is to study relationships between axiomatic set theory and type theory. He has presented two families of relative consistency proofs between ZFC and the calculus of inductive constructions (CIC) in Coq. Based on Werner's work, Barras [23] has formalized the syntactic metatheory of CIC used by the Coq proof assistant, giving it a semantics in set theory and formalizing a soundness proof in Coq itself. Carlos Simpson develops an axiomatization of ZFC and formalizes common set-theoretical notions. See Coq user contribution `coq-contribs/functions-in-zfc`. Jose Grimm wants to implement Bourbaki's "Elements of Mathematics" in Coq based on the work of Carlos Simpson [24]. It is a part of the Gaia project which concerns homological algebra (theory as well as algorithms). In addition, Dominik Kirst and Gert Smolka formalize second-order ZF set theory in the dependent type theory of Coq in [25], [26]. Moreover, Lawrence C. Paulson mechanized the relative consistency of AC and the generalized continuum hypothesis using Isabelle/ZF [27].

Our present work takes from all of the above-cited works. However, the axiomatic system we used is Morse-Kelley axiomatic set theory, which is its first formalization to our knowledge. MK is a proper extension of ZFC and less cumbersome than ZFC and NBG. The formal system we built is concise and complete. We can quickly build a mathematical foundation based on the "axiomatic set theory" formal system. In this paper, we also prove a series of additional theorems on the basis of the formal system.

III. ELEMENTARY LOGIC AND PRIMITIVE CONSTANTS

A. ELEMENTARY LOGIC PROPERTIES

Some basic knowledge of elementary logic is necessary. In this system, we admit the equality '=' and some basic logical constants, including the negation '$\sim$', the conjunction '/\', the disjunction '\/', universal quantification '$\forall$', existential quantification '$\exists$', and so on. Because the formal definition of these constants has been provided in the Coq proof assistant, we can use them directly in the formal system. It should be noted that the symbols in quotes are the symbolic representations of these logical constants in Coq.

In addition to the above basic logical constants, we also recognize some basic logical properties in Fig. 1. The system implements these properties through **`Logic.Classical`** of the Coq standard library. The law of excluded middle (line 1, Fig. 1) is recognized in the library, and some logical properties are proved on the basis of it. Furthermore, as shown in Fig. 2, we also add some other logical properties and construct two '`Ltac`' commands based on these properties. The '`Ltac`' function provides high-level tactics for applying its lemmas and automatically checking their conditions. The first tactic (line 3, Fig. 2) tries to double the specific condition in the proof and the second tactic (line 7, Fig. 2) adds another proven condition to a specific condition. In subsequent formal proofs, we will use these '`Ltac`' functions repeatedly.

B. PRIMITIVE CONSTANTS

There are some primitive constants besides '=' and the other logical constants. Through these concepts, we can construct the set theory. Without these concepts, we can do nothing in mathematics. The first constant we define is the '`Class`'. In the system, the universe of discourse consists of classes. Classes that are members of other classes are called sets. A class that is not a set is a proper class. In the formalization,

```
 1  Definition Ensemble (x: Class) : Prop := exists y: Class, x ∈ y.
 2  Definition Union x y : Class := \{ λ z, z∈x \/ z∈y \}.
 3  Notation "x ∪ y" := (Union x y) (at level 65, right associativity).
 4  Definition Intersection x y : Class := \{ λ z, z∈x /\ z∈y \}.
 5  Notation "x ∩ y" := (Intersection x y) (at level 60, right associativity).
 6  Definition NotIn x y : Prop := ~ x∈y.
 7  Notation "x ∉ y" := (NotIn x y) (at level 10).
 8  Definition Complement x : Class := \{ λ y, y ∉ x \}.
 9  Notation "¬ x" := (Complement x) (at level 5, right associativity).
10  Definition Difference x y : Class := x ∩ (¬ y).
11  Notation "x ~ y" := (Difference x y) (at level 50, left associativity).
12  Definition Subclass x y : Prop := forall z, z∈x -> z∈y.
13  Notation "x ⊂ y" := (Subclass x y) (at level 70).
14  Definition ProperSubclass x y : Prop := x ⊂ y /\ x <> y.
15  Notation "x ⊊ y" := (ProperSubclass x y) (at level 70).
```

FIGURE 3. Elementary algebra operations of classes.

```
1  Parameter Class : Type.
2  Parameter In : Class -> Class -> Prop.
3  Notation "x ∈ y" := (In x y) (at level 10).
```

FIGURE 4. Primitive constants of the formal system.

```
1  Parameter Clf : (Class -> Prop) -> Class.
2  Parameter Clf_P :(Class->Class->Prop) -> Class.
3  Notation "\{ P \}" := (Clf P) (at level 0).
4  Notation "\{\ P \}\":=(Clf_P P) (at level 0).
```

FIGURE 5. The formal statement and symbol of the classifier.

we choose to let ‘Class’ live in ‘Type’, which is the topmost sort in Coq. Formally, the definition is as shown in the line 1 of Fig. 4. Next, we introduce the constant ‘$\in$’, which is read ‘is a member of’ or ‘belongs to’. Since we do not distinguish the type of sets and members, the formal statement of the content ‘$\in$’ is as shown in the line 2. Notations can be introduced to ease the reading and writing of specifications. This also allows us to stay close to the way mathematicians would write. Moreover, we can also define precedence levels and associativity rules of notations in Coq. The symbol of ‘$\in$’ is defined in the line 3.

The Third constant is the classifier ‘$\{\cdots:\cdots\}$’ and is read ‘the class of all $\cdots$ such that $\cdots$’. For example, $\{x : x \in y\}$ is a classifier. The first blank in the classifier constant is to be occupied by a variable which represents the member of the classifier. The second blank is to be occupied by a formula. It should be noted that the formula here can be any property, including incorrect property. The classifier is just a class, and we do not know if there are members in it. As shown in Fig. 5, the formalization of the classifier is divided into two cases. If the member is a single item, then Clf is used. If the member is an ordered pair, then Clf_P is used.

We introduce the correctly usage method of the Clf next. The input of the Clf is a item with type Class → Prop. The first parameter Class represents any variable of the classifier and the second parameter Prop means the expression that the variable satisfies. The λ-abstraction meets our requirements above. It is implemented through function ‘fun’ in Coq. Referring to the symbol definition in the Coq library, we add a notation to the function fun in this system as shown in Fig. 6.

```
1  Notation "'λ' x .. y , t" :=
2    (fun x => .. (fun y => t) ..) (at level 200,
3    x binder, y binder, right associativity,
4    format "'[ ' 'λ' x .. y ']' , t").
```

FIGURE 6. The notation of the function fun.

IV. DEFINITIONS OF THE SYSTEM

In this section, we introduce the definitions in the formal system. On the basis of these definitions, we construct sets, functions, ordinal numbers, integers, and cardinal numbers.

A. BASIC DEFINITIONS OF SETS

Firstly, we present the formal definition of sets. We say that x is a set if and only if for some y, $x \in y$, where x and y are both classes. If a class is not a set, then we call it a proper class. In Coq, sets are defined as shown in the line 1, Fig. 3. Next, we introduce some elementary algebra operations of classes, including union, intersection, complement, difference, inclusion and so on.

The union of two classes x and y is the class of elements which are in x, in y, or in both x and y. The intersection of two classes x and y is the class of all objects that are members of both the classes x and y. We complete the formal definitions of union and intersection based on the Clf in lines 2-5, Fig. 3. Then we define complement and difference in lines 6-11, Fig. 3. The complement of x is the class of elements that do not belong to x. The difference of x and y, denoted $x \sim y$, is the intersection of x and the complement of y. In the system, $x \subset y$ if and only if for each z, if $z \in x$, then $z \in y$. A class x is a subclass of y, or is contained in y, or y contains x if and only if $x \subset y$. It is essential that ‘$\subset$’ not be confused with ‘$\in$’.

```
1  Definition Inequality (x y: Class) : Prop := ~ (x = y).
2  Notation "x ≠ y" := (Inequality x y) (at level 70).
3  Definition ∅ : Class := \{ λ x, x ≠ x \}.
4  Definition μ : Class := \{ λ x, x = x \}.
5  Definition Element_I x : Class := \{ λ z, forall y, y∈x -> z∈y \}.
6  Notation "∩ x" := (Element_I x) (at level 66).
7  Definition Element_U x : Class := \{ λ z, exists y, z∈y /\ y∈x \}.
8  Notation "∪ x" := (Element_U x) (at level 66).
9  Definition PowerClass x : Class := \{ λ y, y ⊂ x \}.
10 Notation "pow( x )" := (PowerClass x) (at level 0, right associativity).
```

FIGURE 7. Basic definitions related to sets.

```
1  Definition Singleton x : Class := \{ λ z, x∈μ -> z=x \}.
2  Notation "[ x ]" := (Singleton x) (at level 0, right associativity).
3  Definition Unordered x y : Class := [x] ∪ [y].
4  Notation "[ x | y ]" := (Unordered x y) (at level 0).
5  Definition Ordered x y : Class := [ [x] | [x|y] ].
6  Notation "[ x , y ]" := (Ordered x y) (at level 0).
7  Definition First z := ∩∩z.
8  Definition Second z := (∩∪z) ∪ (∪∪z) ~ (∪∩z).
9
10 Definition Relation r : Prop := forall z, z∈r -> exists x y, z = [x,y].
11 Definition Inverse r : Class := \{\ λ x y, [y,x] ∈ r \}\.
12 Notation "r ⁻¹" := (Inverse r)(at level 5).
13 Definition Composition r s : Class :=
14   \{\ λ x z, exists y, [x,y] ∈ s /\ [y,z] ∈ r \}\.
15 Definition Composition' r s : Class :=
16   \{ λ u, exists x y z, u = [x,z] /\ [x,y] ∈ s /\ [y,z] ∈ r \}.
17 Notation "r ∘ s" := (Composition r s) (at level 50, no associativity).
```

FIGURE 8. Definitions related to ordered pairs and relation.

For example, $\emptyset \subset \emptyset$ but it is false that $\emptyset \in \emptyset$. The formalization of the subclass and the proper subclass are in lines 12-15, Fig. 3.

As shown in Fig. 7, we present some basic definitions related to sets, including the void class, the universe, the power class, the intersection of the members of a class, and the union of the members of a class. Since there is no class of $x \neq x$, the class $\{x : x \neq x\}$ is the void class, or zero (line 3). $x \neq x$ if and only if it is false that $x = x$. As shown in the line 4, the Class $\mathcal{U}$ is the universe. It has all the possible sets as members. As shown in lines 5-6, the class $\bigcap x$ is the intersection of the members of x. The members of $\bigcap x$ are members of members of x. As shown in lines 7-8, the class $\bigcup x$ is the union of the members of x. Observe that a set z belongs to $\bigcap x$ if and only if z belongs to every member of x. Respectively, a set z belongs to $\bigcup x$ if and only if z belongs to some member of x. Moreover, according to the definition of the subclass, we can define the power class in lines 9-10. As shown in lines 9-10 of Fig.7, the power class of a class x is the class whose members are all of the subclasses of x.

B. ORDERED PAIRS AND FUNCTIONS

This subsection presents the relevant definitions of functions. The construction of the function starts from the singleton. The unordered pair and the ordered pair can be defined according to the singleton. A singleton $\{x\}$ is a class with exactly one element x if x is a set and $\{x\} = \mathcal{U}$ if x is not a set. In formalization, we use the symbol '$[\,x\,]$' to represent a singleton $\{x\}$ in lines 1-2, Fig. 8. When x is a set, the same result is given by the more subjective definition $\{z : z = x\}$. However, it simplifies statements of results greatly if computations are arranged so that $\mathcal{U}$ is the result of applying the computation outside its pertinent domain.

As shown in lines 3-4, Fig. 8, the class $\{xy\}$ which is the union of $\{x\}$ and $\{y\}$ is an unordered pair. We use the symbol '$[\,x\,|\,y\,]$' to represent it in Coq. Then lines 5-6 define the ordered pair according to the unordered pair and the singleton. The class (x, y) is an ordered pair and it is an unordered pair $\{\{x\}\{xy\}\}$. In the ordered pair (a, b), the object a is called the first coordinate, and the object b is the second coordinate of the pair. We can define them by a series of elementary algebraic operations, as shown in lines 7-8.

The line 10, Fig. 8 define the relation. A relation is a class whose members are ordered pairs. For example, r is a relation iff for each member z of r there is x and y such that $z = (x, y)$. Then lines 11-12 define r^{-1} which is the class $\{(x, y) : (y, x) \in r\}$. If r is a relation, r^{-1} is the relation inverse to r. As shown in lines 13-17, the class $r \circ s$ is the composition of r and s. $r \circ s = \{u :$ for some x, some y and some z, $u = (x, z)$, $(x, y) \in s$ and $(y, z) \in r\}$. To avoid excessive notation we

```
 1  Definition Function f : Prop := Relation f /\ (forall x y z, [x,y]∈f /\ [x,z]∈f -> y=z).
 2  Definition Domain f : Class := \{ λ x, exists y, [x,y] ∈ f \}.
 3  Definition Range f : Class := \{ λ y, exists x, [x,y] ∈ f \}.
 4  Definition Value f x : Class := ∩ \{ λ y, [x,y] ∈ f \}.
 5  Notation "dom( f )" := (Domain f) (at level 5).
 6  Notation "ran( f )" := (Range f) (at level 5).
 7  Notation "f [ x ]" := (Value f x)(at level 5).
 8
 9  Definition Function1_1 f : Prop := Function f /\ Function (f⁻¹).
10  Definition Cartesian x y : Class := \{\ λ u v, u∈x /\ v∈y \}\.
11  Notation "x × y" := (Cartesian x y) (at level 2, right associativity).
12  Definition Restriction f x : Class := f ∩ (x × μ).
13  Notation "f | ( x )" := (Restriction f x) (at level 30).
14  Definition Exponent y x : Class := \{ λ f, Function f /\ dom(f) = x /\ ran(f) ⊂ y \}.
15  Notation "Ex y x" := (Exponent y x) (at level 20).
16  Definition On f x : Prop := Function f /\ dom( f ) = x.
17  Definition To f y : Prop := Function f /\ ran(f) ⊂ y.
18  Definition Onto f y : Prop := Function f /\ ran(f) = y.
```

FIGURE 9. Definitions and notations related to function.

```
 1  Definition Rrelation x r y : Prop := [x,y] ∈ r.
 2  Definition Connect r x : Prop :=
 3    forall u v, u∈x /\ v∈x -> (Rrelation u r v) \/ (Rrelation v r u) \/ (u=v).
 4  Definition Transitive r x : Prop :=
 5    forall u v w, (u∈x /\ v∈x /\ w∈x /\ Rrelation u r v /\  Rrelation v r w) -> Rrelation u r w.
 6  Definition Asymmetric r x : Prop :=
 7    forall u v, (u∈x /\ v∈x /\ Rrelation u r v) -> ~ Rrelation v r u.
 8  Definition FirstMember z r x : Prop := z∈x /\ (forall y, y∈x -> ~ Rrelation y r z).
 9  Definition WellOrdered r x : Prop :=
10    Connect r x /\ (forall y, y ⊂ x /\ y ≠ Φ -> exists z, FirstMember z r y).
11  Definition Section y r x : Prop :=
12    y ⊂ x /\ WellOrdered r x /\ (forall u v, (u∈x /\ v∈y /\ Rrelation u r v) -> u∈y).
13  Definition Order_Pr f r s : Prop :=
14    Function f /\ WellOrdered r dom(f) /\ WellOrdered s ran(f) /\
15    (forall u v, u ∈ dom(f) /\ v ∈ dom(f) /\ Rrelation u r v -> Rrelation f[u] s f[v]).
16  Definition Order_PXY f x y r s : Prop :=
17    WellOrdered r x /\ WellOrdered s y /\ Order_Pr f r s /\
18    Section dom(f) r x /\ Section ran(f) s y.
```

FIGURE 10. Definitions related to well-order relation.

agree that $\{(x, z) : \cdots\}$ is to be identical with $\{u :$ for some x, some z, $u = (x, z)$ and $\cdots\}$. Thus $r \circ s = \{(x, z) :$ for some y, $(x, y) \in s$ and $(y, z) \in r\}$.

On the basis of the definition of relations, we can define functions next in Fig. 9. f is a function if and only if f is a relation and for each x, each y, each z, if $(x, y) \in f$ and $(x, z) \in f$, then $y = z$. The domain of the function f is the class $\{x :$ for some y, $(x, y) \in f\}$. The range of the function f is $\{y :$ for some x, $(x, y) \in f\}$. The value of the function f is the intersection of the members of the class $\{y : (x, y) \in f\}$. The formalization and notations of these definitions is as shown in lines 1-7, Fig. 9.

Then we define 1-1 function based on the definitions of functions and inverse relations in the line 9, Fig. 9. f is a 1-1 function if and only if both f and f^{-1} are functions. The line 10 is the definition of the Cartesian product, on the basis of which we can define the restriction of functions in the line 12. The class $\{(u, v) : u \in x$ and $v \in y\}$ is the cartesian product of x and y, we denoted it as $x \times y$. The restriction of a function f to x, denoted $f|x$, is the intersection of f and $x \times \mathcal{U}$. This definition will also be used in a case f is a relation. In this case, $f|x$ is also a relation. As shown in lines 14-15, we define a class y^x which consists of functions whose domain is x and range is a subclass of y. Through the following three definitions, we can construct the concepts of surjection, injection, and bijection in lines 16-18.

C. WELL ORDERING

As shown in Fig. 10, we define a strict well-order relation by some basic properties of order relations. First, we define the order relation between the two classes. If $(x, y) \in r$ (denoted as $x\,r\,y$), then x is r-related to y or x r-precedes y. Its Coq definition is as shown in the line 1, Fig. 10. Next we define three properties of order relations in lines 2-7. r connects x if and only if when u and v belong to x either $u\,r\,v$ or $v\,r\,u$ or $v = u$. r is transitive in x if and only if, when u, v, and w are members of x and $u\,r\,v$ and $u\,r\,w$, then $u\,r\,w$. r is asymmetric in x if and only if, when u and v are members of x and $u\,r\,v$, then it is not true that $v\,r\,u$.

Then we define the first member of the class in the line 8, Fig. 10. z is an r-first member of x if and only if $z \in x$ and if $y \in x$, then it is false that $y\,r\,z$. The well-order relation can be defined based on the first member in lines 9-10. r well-orders x if and only if r connects x and if $y \subset x$ and $y \neq \emptyset$, then there is an r-first member of y. Finally, the section can be defined naturally in lines 11-12. A subset y of x is an r-section of x iff r well-orders x and no member of $x \sim y$ r-precedes a member of y. The following describes two definitions that combine well-order relation and functions. f is r-s order preserving (lines 13-15) if and only if f is a function, r well-orders *domain* f, s well-orders *range* f, and $f(u)\,s\,f(v)$ whenever u and v are members of *domain* f such that $u\,r\,v$. Furthermore, f is r-s order preserving in x and y (lines 16-18) if and only if r well-orders x, s well-orders y, f is r-s order preserving, *domain* f is an r-section of x, and *range* f is an s-section of y.

D. ORDINALS

In this subsection, a special relationship '$\in$' will be discussed and the ordinal numbers are defined in Fig. 11. First, we define the class E in the line 1, which is the ε-relation. Next, as shown in the line 3, we define the ordinal according to the class E. x is an ordinal if and only if E connects x and x is full. As shown in the line 2, x is full if and only if each member of x is a subset of x. R is the class of all ordinals. Therefore, x is an ordinal number if and only if $x \in R$. The specific formal definitions are as shown in lines 4-5.

As shown in lines 7-11, Fig. 11, we define a usual operation '$x + 1$' for the class x and two relationship symbols '$\prec$, $\preceq$'. On the basis of them, we can further discuss some properties of ordinal numbers.

E. INTEGERS AND THE CHOICE FUNCTION

We define non-negative integers and the class of non-negative integers in this subsection. In addition, we also present some definitions involved in the axiom of choice, including the choice function and nest. The definition of non-negative integers is essential to the system. Peano's postulates can be derived as theorems based on it. The real number may be constructed from the integers [21]. Moreover, we can prove mathematical induction. As shown in lines 1-2, Fig. 12, the definition of non-negative integers is based on ordinals. x is a non-negative integer if and only if x is an ordinal and E^{-1} well-order x. As shown in the line 3, ω is the class of non-negative integers. According to the definition of well-orders, there is a E^{-1}-first member of x if x is a non-negative integer. The E^{-1}-first member is called E-last member (lines 5-6).

```
1  Definition E : Class := \{\ λ x y, x ∈ y \}\.
2  Definition full x := forall m, m∈x -> m ⊂x.
3  Definition Ordinal x := Connect E x /\ full x.
4  Definition R : Class := \{ λ x, Ordinal x \}.
5  Definition Ordinal_Number x : Prop := x ∈ R.
6
7  Definition PlusOne x : Prop := x ∪ [x].
8  Definition Less x y : Prop := x ∈ y.
9  Notation "x ≺ y" := (Less x y) (at level 67).
10 Definition LessEqual x y := x ∈ y \/ x = y.
11 Notation "x ≼ y":=(LessEqual x y)(at level 67).
```

FIGURE 11. Definitions related to ordinal numbers.

```
1  Definition NInteger x : Prop :=
2    Ordinal x /\ WellOrdered (E ⁻¹) x.
3  Definition W : Class := \{ λ x, NInteger x \}.
4
5  Definition LastMember x E y : Prop :=
6    FirstMember x (E ⁻¹) y.
```

FIGURE 12. Definitions related to integers.

The following are definitions related to the axiom of choice, including the choice function and the nest. We can describe the axiom of choice by the choice function. As shown in lines 1-2, Fig. 13, c is a choice function if and only if c is a function and $c(x) \in x$ for each member x of *domain* c. Intuitively, a choice function is a simultaneous selection of a member from each set belonging to *domain* c. Through the definition of the nest, the Hausdorff maximal principle can be described. As shown in lines 4-5, n is a nest if and only if, whenever x and y are members of n, then $x \subset y$ or $y \subset x$.

```
1  Definition ChoiceFunction c : Prop :=
2  Function c /\ (forall x, x∈dom(c) -> c[x]∈x).
3
4  Definition Nest n : Prop :=
5    forall x y, x∈n /\ y∈n -> x ⊂ y \/ y ⊂ x.
```

FIGURE 13. Definitions of the choice function and the nest.

F. CARDINAL NUMBERS

In this subsection, cardinal numbers are defined in Fig. 14. First we define that $x \approx y$ if and only if there is a 1_1 function f with *domain* $f = x$ and *range* $f = y$. If $x \approx y$, then x is equivalent to y, or x and y are equipollent. The formalization of the content is as shown in lines 1-2, Fig. 14. Next, we define cardinal numbers in the line 3. x is a cardinal number if and only if x is an ordinal number and, if $y \in R$ and $y \prec x$, then it is false that $x \approx y$. That is, a cardinal number is an ordinal number which is not equivalent to any smaller ordinal. In addition, C is defined as a class of cardinal numbers in the line 4. The class P consists of all pairs (x, y) such that x is a set and y is a cardinal number equivalent to x. For each set x, the cardinal number $P(x)$ is the power of x or the cardinal of x. The Coq statement of the class P is as shown in the line 5.

```
 1  Definition Equivalent x y : Prop := exists f, Function1_1 f /\ dom(f) = x /\ ran(f) = y.
 2  Notation "x ≈ y" := (Equivalent x y) (at level 70).
 3  Definition Cardinal_Number x : Prop := Ordinal_Number x /\ (forall y, y ∈ R -> y ≺ x -> ~ x≈y).
 4  Definition C : Class := \{ λ x, Cardinal_Number x \}.
 5  Definition P : Class := \{\ λ x y, x ≈ y /\ y ∈ C \}\.
 6
 7  Definition Finite (x: Class) : Prop := P[ x ] ∈ W.
 8  Definition Infinite (x: Class) : Prop := ~ Finite x.
 9
10  Definition Max x y : Class := x ∪ y.
11  Definition LessLess : Class :=
12    \{ λ z, exists u v x y, [u,v] ∈ (R × R) /\ [x,y] ∈ (R × R) /\ z = [ [u,v], [x,y] ] /\
13    ((Max u v ≺ Max x y) \/ (Max u v = Max x y /\ u ≺ x) \/ (Max u v = Max x y /\ u=x /\ v ≺ y)) \}.
14  Notation "≪" := (LessLess) (at level 0, no associativity).
```

FIGURE 14. Definitions related to cardinal numbers.

As shown in lines 7-8, Fig. 14, we can divide the cardinal into two classes, the finite cardinals and the infinite cardinals. A class that is not finite is called infinite. x is finite if and only if $P(x) \in \omega$. As shown in lines 10-14, if x and y are ordinals, the larger of them is $x \cup y$. Therefore, $\max[x, y]$ is the union of x and y. Next we define an order which will be assigned to the cartesian product $R \times R$. For some (u, v), (x, y) in $R \times R$, $(u, v) \ll (x, y)$ if and only if $\max[u, v] \prec \max[x, y]$, or $\max[u, v] = \max[x, y]$ and $u \prec x$, or $\max[u, v] = \max[x, y]$ and $u = x$ and $v \prec y$.

V. THE SYSTEM OF AXIOMS

A. THE CLASSIFICATION AXIOM SCHEME

The classification axiom scheme is essential in this system and some apparent paradoxes in set theory can be avoided by it. A precise statement of the classification axiom scheme requires a description of formulae [19]. The formulae of the system is agreed that:

(a) The result of replacing 'α' and 'β' by variables is, for each of the following, a formula.

$$\alpha = \beta \qquad \alpha \in \beta$$

(b) The result of replacing 'α' and 'β' by variables and 'A' and 'B' by formulae is, for each of the following, a formula.

$$\begin{array}{ccc} \text{if } A, \text{ then } B & A \text{ iff } B & \text{it is false that } A \\ A \text{ and } B \quad A \text{ or } B & \text{for every } \alpha, A & \text{for some } \alpha, A \\ \beta \in \{\alpha : A\} & \{\alpha : A\} \in \beta & \{\alpha : A\} \in \{\beta : B\} \end{array}$$

Formulae are constructed recursively, beginning with the primitive formulae of (a) and proceeding via the constructions permitted by (b).

In the following, 'α' and 'β' are replaced by variables. The formula F of '$F(\alpha)$' is constructed by the above method. And '$F(\beta)$' is represented by the formula obtained from '$F(\alpha)$' by replacing each occurrence of the variable which replaced α by the variable which replaced β. The formula F may contain parameters that are either sets or proper classes. More consequentially, the quantified variables in F may range over all classes and not just over all sets. Nevertheless, the members of $\{\alpha : F(\alpha)\}$ are exactly those sets such that F comes out true. The specific description of the classification axiom scheme is as follows:

Classification axiom-scheme For each β, $\beta \in \{\alpha : F(\alpha)\}$ if and only if β is a set and $F(\beta)$.

As shown in Fig. 15, formulae are not recursively defined in our formalization. Preferably, an embedding of MK formulae in Coq's type of propositions `Prop` is used. The `Prop` includes quantification over arbitrary types. The main feature of MK is to consider classes as the range of quantification. This is adequately represented by a `forall x: Class, ...` sentence. However, the embedding also allows us to quantify over more complex collections. For instance `forall (P: Class → Prop)` can be seen as a quantification over "classes of classes". For the two cases of the classifier, the formalization of the axiom scheme is divided into two cases, as shown in lines 1-4. The principles of the two cases are consistent. It is defined on all properties `F: Class → Prop`. It is a second-order logic and is stronger than Kelley's first-order logic. In this system, we only apply the axiom to those properties that satisfy the requirements of Kelley's first-order logic. Accordingly, there is no problem. In the second case, we often encounter situations where the input value is a single variable in the Coq proof. However, the input variable of `Axiom_SchemP` is required to be an ordered pair. Therefore, lines 6-7 add a property to make the single variable into an ordered pair when there is a single variable that belongs to the classifier of ordered pairs.

This axiom scheme is precisely the general intuitive construction of classes except for the requirement 'β is a set'. This requirement is very evidently unnatural. However, a contradiction may be constructed simply without it. This complication, which needs much technical work on the existence of sets, is the price paid to avoid apparent paradoxes.

B. EIGHT AXIOMS

There are eight axioms in the system. The development of the whole axiom system is based on these eight axioms and the classification axiom-scheme. First, we introduce the axiom of extent. The specific details are as follows.

```
Axiom Axiom_Scheme : forall (b: Class) (P: Class -> Prop),
  b ∈ \{ P \} <-> Ensemble b /\ (P b).
Axiom Axiom_SchemeP : forall (a b: Class) (P: Class -> Class -> Prop),
  [a,b] ∈ \{\ P \}\ <-> Ensemble [a,b] /\ (P a b).

Axiom Property_P : forall (z: Class) (P: Class -> Class -> Prop),
  z ∈ \{\ P \}\ -> (exists a b, z = [a,b]) /\ z ∈ \{\ P \}\.
```

FIGURE 15. Formalization of the classification axiom scheme.

```
Axiom Axiom_Extent : forall x y, x = y <-> (forall z, z∈x <-> z∈y).
Axiom Axiom_Subsets : forall x, Ensemble x -> exists y, Ensemble y /\ (forall z, z ⊂x -> z∈y).
Axiom Axiom_Union : forall x y, Ensemble x /\ Ensemble y -> Ensemble (x ∪ y).
Axiom Axiom_Substitution : forall f, Function f -> Ensemble dom(f) -> Ensemble ran(f).
Axiom Axiom_Amalgamation : forall x, Ensemble x -> Ensemble (∪ x).
Axiom Axiom_Regularity : forall x, x ≠ Φ -> exists y, y ∈ x /\ x ∩ y = Φ.
Axiom Axiom_Infinity : exists y, Ensemble y /\ Φ ∈ y /\ (forall x, x ∈ y -> (x ∪ [x]) ∈ y).
Axiom Axiom_Choice : exists c, ChoiceFunction c /\ dom(c) = μ ~ [Φ].
```

FIGURE 16. Formalization of the axiom system.

Axiom of extent For each x and each y it is true that $x = y$ if and only if for each z, $z \in x$ when and only when $z \in y$.

The axiom of extent can be the definition of equality, thus dispensing with one axiom and with all logical presuppositions about equality. However, there would be no unlimited substitution rule for equality and one would have to assume as an axiom: If $x \in z$ and $y = x$, then $y \in z$. Our system did not take the above measure. We retain the axiom of the extent and use it in combination with the default equality logic.

Next, we introduce two axioms about the existence of sets, including the axiom of subsets and the axiom of union.

Axiom of subsets If x is a set there is a set y such that for each z, if $z \subset x$, then $z \in y$.

Axiom of union If x is a set and y is a set so is $x \cup y$.

According to the axiom of subsets, we can prove that 2^x and $\{x\}$ are set if x is a set. Then the axiom of union implies that $\{x, y\}$ is a set if x and y are sets. The two following axioms further delineate the class of all sets.

Axiom of substitution If f is a function and $domain\, f$ is a set, then $range\, f$ is a set.

Axiom of amalgamation If x is a set so is $\bigcup x$.

The axiom of regularity is presented when defining the ordinal numbers. It is possible that there are classes x and y such that x is the only member of y and y is the only member of x. More generally, it is possible that there is a class z whose members exist by taking in each other's laundry, in the sense that every member of z consists of members of z. The following axiom explicitly denies this possibility by requiring that each non-void class z has at least one member whose elements do not belong to z.

Axiom of regularity If $x \neq \emptyset$ there is a member y of x such that $x \cap y = \emptyset$.

Next, we introduce the axiom of infinity. The non-negative integers and the real numbers can be constructed on the basis of the axiom of infinity.

Axiom of infinity For some y, y is a set, $\emptyset \in y$ and $x \cup \{x\} \in y$ whenever $x \in y$.

The axiom of infinity asserts the unconditional existence of two sets, the infinite inductive set y, and the void class $\emptyset$. $\emptyset$ is a set simply because it is a member of y. Up to this point, everything that has been proved to exist is a class, and Kelley's discussion of sets was entirely hypothetical.

The axiom of choice (AC) is an axiom about the existence of mapping in set theory. AC has a significant role in modern mathematics and has very close ties with many profound mathematics conclusions.

Axiom of choice There is a choice function c whose domain is $\mathcal{U} \sim \{\emptyset\}$.

In this system, we present formal proofs of the Hausdorff maximal principle and Schroeder-Bernstein theorem on the basis of AC. According to the definition in Section IV, we can directly complete the formalization of the above axioms. The Coq statement of these axioms is as shown in Fig. 16.

VI. IMPORTANT THEOREMS

In this system, we have completed the formal proof of all the theorems in Kelley's set theory [19]. In addition, we also prove 28 supplementary corollaries or theorems. We choose some essential theorems of the system to introduce in this section. The complete formal description of the theorems of this section will be listed in Appendix.

A. RUSSELL'S PARADOX

According to naive set theory, let M be the set of all sets that are not members of themselves. If M is not a member of itself, then it must belong to itself. If it belongs to itself, then it contradicts its definition. This contradiction is Russell's paradox. In this system, we avoid Russell's paradox by the classification axiom scheme. As shown in Fig. 17, let N be the class $\{x : x \notin x\}$. According to the classification axiom scheme, $N \in N$ if and only if $N \notin N$ and N is a set. It follows that N is not a set. Therefore, there is no Russell's paradox in the "axiomatic set theory" formal system.

```
Lemma Russell_N : ~ Ensemble \{ λ x, x ∉ x \}.
```

FIGURE 17. **Lemma related to Russell's paradox.**

Then we can prove that the universe $\mathcal{U}$ is not a set based on the `Russell_N`. It is the theorem `universe_notset` in Appendix. The proof details are as follows:

Proof: The class N is a subclass of $\mathcal{U}$. If the class $\mathcal{U}$ is a set, we can prove that N is a set according to the theorem `sub_set` [2]. It is contradictory to the `Russell_N`. Therefore, the universe $\mathcal{U}$ is not a set.

B. HAUSDORFF MAXIMAL PRINCIPLE

The Hausdorff maximal principle is one of many statements equivalent to AC. It is an alternate and earlier formulation of Zorn's lemma proved by Felix Hausdorff in 1914. The Hausdorff maximal principle asserts the existence of a maximal nest in any set. We can prove it according to the axiom of choice and the definition of the nest.

Theorem 6.1 (Hausdorff maximal principle): If x is a set there is a nest n such that $n \subset x$ and if m is a nest, $m \subset x$, and $n \subset m$, then $m = n$.

Proof: The proof is by transfinite induction. According to AC, there is a choice function c whose domain is $\mathcal{U} \sim \{\emptyset\}$. For each h, if h is a set, then there is a function g and $g(h) = c(\{m : m\ is\ a\ nest, m \subset x,\ for\ \forall p \in range(h),\ p \subset m\ and\ p \neq m)\})$. $g(h)$ is a nest in x containing properly each previously selected nest. The formalization of the property is as shown in Fig. 18.

```
Definition En_c x h : Class :=
  \{ λ m, Nest m /\ m ⊂ x /\
  (forall p, p ∈ ran(h) -> p ⊂ m /\ p<>m) \}.

Lemma Ex_Enc : forall x c, Ensemble x ->
  ChoiceFunction c -> (exists g, forall h,
  Ensemble h -> g[ h ] = c[ En_c x h ]).
```

FIGURE 18. **Formalization of the property.**

On the basis of the theorem `unique_fun` in Appendix, there is a function f such that $domain f$ is an ordinal and $f(u) = g(f|u)$ for each ordinal number u. From the definition of the function g it follows that, if $u \in domain f$, then $f(u)$ is a nest, $f(u) \subset x$ and $p \subsetneq f(u)$ for every $p \in range\,(f|u)$. Consequently if u and v are members of $domain f$ and $u \prec v$, then $f(u) \subsetneq f(v)$. Finally, according to the above conditions, we prove that $\bigcup(range f)$ is a nest such that $\bigcup(range f) \subset x$ and $m = \bigcup(range f)$ if m is a nest, $m \subset x$ and $\bigcup(range f) \subset m$. Therefore, $\bigcup(range f)$ is a maximal nest in x.

C. SCHRÖder-Bernstein theorem

The Schröder-Bernstein theorem is named after Felix Bernstein and Ernst Schröder. It is also known as Cantor-Bernstein theorem after Georg Cantor first published it without proof. It can be proved directly without AC, but the proof of the system relies on AC. The theorem states as follows,

Theorem 6.2 (Schröder-Bernstein theorem): If x and y are sets, $u \subset x$, $v \subset y$, $x \approx v$, and $y \approx u$, then $x \approx y$.

This theorem is a useful feature in the ordering of cardinal numbers. We can prove it according to theorems `card_eq` and `card_le` in Appendix.

Proof: AC is needed to prove that $range\,(P) = C$ in the theorem `card_fun` (See Appendix). On the basis of the theorem `card_fun`, we can prove some properties of P, such as if x and y are sets, then $x \approx y$ iff $P(x) = P(y)$ and if y is a set and $x \subset y$, then $P(x) \preceq P(y)$. Using the theorem `card_le`, $P(u) \preceq P(x)$ and $P(v) \preceq P(y)$. In addition, we prove that $P(x) = P(v)$ and $P(y) = P(u)$ based on the theorem `card_eq`. Consequently $P(x) \preceq P(y)$ and $P(y) \preceq P(x)$ are established at the same time. Finally, we prove that $x \approx y$ because of $P(x) = P(y)$.

D. CONTINUUM HYPOTHESIS

Next, we discuss a brief statement on one of the classic unsolved problems of set theory. It is the theorem `cont_hypo` in Appendix and has a direct relationship with the continuum hypothesis. The theorem is described as follows:

Theorem 6.3: There is a unique $\prec$-$\prec$ order-preserving function with *domain* R and *range* $C \sim \omega$.

Proof: According to the theorem `well_order_pre` in Appendix, there is a unique $\prec$-$\prec$ order-preserving function f in R and $C \sim \omega$ such that either $domain f = R$ or $range f = C \sim \omega$. Since every E-section of R and of $C \sim \omega$ is a set and neither R nor $C \sim \omega$ is a set, $domain f = R$ and $range f = C \sim \omega$ are established at the same time.

The unique $\prec$-$\prec$ order-preserving function is usually denoted by $\aleph$. Thus the value $\aleph(0)$ (or $\aleph_0$) of the function is ω. The next cardinal $\aleph_1$ is the first uncountable ordinal. Since $\aleph_0 \prec P(2^{\aleph_0})$ it follows that $\aleph_1 \preceq P(2^{\aleph_0})$. The equality of $P(2^{\aleph_0})$ and $\aleph_1$ is an extremely attractive conjecture. It is called the continuum hypothesis. The generalized continuum hypothesis states that if x is an ordinal number, then $P(2^{\aleph_x}) = \aleph_{x+1}$. Gödel and Cohen proved that the continuum hypothesis is independent of ZFC. Therefore, the continuum hypothesis cannot prove its correctness in axiomatic set theory.

[2] The theorem `sub_set` is a direct consequence of the axiom of subsets. The formal description of the theorem `sub_set` is given in Appendix.

E. MATHEMATICAL INDUCTION

In this system, Peano's postulates are derived as theorems[3]. The theorem `math_ind` in Appendix is the principle of mathematical induction, but it has some differences from the mathematical induction that we usually use. Therefore, we supplement and prove the general form of mathematical induction in the system. Mathematical induction is a fundamental method in a mathematical proof. It is essentially used to prove that a property $P(n)$ holds for every non-negative integer n. First, we prove a lemma which is a basic property of the class of non-negative integers.

Lemma 6.1 (Minimum Principle): Assume that S is a non-empty subclass of ω. There must exist a class $a \in S$, $a \preceq c$ for every $c \in S$.

According to the definition of well-order relation and first member, the above lemma is easy to prove. Then we can prove mathematical induction according to Lemma 6.1. The specific description of the theorem is as follows:

Theorem 6.4 (Mathematical Induction): A property $P(n)$ holds for every non-negative integer n, if

(i) the property holds for the class $\emptyset$;

(ii) if the property holds for one non-negative integer k, then it holds for the next non-negative integer $k + 1$.

Proof: Let S be a class of non-negative integers that make the property untenable. The Coq formalization of the class S is defined as shown in Fig. 19.

```
Definition En_S P := \{ λ x, x∈W /\ ~(P x) \}.
```

FIGURE 19. Formalization of the class S.

Assuming that the property does not hold for all non-negative integers, then $S \neq \emptyset$. Thus, according to the Lemma 6.1, there is a minimum member h of S. Since the property holds for the class $\emptyset$, $h \neq \emptyset$. Let $m + 1 = h$, then m is a non-negative integer. We can prove that $m \notin S$ because h is a minimum member of S, that is, the property holds for m. According to the condition (ii), the property holds for h. Hence $h \notin S$ which is a contradiction. The theorem is completely proved.

F. THEOREM 180 OF KELLEY'S SET THEORY

When proving the Theorem 180 of Kelley's set theory (K.Theorem 180) [19], we found that it cannot be proved. Thus we try to prove its negative form. We regard it as a new theorem as follows:

Theorem 6.5: There are two classes x and y which are members of C. If one of them fails to belong to ω, then $P(x \times y) \neq \max\,[P(x), P(y)]$.

Proof: We assume that x does not belong to ω. Let $x = \omega$ and $y = \emptyset$, then we can prove $\omega \in C$ and $\omega \notin \omega$. According to the theorem `zero_not_int` in Appendix and $\omega \subset C$, the class $\emptyset$ belongs to ω and ω is a subclass of C. Thus $\emptyset \in C$ is proved. Next we prove that $P(\omega \times \emptyset) \neq \max\,[P(\omega), P(\emptyset)]$. $P(\omega \times \emptyset) = P(\emptyset)$ because of $\omega \times \emptyset = \emptyset$. In addition, $P(\omega) \cup P(\emptyset) = P(\omega)$ based on the definition of cardinal numbers. Since $P(\emptyset) \neq P(\omega)$ is obvious, the theorem is proved.

Theorem 6.5 is proved, which means that K,Theorem 180 is wrong. Therefore, we modify K.Theorem 180 according to its proof process as follows. The revised theorem can be proved, and the detailed proof process of the theorem can be seen in the source code.

Theorem 6.6: If x and y are members of C and both are non-zero, one of which fails to belong to ω, then $P(x \times y) = \max\,[P(x), P(y)]$.

The discovery of errors in K.Theorem 180 fully demonstrates that the Coq-based formal proof of mathematics theorem is highly reliable and rigorous.

VII. APPLICATIONS OF THE SYSTEM

The "axiomatic set theory" formal system has essential applications in both mathematics and computer science. According to the system, many complex mathematical theorems can be directly proved. In addition, we can quickly build formal systems of fundamental mathematical theories through this system. The basic theory of computer science can also be formalized on the basis of it. In this section, we take the formal proof of the equivalence of AC and the formal verification of smart contracts as examples to show how to apply this system in mathematics and computer science.

A. AXIOM OF CHOICE

AC is an essential axiom in axiomatic set theory. It is the axiom about the existence of mapping. It was first proposed by Zermelo [28] in 1904 and used for proving the well-ordering theorem. AC has a significant role in modern mathematics and has very close ties with many profound mathematics conclusions. Without AC, we do not even know if two sets can compare their element numbers with each other, if the product of a family of the nonempty set is empty, if the liner space must have a group of bases, if any family of compact space must be compact and so on [16].

AC has a lot of equivalent forms. The most famous are Tukey's lemma, the Hausdorff maximal principle, the maximal principle, Zermelo's postulate, Zorn's lemma, the well-ordering theorem. As shown in Fig. 20, we can prove the equivalence between them based on the "axiomatic set theory" formal system. The specific formal proof of the equivalence has been implemented in [29], [30].

B. SMART CONTRACT

Blockchain technology has attracted more and more attention from academia and industry recently. Ethereum, which uses blockchain technology, is a distributed computing

[3] Theorems `int_succ`, `zero_not_int`, `int_succ_eq` and the theorem `math_ind` in Appendix are Peano's axioms for non-negative integers in the "axiomatic set theory" formal system.

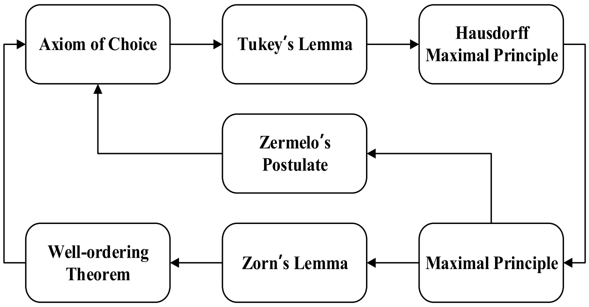


FIGURE 20. The relation of AC and its equivalent theorems.

```
1  Definition safe_plus (a b: uint256) :=
2    if (nat_ble a (a + b)) && (nat_ble b (a + b))
3    then (a + b) else Error.
4
5  Definition safe_minus (a b: uint256) :=
6    if (nat_ble b a) then (a - b) else Error.
7
8  Definition safe_mult (a b: uint256) :=
9    if (nat_beq a 0) || (nat_beq (a * b / a) b)
10   then (a * b) else Error.
11
12 Definition safe_div (a b: uint256) :=
13 if(nat_blt 0 b)&&(nat_beq a (b*(a/b)+ a mod b))
14 then (a / b) else Error.
```

FIGURE 21. Definitions of operators in safae math library.

platform and operating system. Smart contracts are small programs deployed to the Ethereum blockchain for execution. It can be widely used in finance, insurance, the Internet of Things, and other fields. However, errors in smart contracts will lead to huge losses [31]. Formal verification can provide a reliable guarantee for the security of smart contracts [32].

Integer overflow is the most common security issue in smart contracts. This security vulnerability is usually inadvertently introduced by programmers. It may cause some features of the contract to fail. In the most severe case, it may lead to hacker attacks and economic losses. We formally verify the integer overflow issue in smart contracts by building a safe math library. The library can be constructed based on the "axiomatic set theory" formal system.

As shown in Fig. 21, we define safe arithmetic which consists of some basic operators. These operators, which can prevent overflow, include addition `safe_plus`, subtraction `safe_minus`, multiplication `safe_mult`, and division `safe_div`. Through the definition of non-negative integers in the "axiomatic set theory" formal system, we can directly define natural numbers. We can also define other data types related to natural numbers, such as the `uint256` type. Functions `nat_ble`, `nat_blt` and `nat_beq` can determine the magnitude relationship between two natural numbers. Taking the function `safe_plus` as an example, it implements the safe addition of two uint256-type variables.

TABLE 1. Overview of our formal development.

File	Spec	Proof
Elementary_Logic.v	15	10
Classification_Axiom_Scheme.v	50	10
Elementary_Algebra.v	240	260
Sets_Existence.v	120	200
Ordered_Pairs.v	100	260
Functions.v	120	240
Well_Ordering.v	150	750
Ordinals.v	160	760
Integers.v	80	300
Choice_Axiom.v	60	320
Cardinal_Numbers.v	800	3200

By the judging condition in the line 2, it ensures that the sum of two numbers must be greater than or equal to each addend.

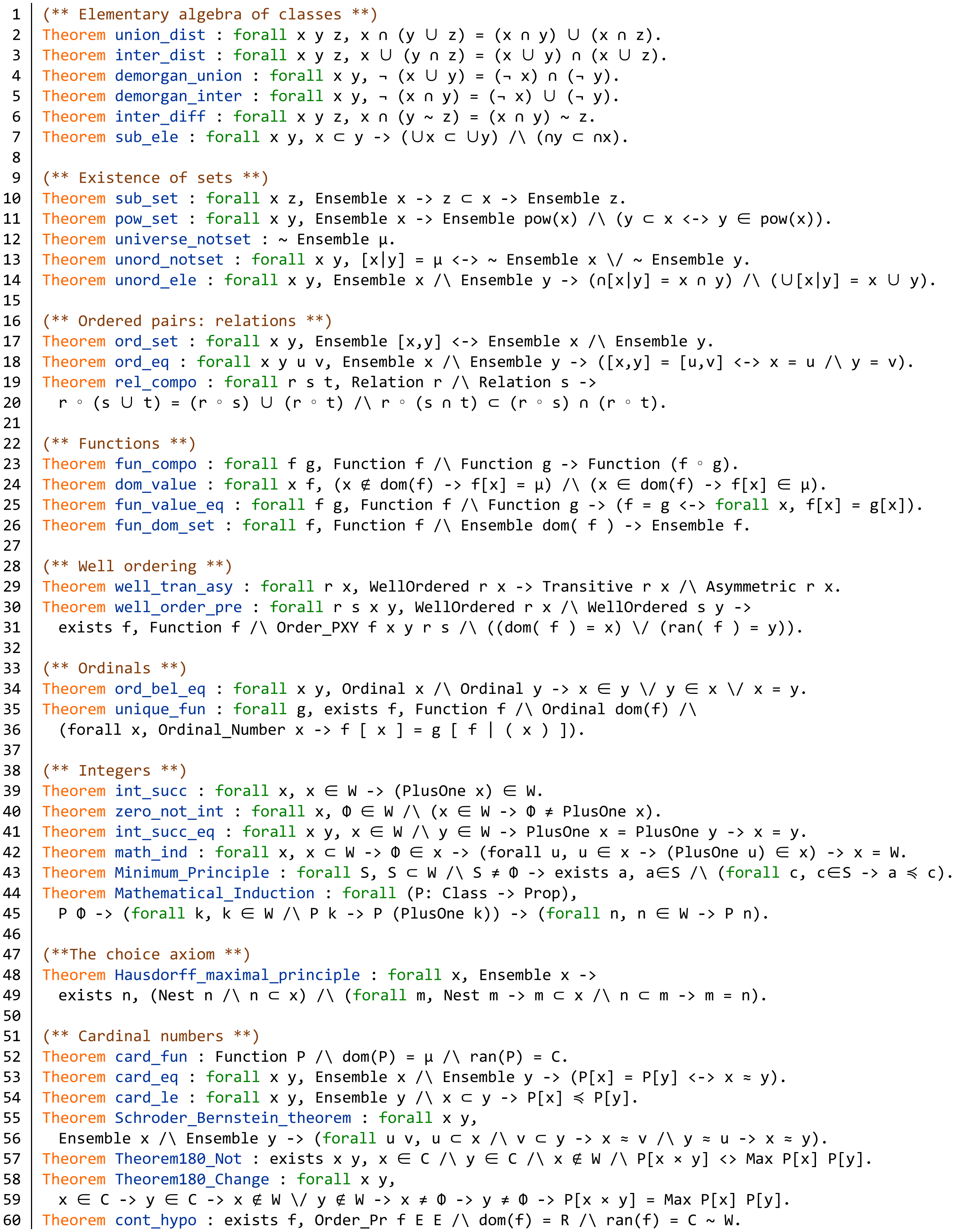
VIII. CONCLUSION AND FUTURE WORK

This paper presents a formal system of axiomatic set theory in Coq. We construct sets, functions, ordinal numbers, integers, and cardinal numbers and prove some commonly used theorems. Furthermore, we prove the Hausdorff maximal principle and Schröeder-Bernstein theorem. Finally, we divide the cardinals into two classes, the finite cardinals and the infinite cardinals. We discuss the related issues of the continuum hypothesis on the basis of the infinite cardinals. The entire system consists of eight axioms, one axiom schema, 62 definitions, and 148 corollaries or theorems. Overall, our current development counts around 9,000 lines of Coq code. It has been tested and should compile under Coq 8.9.0. Table 1 provides a detailed account of the formalization in terms of script files. The count in terms of lines of code distinguishes between specifications and proofs.

In the future, we will prove more famous theorems on the basis of the formal system, such as Tychonoff's theorem, Goodstein's theorem, continuum hypothesis, and so on. Moreover, we plan to complete the formalization of "abstract algebra" and "general topology" on the basis of the "axiomatic set theory" formal system. It will be a meaningful exploration and attempt on the formalization of three modern mathematical structures – ordered structure, algebraic structure, and topological structure – which are proposed by the Bourbaki group. Furthermore, we will formalize the basic concepts and theories of data structure and artificial intelligence based on the formal system.

APPENDIX

As shown in Fig. 22, we list the formal description of important partial theorems in the system. All the theorems involved



```
(** Elementary algebra of classes **)
Theorem union_dist : forall x y z, x ∩ (y ∪ z) = (x ∩ y) ∪ (x ∩ z).
Theorem inter_dist : forall x y z, x ∪ (y ∩ z) = (x ∪ y) ∩ (x ∪ z).
Theorem demorgan_union : forall x y, ¬ (x ∪ y) = (¬ x) ∩ (¬ y).
Theorem demorgan_inter : forall x y, ¬ (x ∩ y) = (¬ x) ∪ (¬ y).
Theorem inter_diff : forall x y z, x ∩ (y ~ z) = (x ∩ y) ~ z.
Theorem sub_ele : forall x y, x ⊂ y -> (∪x ⊂ ∪y) /\ (∩y ⊂ ∩x).

(** Existence of sets **)
Theorem sub_set : forall x z, Ensemble x -> z ⊂ x -> Ensemble z.
Theorem pow_set : forall x y, Ensemble x -> Ensemble pow(x) /\ (y ⊂ x <-> y ∈ pow(x)).
Theorem universe_notset : ~ Ensemble μ.
Theorem unord_notset : forall x y, [x|y] = μ <-> ~ Ensemble x \/ ~ Ensemble y.
Theorem unord_ele : forall x y, Ensemble x /\ Ensemble y -> (∩[x|y] = x ∩ y) /\ (∪[x|y] = x ∪ y).

(** Ordered pairs: relations **)
Theorem ord_set : forall x y, Ensemble [x,y] <-> Ensemble x /\ Ensemble y.
Theorem ord_eq : forall x y u v, Ensemble x /\ Ensemble y -> ([x,y] = [u,v] <-> x = u /\ y = v).
Theorem rel_compo : forall r s t, Relation r /\ Relation s ->
  r ∘ (s ∪ t) = (r ∘ s) ∪ (r ∘ t) /\ r ∘ (s ∩ t) ⊂ (r ∘ s) ∩ (r ∘ t).

(** Functions **)
Theorem fun_compo : forall f g, Function f /\ Function g -> Function (f ∘ g).
Theorem dom_value : forall x f, (x ∉ dom(f) -> f[x] = μ) /\ (x ∈ dom(f) -> f[x] ∈ μ).
Theorem fun_value_eq : forall f g, Function f /\ Function g -> (f = g <-> forall x, f[x] = g[x]).
Theorem fun_dom_set : forall f, Function f /\ Ensemble dom( f ) -> Ensemble f.

(** Well ordering **)
Theorem well_tran_asy : forall r x, WellOrdered r x -> Transitive r x /\ Asymmetric r x.
Theorem well_order_pre : forall r s x y, WellOrdered r x /\ WellOrdered s y ->
  exists f, Function f /\ Order_PXY f x y r s /\ ((dom( f ) = x) \/ (ran( f ) = y)).

(** Ordinals **)
Theorem ord_bel_eq : forall x y, Ordinal x /\ Ordinal y -> x ∈ y \/ y ∈ x \/ x = y.
Theorem unique_fun : forall g, exists f, Function f /\ Ordinal dom(f) /\
  (forall x, Ordinal_Number x -> f [ x ] = g [ f | ( x ) ]).

(** Integers **)
Theorem int_succ : forall x, x ∈ W -> (PlusOne x) ∈ W.
Theorem zero_not_int : forall x, Φ ∈ W /\ (x ∈ W -> Φ ≠ PlusOne x).
Theorem int_succ_eq : forall x y, x ∈ W /\ y ∈ W -> PlusOne x = PlusOne y -> x = y.
Theorem math_ind : forall x, x ⊂ W -> Φ ∈ x -> (forall u, u ∈ x -> (PlusOne u) ∈ x) -> x = W.
Theorem Minimum_Principle : forall S, S ⊂ W /\ S ≠ Φ -> exists a, a∈S /\ (forall c, c∈S -> a ≼ c).
Theorem Mathematical_Induction : forall (P: Class -> Prop),
  P Φ -> (forall k, k ∈ W /\ P k -> P (PlusOne k)) -> (forall n, n ∈ W -> P n).

(**The choice axiom **)
Theorem Hausdorff_maximal_principle : forall x, Ensemble x ->
  exists n, (Nest n /\ n ⊂ x) /\ (forall m, Nest m -> m ⊂ x /\ n ⊂ m -> m = n).

(** Cardinal numbers **)
Theorem card_fun : Function P /\ dom(P) = μ /\ ran(P) = C.
Theorem card_eq : forall x y, Ensemble x /\ Ensemble y -> (P[x] = P[y] <-> x ≈ y).
Theorem card_le : forall x y, Ensemble y /\ x ⊂ y -> P[x] ≼ P[y].
Theorem Schroder_Bernstein_theorem : forall x y,
  Ensemble x /\ Ensemble y -> (forall u v, u ⊂ x /\ v ⊂ y -> x ≈ v /\ y ≈ u -> x ≈ y).
Theorem Theorem180_Not : exists x y, x ∈ C /\ y ∈ C /\ x ∉ W /\ P[x × y] <> Max P[x] P[y].
Theorem Theorem180_Change : forall x y,
  x ∈ C -> y ∈ C -> x ∉ W \/ y ∉ W -> x ≠ Φ -> y ≠ Φ -> P[x × y] = Max P[x] P[y].
Theorem cont_hypo : exists f, Order_Pr f E E /\ dom(f) = R /\ ran(f) = C ~ W.
```

FIGURE 22. The formal description of important partial theorems in the system.

in this paper are listed in the appendix. All corollaries and lemmas will not be listed here. The detailed proof process for all theorems can be found in the source code.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers, whose comments much helped to improve the presentation of their research in this article.

REFERENCES

[1] Y. Bertot and P. Castéran, *Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions*. Heidelberg, Germany: Springer, 2004.

[2] *The Coq Proof Assistant Reference Manual*. Accessed: May 20, 2019. [Online]. Available: https://coq.inria.fr/distrib/current/refman/

[3] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: A Proof Assistant for Higher-Order Logic* Berlin, Germany: Springer-Verlag, 2002.

[4] J. Avigad, "The mechanization of mathematics," *Notices Amer. Math. Soc.*, vol. 65, no. 6, pp. 681–690, Jun. 2018.

[5] T. C. Hales, "Formal proof," *Notices Amer. Math. Soc.*, vol. 55, no. 11, pp. 1370–1380, Dec. 2008.

[6] G. Gonthier, A. Asperti, J. Avigad, Y. Bertot, C. Cohen, F. Garillot, S. Le Roux, A. Mahboubi, R. O'Connor, S. O. Biha, I. Pasca, L. Rideau, A. Solovyev, E. Tassi, and L. Théry, "Formal proof–the four-color theorem," *Notices Amer. Math. Soc.*, vol. 55, no. 11, pp. 1382–1393, Dec. 2008.

[7] G. Gonthier, A. Asperti, J. Avigad, Y. Bertot, C. Cohen, F. Garillot, S. Le Roux, A. Mahboubi, R. O'Connor, S. O. Biha, and I. A. Pasca, "A machine-checked proof of the odd order theorem," in *Proc. Int. Conf. Interact. Theorem Proving (ITP)*, Jun. 2013, pp. 163–179.

[8] F. Wiedijk, "Formal proof—Getting started," *Notices Amer. Math. Soc.*, vol. 55, no. 11, pp. 1408–1414, Dec. 2008.

[9] *The CompCert C verified Compiler: Documentation User's Manual*. Accessed: Sep. 17, 2019. [Online]. Available: http://compcert.inria.fr/man/manual.pdf

[10] R. H. Gu, Z. Shao, and H. Chen, "CertiKOS: An extensible architecture for building cerified concurrent os kernels," in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, Nov. 2016, pp. 653–669.

[11] A. W. Appel, L. Beringer, and A. Chlipala, "Position paper: The science of deep specifcation," *Philos. Trans. Roy. Soc.*, vol. 375, no. 2104, pp. 1–24, Oct. 2017.

[12] A. A. Fraenkel, *Abstract Set Theory (Third Revised Edition)*. Amsterdam, The Netherlands: North Holland Publishing Company, 1966.

[13] A. A. Fraenkel, Y. Bar-Hillel, and A. Levy, *Foundations of Set Theory (Second Revised Edition)*. Amsterdam, The Netherlands: Elsevier, 1973.

[14] P. R. Halmos, *Naive Set Theory*. New York, NY, USA: Springer-Verlag, 1974.

[15] J. V. Heijenoort, *From Frege to Gödel: A Source Book in Mathematical Logic*. Cambridge, MA, USA: Harvard Univ. Press, 1967pp. 1879–1931.

[16] P. Bernays and A. A. Fraenkel, *Axiomatic Set Theory*. Amsterdam, The Netherlands: North Holland Publishing Company, 1958.

[17] T. Jech, *Set Theory The Third Millennium Edition*. Berlin, Germany: Springer-Verlag, 2003.

[18] H. Wang, "On Zermelo's and Von Neumann's axioms for set theory," *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 3, pp. 150–155, 1949.

[19] J. L. Kelley, *General Topology*. New York, NY, USA: Springer-Verlag, 1955.

[20] A. P. Morse, *A Theory of Sets*. New York, NY, USA: Academic, 1965.

[21] E. Landau, *Foundations of Analysis: The Arithmetic of Whole, Rational, Irrational and Complex Numbers*. New York, NY, USA: Chelsea, 1951.

[22] B. Werner, "Sets in types, types in sets," in *Proc. 3rd Int. Symp. Theor. Aspects Comput. Softw. (TACS)*, 1997, pp. 530–546.

[23] B. Barras, "Sets in Coq, Coq in Sets," *J. Formalized Reasoning*, vol. 3, no. 1, pp. 29–48, Jan. 2010.

[24] J. Grimm, *Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two; Ordered Sets, Cardinals, Integers*. Accessed: Dec. 5, 2018. [Online]. Available: https://hal.inria.fr/inria-00440786v10

[25] D. Kirst and G. Smolka, "Categoricity results for second-order ZF in dependent type theory," in *Proc. Int. Conf. Interact. Theorem Proving (ITP)*, Aug. 2017, pp. 304–318.

[26] D. Kirst and G. Smolka, "Large model constructions for second-order ZF in dependent type theory," in *Proc. 7th ACM SIGPLAN Int. Conf. Certified Programs Proofs (CPP)*, Jan. 2018, pp. 228–239.

[27] L. C. Paulson, "The relative consistency of the axiom of choice mechanized using Isabelle/zf," *LMS J. Comput. Math.*, vol. 6, pp. 198–248, Oct. 2003.

[28] E. Zermelo, *Collected works. Volume I, Set Theory, Miscellanea [electronic resource] = Gesammelte Werke. Band I, Mengenlehre, Varia/edited by/Herausgegeben von Heinz-Dieter Ebbinghaus, Akihiro Kanamori*. Berlin, Germany: Springer-Verlag, 2010.

[29] T. Sun and W. Yu, "Machine proving system for mathematical theorems in Coq—Machine proving of Hausdorff maximal principle and Zermelo postulate," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 9871–9878.

[30] T. Y. Sun and W. S. Yu, "Formalization of the axiom of choice and its equivalent theorems," Jun. 2019, *arXiv:1906.03930*. [Online]. Available: https://arxiv.org/abs/1906.03930

[31] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. Int. Conf. Princ. Secur. Trust,* Apr. 2017, pp. 164–186.

[32] J. Liu and Z. Liu, "A survey on security verification of blockchain smart contracts," *IEEE Access*, vol. 7, pp. 77894–77904, 2019.

TIANYU SUN (Member, IEEE) received the bachelor's degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2015, where he is currently pursuing the Ph.D. degree with the School of Electronic Engineering. His research interests include formal verification, formalizing mathematics, and blockchain smart contracts.

WENSHENG YU received the Ph.D. degree from the College of Engineering, Peking University, China, in 1998. After graduation, he conducted research in the fields of system control, formalizing mathematics, and program verification, as a Researcher (a Ph.D. Supervisor) with the Institute of Automation, Chinese Academy of Sciences. He is currently a Professor (a Ph.D. Supervisor) with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. His research interests include formal method, formalizing mathematics, and program verification.