Theoretical Computer Science 554 (2014) 2–21

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A systematic methodology for automated theorem finding

Hongbiao Gao, Yuichi Goto, Jingde Cheng *

Department of Information and Computer Sciences, Saitama University, Saitama 338-8570, Japan

ARTICLE INFO

Article history:
Received 14 September 2013
Accepted 18 June 2014
Available online 22 July 2014

Keywords:
Automated theorem finding
Forward deduction
Strong relevant logic
NBG set theory
Peano's arithmetic

ABSTRACT

The problem of automated theorem finding is one of the 33 basic research problems in automated reasoning which was originally proposed by Wos in 1988, and it is still an open problem. To solve the problem, an approach of forward deduction based on the strong relevant logics was proposed. Following the approach, this paper presents a systematic methodology for automated theorem finding. To show the effectiveness of our methodology, the paper presents two case studies, one is automated theorem finding in NBG set theory and the other is automated theorem finding in Peano's arithmetic. Some known theorems have been found in our case studies.

1. Introduction

The problem of automated theorem finding (ATF for short) is one of the 33 basic research problems in automated reasoning which was originally proposed by Wos in 1988 [14,15], and it is still an open problem until now. The problem of ATF is "What properties can be identified to permit an automated reasoning program to find new and interesting theorems, as opposed to proving conjectured theorems?" [14,15].

The most important and difficult requirement of the problem is that, in contrast to proving conjectured theorems supplied by the user, it asks for the criteria that an automated reasoning program can use to find some theorems in a field that must be evaluated by theorists of the field as new and interesting theorems [3]. The significance of solving the problem is obvious because an automated reasoning program satisfying the requirement can provide great assistance for scientists in various fields [3].

ATF cannot be solved by any automated theorem proving approach, and reasoning is the only way to fit for ATF [3]. Reasoning is the process of drawing new conclusions from some premises which are already known facts and/or previously assumed hypotheses. In contrast, proving is the process of finding a justification for an explicitly specified statement from given premises which are already known facts or previously assumed hypotheses. Discovery is the process to find out or bring to light that which was previously unknown. For any discovery, the discovered thing and its truth must be unknown before the completion of discovery process. Because reasoning is the only way to draw new, previously unknown conclusions from given premises, there is no discovery process that does not invoke reasoning [4].

However, not all logics can serve well as the fundamental logic underlying reasoning for ATF. The classical mathematical logic (CML for short) and its various conservative extensions are not suitable for ATF because they have the well-known "implicational paradoxes" [4]. In order to avoid the implicational paradoxes, relevant logics T, E, and R were constructed [1,2]. However, there are still some paradoxes in theorems of the relevant logics from the viewpoint of relevant reasoning. Cheng named them "conjunction-implicational paradoxes" and "disjunction-implicational paradoxes [4]", and proposed strong relevant logics Tc, Ec, and Rc for relevant reasoning. Tc, Ec, and Rc reject all the conjunction-implicational

* Corresponding author.

http://dx.doi.org/10.1016/j.tcs.2014.06.028
0304-3975/

and disjunction-implicational paradoxes in T, E, and R, respectively, and therefore, by using strong relevant logics as the fundamental logic to underlie reasoning for ATF, one can avoid those problems in using CML, various conservative extensions of CML, and relevant logics T, E, and R. Cheng also proposed predicate strong relevant logics, named TcQ, EcQ, and RcQ [4].

To solve the ATF problem, a forward deduction approach based on the strong relevant logics was proposed by Cheng [3]. Forward deduction is a process which is applying inference rules to the given premises and the deduced conclusions repeatedly to deduce new conclusions until a terminal condition satisfied. Cheng's approach uses the degrees of logic connectives as the terminal condition, uses axioms, definitions, and known facts in one field as premises, and uses inference rules and logic fragments of strong relevant logics to do forward deduction.

To confirm the effectiveness of the approach, we presented a case study of ATF in von Neumann–Bernays–Godel (NBG) set theory by automated forward deduction based on the strong relevant logics [10,11]. In the case study, by using Cheng's approach, we rediscovered some known theorems of NBG set theory. However, in the experiment, our ATF method is ad hoc, but not systematic and general so that our method cannot be used in other case study or other fields [10,11]. To solve the ATF problem, it is necessary to establish a systematic and general methodology for ATF based on Cheng's approach.

This paper presents a systematic methodology for ATF, by which we can find theorems systematically in different mathematical fields. To show the effectiveness of our methodology, the paper presents two case studies, one is automated theorem finding in NBG set theory and the other is automated theorem finding in Peano's arithmetic. Some known theorems have been found in the two case studies, that means we may also find new and interesting theorems in the two fields even in other different fields by using the methodology. The paper also shows some future research directions for ATF.

The rest of the paper is organized as follows: Section 2 explains the basic notions and notations used in the paper. Section 3 presents our systematic methodology for ATF. Section 4 shows how to prepare logic fragments for ATF in various fields. Section 5 presents the case study of ATF in NBG set theory. Section 6 presents the case study of ATF in Peano's arithmetic. Section 7 gives a discussion about our methodology and shows some future research directions for ATF. Finally, concluding remarks are given in Section 8.

2. Basic notions and notations

A formal logic system L is an ordered pair $(F(L), \vdash_L)$ where $F(L)$ is the set of well formed formulas of L, and $\vdash_L$ is the consequence relation of L such that for a set P of formulas and a formula C, $P \vdash_L C$ means that within the framework of L taking P as premises we can obtain C as a valid conclusion. $Th(L)$ is the set of logical theorems of L such that $\phi \vdash_L T$ holds for any $T \in Th(L)$. According to the representation of the consequence relation of a logic, the logic can be represented as a Hilbert style system, a Gentzen sequent calculus system, a Gentzen natural deduction system, and so on [4].

Let $(F(L), \vdash_L)$ be a formal logic system and $P \subseteq F(L)$ be a non-empty set of sentences. A formal theory with premises P based on L, called an L-theory with premises P and denoted by $T_L(P)$, is defined as $T_L(P) =_{df} Th(L) \cup Th^e_L(P)$ where $Th^e_L(P) =_{df} \{A|P \vdash_L A \text{ and } A \notin Th(L)\}$, $Th(L)$ and $T^e_L(P)$ are called the logical part and the empirical part of the formal theory, respectively, and any element of $Th^e_L(P)$ is called an empirical theorem of the formal theory [4].

Based on the definition above, the problem of ATF can be defined as "for any given premises P, how to construct a meaningful formal theory $T_L(P)$ and then find new and interesting theorems in $Th^e_L(P)$ automatically?" [3].

The notion of the degree [6] of a connective is defined as follows: Let θ be an arbitrary n-ary $(1 \le n)$ connective of logic L and A be a formula of L, the degree of θ in A, denoted by $D_\theta(A)$, is defined as follows: (1) $D_\theta(A) = 0$ if and only if there is no occurrence of θ in A, (2) if A is in the form $\theta(a_1, a_2, \dots, a_n)$ where $a_1, a_2, \dots, a_n$ are formulas, then $D_\theta(A) = \max\{D_\theta(a_1), D_\theta(a_2), \dots, D_\theta(a_n)\} + 1$, (3) if A is in the form $\sigma(a_1, a_2, \dots, a_n)$ where σ is a connective different from θ and $a_1, a_2, \dots, a_n$ are formulas, then $D_\theta(A) = \max\{D_\theta(a_1), D_\theta(a_2), \dots, D_\theta(a_n)\}$, and (4) if A is in the form $Q\,B$ where B is a formula and Q is the quantifier prefix of B, then $D_\theta(A) = D_\theta(B)$.

The notion of a fragment of a logic [6] is defined as follows: Let $\theta_1, \theta_2, \dots, \theta_n$ be connectives of logic L and $k_1, k_2, \dots, k_n$ be natural numbers, the fragment of L about $\theta_1, \theta_2, \dots, \theta_n$ and their degrees $k_1, k_2, \dots, k_n$, denoted by $Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$, is a set of logical theorems of L which is inductively defined as follows (in the terms of Hilbert style axiomatic system): (1) if A is an axiom of L and $D_{\theta_1}(A) \le k_1$, $D_{\theta_2}(A) \le k_2$, ..., $D_{\theta_n}(A) \le k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$, (2) if A is the result of applying an inference rule of L to some members of $Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$ and $D_{\theta_1}(A) \le k_1$, $D_{\theta_2}(A) \le k_2$, ..., $D_{\theta_n}(A) \le k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$, (3) Nothing else is in $Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$. Similarly, the notion of degree of formal theory about conditional can also be generally extended to other logic connectives, and a fragment of a formal theory with premises P based on the logic fragment $Th^{(\theta_1,k_1,\theta_2,k_2,\dots,\theta_n,k_n)}(L)$ denoted by $T^{(\eta_1,j_1\dots\eta_s,j_s)}_{Th^{(\theta_1,k_1,\dots,\theta_n,k_n)}(L)}(P)$ is also similarly defined as the notion of a fragment of a logic.

A finite semi-lattice of formal theories [5,7] can be modeled such that its different elements represent different fragments of various formal theories respectively, its partial order among elements represents the inclusion relation between two fragments of formal theories, and its minimum element represents, say, a fragment of the axiomatic set theory.

The notion of predicate abstract level is defined as follows: (1) Let $pal(X) = k$ denote that an abstract level of a predicate X is k where k is a natural number, (2) $pal(X) = 1$ if X is the most primitive predicate in the target field, (3) $pal(X) = \max(pal(Y_1), pal(Y_2), \dots, pal(Y_n)) + 1$ if a predicate X is defined by other predicates $Y_1, Y_2, \dots, Y_n$ in the target field where

n is a natural number. A predicate X is called k-level predicate, if $pal(X) = k$. If $pal(X) < pal(Y)$, then the abstract level of predicate X is lower than Y, and Y is higher than X.

The notion of function abstract level is defined as follows: (1) Let $fal(f) = k$ denote that an abstract level of a function f is k where k is a natural number, (2) $fal(f) = 1$ if f is the most primitive function in the target field, (3) $fal(f) = \max(fal(g_1), fal(g_2), \ldots, fal(g_n)) + 1$ if a function f is defined by other functions $g_1, g_2, \ldots, g_n$ in the target field where n is a natural number. A function f is k-level function, if $fal(f) = k$. If $fal(f) < fal(g)$, then the abstract level of function f is lower than g, and g is higher than f.

The notion of abstract level of a formula is defined as follows: (1) $lfal(A) = (k, m)$ denote that an abstract level of a formula A where $k = pal(A)$ and $m = fal(A)$, (2) $pal(A) = \max(pal(Q_1), pal(Q_2), \ldots, pal(Q_n))$ where Q_i is a predicate and occurs in A $(1 \le i \le n)$, or $pal(A) = 0$ if there is no predicate in A, (3) $fal(A) = \max(fal(g_1), fal(g_2), \ldots, fal(g_n))$ where g_i is a function and occurs in A $(1 \le i \le n)$, or $fal(A) = 0$ if there is no function in A. A formula A is (k, m)-level formula, if $lfal(A) = (k, m)$.

(k, m)-fragment of premises P, denoted by $P_{(k,m)}$, is a set of all formulas in P that consists of only (j, n)-level formulas where m, n, j and k are natural numbers $(0 \le j \le k$ and $0 \le n \le m)$.

FreeEnCal [6] is a forward reasoning engine for general purpose, that provides an easy way to customize reasoning task by providing different axioms, inference rules and facts. Users can set the degree of logical connectives to make FreeEnCal to reason out in principle all logical theorem schemata of the fragment $Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$. FreeEnCal can also reason out in principle all empirical theorems of $T^{(\eta_1,j_1\ldots\eta_s,j_s)}_{Th^{(\theta_1,k_1,\ldots,\theta_n,k_n)}(L)}(P)$ from $Th^{(\theta_1,k_1,\ldots,\theta_n,k_n)}(L)$ and P with inference rules of L.

3. A systematic methodology for automated theorem finding

We proposed a systematic methodology following Cheng's approach. The methodology consists of five phases. The first phase is to prepare logical fragments for various empirical theories. The second phase is to prepare empirical premises of the target theory. The third phase is to reason out empirical theorems in the target theory. The fourth phase is to abstract empirical theorems. The fifth phase is to find interesting theorems from the empirical theorems. Fig. 1 shows the procedure flow of the systematic methodology.

Phase 1 Prepare logical fragments for various empirical theories

The first phase is to prepare various logic fragments of strong relevant logics. Logic fragments of strong relevant logics are independent from any target field. Thus, the prepared logic fragments can be reused for ATF in different fields.

Phase 1.1 Define a semi-lattice of logic fragments

In this phase, we define a semi-lattice [8] of logic fragments of EcQ such that we can prepare logic fragments systematically [5,7]. EcQ is a suitable logic system for reasoning in ATF, because EcQ is the logic system of strict and relevant implication [4]. To define the semi-lattice of logic fragments of EcQ, we use the elements of the semi-lattice to represent the different logic fragments respectively, and use the partial order of the semi-lattice to represent the inclusion relation between two logic fragments [5]. First, we define the minimum element of the semi-lattice. All of the logic fragments of EcQ conclude the axioms of strong relevant logic system Ee, so we can define $Th^{(\Rightarrow,1)}(Ee)$ as the minimum element. Then, we define the semi-lattice of logic fragments according to the inclusion relation of the logic systems in EcQ, i.e., $Th(Ee) \subset Th(Een) \subset Th(Ec)$, $Th(EeQ) \subset Th(EenQ) \subset Th(EcQ)$, $Th(Ee) \subset Th(EeQ)$, $Th(Een) \subset Th(EenQ)$, and $Th(Ec) \subset Th(EcQ)$. After that, we define the degree of each logic connective for each above logic system from low degree to high degree. For example, we can define the degree for the logic fragments of EeQ in this order: first the logic fragment $Th^{(\Rightarrow,1)}(EeQ)$, then $Th^{(\Rightarrow,2)}(EeQ)$, finally the logic fragment $Th^{(\Rightarrow,3)}(EeQ)$ and so on. A defined semi-lattice of logic fragments of EcQ is shown in Fig. 2.

Phase 1.2 Deduce logical theorems

In this phase, we deduce logic theorems according to the partial order of the defined semi-lattice of logic fragments. We input the axioms and inference rule of EcQ, set the degree of each logic connective, and use FreeEnCal to deduce logical theorems automatically. Cheng conjectured that almost all new theorems and questions of a formal theory can be deduced from the premises of that theory by finite inference steps concerned with finite number of low degree entailments [3–5]. We set the degree of $\Rightarrow$ below 4, and set the degrees of $\neg$ and $\wedge$ below 2. We start to deduce the logic fragments from $Th^{(\Rightarrow,1)}(EeQ)$, but not $Th^{(\Rightarrow,1)}(Ee)$. First we input all the axioms of EeQ and set the degree of $\Rightarrow$ to 1 to deduce the logic fragment of $Th^{(\Rightarrow,1)}(EeQ)$. Second we input the deduced logical theorems of $Th^{(\Rightarrow,1)}(EeQ)$ as new premises and set the degree of $\Rightarrow$ to 2 to deduce $Th^{(\Rightarrow,2)}(EeQ)$. Third, we input the deduced logical theorems of $Th^{(\Rightarrow,2)}(EeQ)$ as new premises to deduce $Th^{(\Rightarrow,3)}(EeQ)$. After that, we put together the axioms $N1$–$N3$ of $EenQ$ with deduced logical theorems in the logic fragments $Th^{(\Rightarrow,k)}(EeQ)$ $(1 \le k \le 3)$ and set the degree of $\neg$ to 1 to deduce the logic fragments $Th^{(\Rightarrow,k,\neg,1)}(EenQ)$ $(1 \le k \le 3)$. Finally we put together the axioms $C1$–$C10$ of EcQ with deduced logic theorems in $Th^{(\Rightarrow,k,\neg,1)}(EenQ)$ $(1 \le k \le 3)$ and set the degree of $\wedge$ to 1 to deduce the logic fragments $Th^{(\Rightarrow,k,\neg,1,\wedge,1)}(EcQ)$ $(1 \le k \le 3)$.

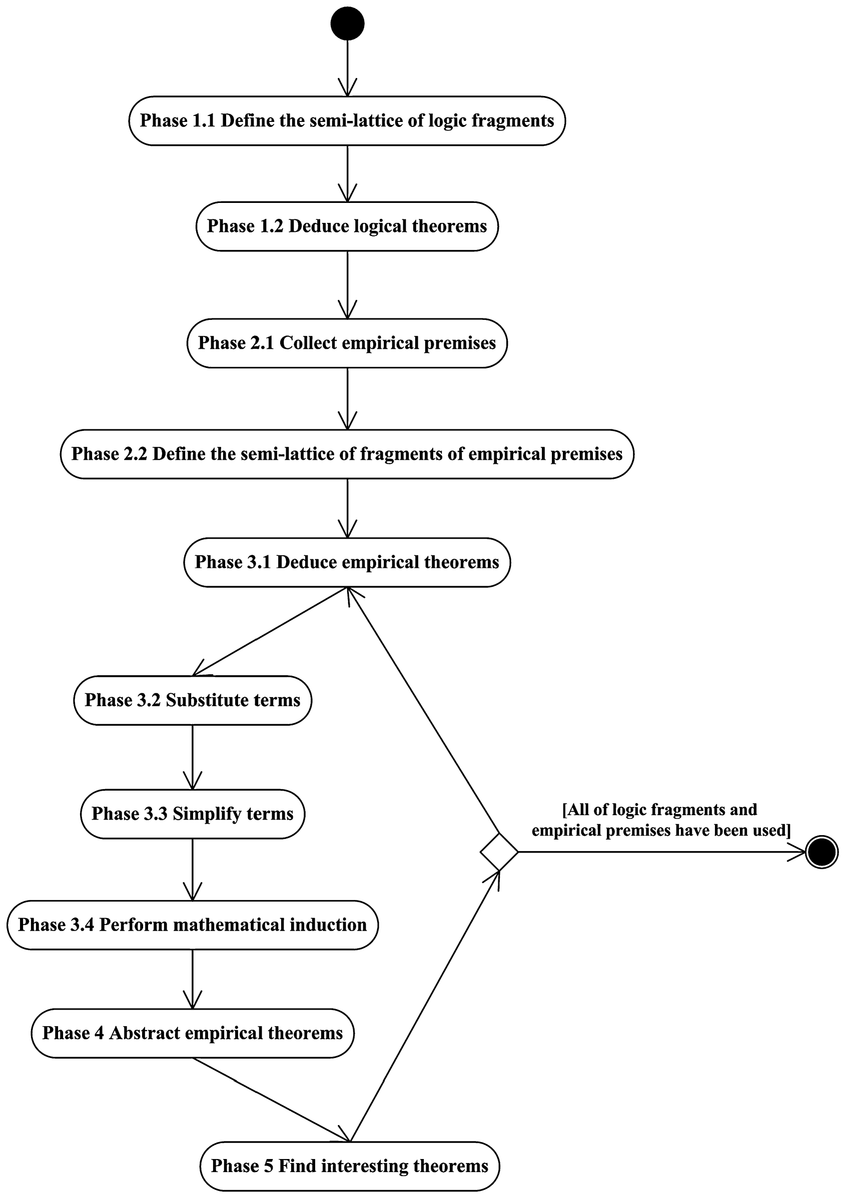

Fig. 1. The activity diagram of the systematic method for ATF.

Phase 2 Prepare empirical premises

The second phase is to prepare empirical premises. In the phase, we collect the definitions and axioms in the target field and draw up a plan to use those collected empirical premises to do ATF.

Phase 2.1 Collect empirical premises

In this phase, we collect the formalized definitions and axioms of the target field as empirical premises. We also collect the known theorems of the target field as more as possible, because the possibility to reason out new and interesting theorems will be increased, if we use the known theorems as empirical premises.

Phase 2.2 Define the semi-lattice of fragments of empirical premises

In this phase, we prepare (k, m)-fragments of collected empirical premises in the target field. A set of the prepared fragments and inclusion relation on the set is a partial order set, and is a finite semi-lattice. Moreover, a set of formal theories with the fragments and inclusion relation on the set is also a partial order set, and is also a finite semi-lattice. Partial order of the set of the prepared fragments can be used for a plan to reason out fragments of formal theories with collected empir-

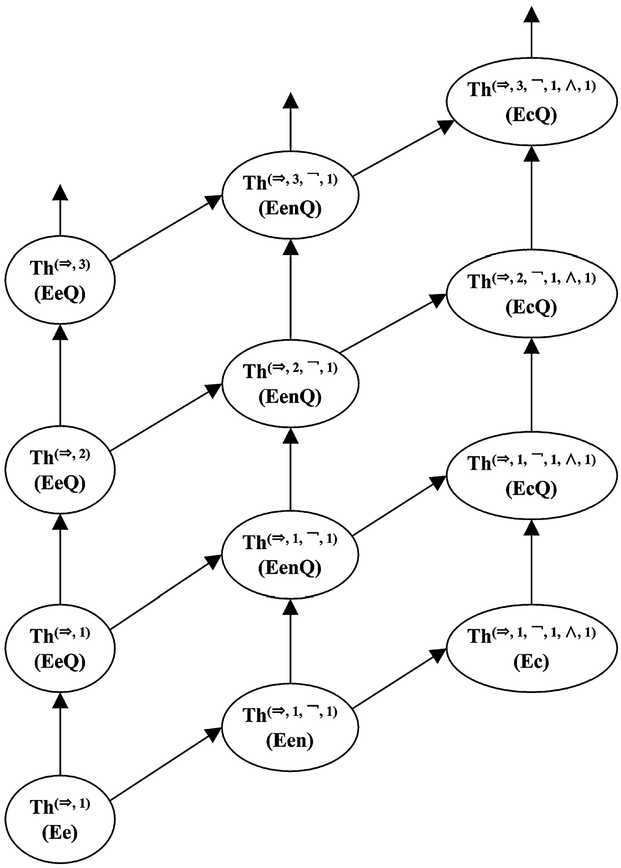

Fig. 2. The semi-lattice of logic fragments of EcQ.

ical premises. According to the partial order, we can systematically do ATF from simple theorems to complex theorems. In detail, first, we summarize all the predicate abstract levels of all of the predicates in collected empirical premises. Second, we summarize all the function abstract levels of all of the functions in collected empirical premises. Third, we summarize the abstract level of all of the collected empirical premises, i.e., axioms and definitions. Fourth, we use the elements of the semi-lattice to represent (k, m)-fragments of the premises in the target field, use the partial order to represent the inclusion relation between two fragments such that the semi-lattice of fragments of empirical premises is defined.

Phases 3–5 ATF loop by loop

We reason out empirical theorem, abstract empirical theorems, and find interesting empirical theorems loop by loop from Phase 3 to Phase 5 as shown in Fig. 1 such that we can do ATF systematically according to the partial order of defined semi-lattice of logic fragments and the partial order of defined semi-lattice of fragments of collected premises. In detail, we first choose the prepared minimum logic fragment to do ATF. After we chose logic fragment, we also use minimum abstract level fragment of premises to reason out empirical theorems rather than inputting all of premises. Then we enter into Phase 4 to abstract empirical theorems and Phase 5 to find interesting theorems from empirical theorems. After that, we come back to the Phase 3 and input the premises in the least upper bound of the inputted fragment of premises in the last loop and also input the reasoned out empirical theorems in the last loop to reason out next higher abstract level empirical theorems by using same logic fragment. Then, we enter into Phase 4 and Phase 5 to abstract theorems and find interesting theorems again. We deduce the empirical theorems by the method again and again until all of empirical premises have been inputted. After we input all premises, we choose the bigger logic fragment according to the partial order in the defined semi-lattice to do ATF continuously. Because there is not a maximum element in the defined semi-lattice of the logic fragments, the ATF process should be continued until we can find new and interesting theorems or we used all of the prepared logic fragments.

Phase 3 Reason out empirical theorems

In this phase, we reason out empirical theorems by forward deduction with strong relevant logics, substitution, simplifying terms, and mathematical induction. The reason why we separate substitution from forward deduction is that substitution as forward deduction is a quite time and memory consuming process. Substitution as forward deduction is to get formulas by substituting all terms in vocabulary of a target field for individual variables in each obtained formula. Our methodology specifies cases to do substitution. Mathematical induction is indispensable process to get empirical theorems in several fields, e.g., number theory. However, mathematical induction is a proving method because we have to prepare a proof target before doing mathematical induction. Thus, it is impossible to directly use mathematical induction for ATF. Our methodology provides a method to generate proof targets of mathematical induction from drawn formulas.

Phase 3.1 Deduce empirical theorems

In this phase, we use FreeEnCal to deduce empirical theorems automatically. We input empirical premises, chosen logic fragment, inference rule, and set the degree of each logic connective to deduce empirical theorems. We choose logic fragment according to the partial order of defined semi-lattice of logic fragments. We input empirical premises according to the partial order of defined semi-lattice of fragments of empirical premises. We deduce empirical theorems according to the partial order of the defined semi-lattice of fragments of premises loop by loop, rather than deducing empirical theorems one time. The deduced lower abstract level empirical theorems in the last loop should be also inputted as new empirical premises to deduce the higher abstract level empirical theorems.

Phase 3.2 Substitute terms

The phase is to choose those empirical theorems whose terms should be substituted, and to substitute their terms. Our method is to search those empirical theorems which can deduce new empirical theorems by substitution and then do the substitution of terms. For example, if the empirical theorems $\forall x\forall y((x = y) \Rightarrow (x \subseteq y))$ and $\forall x((x = (x \cap x))$ exist in deduced empirical theorems, the term "y" in the first empirical theorem should be substituted by function "$x \cap x$" such that a new empirical theorem "$\forall x(x \subseteq (x \cap x))$" can be found.

Phase 3.3 Simplify terms

In this phase, from the deduced and substituted empirical theorems, we find the $A = B$ type empirical theorems as simplifying rules, in which the degree of nested function in A is higher (or lower) than B, then we use the lower one to replace the higher one in all of the empirical theorems. For example, if a deduced empirical theorem is like the formula $\forall x(g(x, x) = g(x, g(x, x)))$ then we can simplify $g(x, g(x, x))$ with $g(x, x)$ in the other empirical theorems. As another example, if an empirical theorem is like $(0 + 0) = 0$, we simplified the terms $0 + 0$ in all of the empirical theorems with 0.

Phase 3.4 Perform mathematical induction

In this phase, we reason out empirical theorems by mathematical induction. This phase consists of two processes. The first one is a process to generate proof targets of mathematical induction from obtained theorems in previous sub-phases. The second one is a process to prove the targets by mathematical induction. Generating proof targets is done as follows. At first, we substitute the base element and the successor of the base element in a target field, e.g., 0 (zero) and $s(0)$ in number theory, for obtained theorems that form $\forall xA$ where A is a formula. In other words, we get all $A[x/0]$ and $A[x/s(0)]$ from obtained formulas $\forall xA$ where $A[x/t]$ means a term t substitutes for occurrence of x in sub-formula A. Secondly, we find all couples of formulas $B[x/0]$ and $B[x/s(0)]$ where B is a closed formula and $\forall xB$ is not included in a set of obtained formulas. Finally, we generate $\forall xB$ as a proof target of mathematical induction if we can find the couple of formulas $B[x/0]$ and $B[x/s(0)]$. For generating proof targets, we have to do substitution and simplification again in this phase. Obtained proof targets are proved by mathematical induction. We can use already existing automated theorem proving systems, e.g., OTTER [12]. If we succeed to prove the target, we adopt the target as an empirical theorem.

Phase 4 Abstract empirical theorems

In this phase, we abstract the empirical theorems according to the current level in the defined semi-lattice of abstract level fragments of empirical premises. In detail, for the abstraction of predicates, if a formula A holding a higher abstract level predicate is defined by the formula B holding other lower abstract level predicates by the axiom or definition, then we abstract B to A in the empirical theorems. For example if a definition is $\forall x\forall y(P(x, y) \Leftrightarrow (Q(x, y) \wedge Q(y, x)))$, then we can abstract $(Q(x, y) \wedge Q(y, x))$ to $P(x, y)$ in the empirical theorems. For the abstraction of functions, if a formula A holding a higher abstract level function is defined by the formula B holding other lower abstract level functions by the axiom or definition, then we abstract B to A in the empirical theorems. For example, if the definition of the function $f()$ is $\forall x(f(x) = g(x, x))$, then we can abstract $g(x, x)$ to $f(x)$ in the empirical theorems.

Phase 5 Find interesting theorems

In this phase, we use filtering methods to remove uninteresting theorems from deduced empirical theorems step by step, and then provide the rest theorems as candidates of interesting theorems [11].

Table 1
Degree of collected theorems.

$\Rightarrow$, 0	241	56%	$\wedge$, 0	359	83%	$\neg$, 0	404	94%
$\Rightarrow$, 1	188	44%	$\wedge$, 1	61	14%	$\neg$, 1	25	6%
$\Rightarrow$, 2	0	0%	$\wedge$, 2	7	2%	$\neg$, 2	0	0%
$\Rightarrow$, 3	0	0%	$\wedge$, 3	1	<1%	$\neg$, 3	0	0%
$\Rightarrow$, 4	0	0%	$\wedge$, 4	1	<1%	$\neg$, 4	0	0%

First, to conveniently analyze empirical theorems, we remove the quantifiers of all the empirical theorems and call those theorems "core empirical theorems" of deduced empirical theorems. If the core empirical theorem is an interesting theorem, then we can see the primitive theorem as candidate of interesting theorems [11].

Second, we check whether a formula includes a tautological sub-formula or not. If a formula includes a tautological sub-formula, the theorem should be filtered, because if one theorem contains a tautology part, this empirical theorem must not be an interesting empirical theorem. For example, $((x = x) \Rightarrow (x = x)) \Rightarrow (x \subseteq x)$ cannot be called an interesting theorem, because it is like $(A \Rightarrow A) \Rightarrow B$ which contains a tautology $A \Rightarrow A$.

Third, we filter those high degree empirical theorems although they do not contain tautology patterns. For example, $(((y \in x) \Rightarrow (y \in (x \cap x))) \Rightarrow (y \in (x \cap x))) \Rightarrow (((y \in (x \cap x)) \Rightarrow (y \in x)) \Rightarrow (y \in x))$ cannot be called an interesting theorem. This is based on our conjecture: In each abstract level of deduced empirical theorems, the logic connectives of the interesting theorems must be in low degree. For example, we have analyzed more than 400 known theorems of NBG set theory recorded in Quaife's book [12], and recorded our analysis results in Table 1. We can find that the logic connectives of those theorems are almost lower than degree 3.

4. Case study of preparation of logic fragments

In our case study, we chose strong relevant logic system EcQ [4] as fundamental logic system, and we used the forward reasoning engine FreeEnCal to deduce the logical theorems automatically. The logical connectives, axiom schemata [4], and inference rules of strong relevant logics EcQ are shown as follows.

Primitive logical connectives:

$\Rightarrow$: entailment
$\neg$: negation
$\wedge$: extensional conjunction

Axiom schemata:

E1: $A \Rightarrow A$
E2: $(A \Rightarrow B) \Rightarrow ((C \Rightarrow A) \Rightarrow (C \Rightarrow B))$
E2′: $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$
E3: $(A \Rightarrow (A \Rightarrow B)) \Rightarrow (A \Rightarrow B)$
E3′: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
E3″: $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$
E4: $(A \Rightarrow ((B \Rightarrow C) \Rightarrow D)) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow D))$
E4′: $(A \Rightarrow B) \Rightarrow (((A \Rightarrow B) \Rightarrow C) \Rightarrow C)$
E4″: $((A \Rightarrow A) \Rightarrow B) \Rightarrow B$
E4‴: $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (((A \Rightarrow C) \Rightarrow D) \Rightarrow D))$
N1: $(A \Rightarrow (\neg A)) \Rightarrow (\neg A)$
N2: $(A \Rightarrow (\neg B)) \Rightarrow (B \Rightarrow (\neg A))$
N3: $(\neg(\neg A)) \Rightarrow A$
C1: $(A \wedge B) \Rightarrow A$
C2: $(A \wedge B) \Rightarrow B$
C3: $((A \Rightarrow B) \wedge (A \Rightarrow C)) \Rightarrow (A \Rightarrow (B \wedge C))$
C4: $(LA \wedge LB) \Rightarrow L(A \wedge B)$, where $LA =_{df} (A \Rightarrow A) \Rightarrow A$
C5: $(A \wedge A) \Rightarrow A$
C6: $(A \wedge B) \Rightarrow (B \wedge A)$
C7: $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$
C8: $(A \wedge (A \Rightarrow B)) \Rightarrow B$
C9: $\neg(A \wedge \neg A)$
C10: $A \Rightarrow (B \Rightarrow (A \wedge B))$
IQ1: $\forall x(A \Rightarrow B) \Rightarrow (\forall xA \Rightarrow \forall xB)$
IQ2: $(\forall xA \wedge \forall xB) \Rightarrow \forall x(A \wedge B)$

Table 2
Prepared logic fragments.

Logic fragments	Logical theorem schemata
$Th^{(\Rightarrow,2)}(EeQ)$	60
$Th^{(\Rightarrow,3)}(EeQ)$	970
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	90
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	3098
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	311
$Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$	1,240,643+

IQ3: $\forall xA \Rightarrow A[t/x]$ (if x may appear free in A and t is free for x in A, i.e. free variables of t do not occur bound in A)
IQ4: $\forall x(A \Rightarrow B) \Rightarrow (A \Rightarrow \forall xB)$ (if x does not occur free in A)
IQ5: $\forall x_1 \cdots \forall x_n(((A \Rightarrow A) \Rightarrow B) \Rightarrow B)$ $(0 \leq n)$

Inference rules:

$\Rightarrow$E: "from A and $A \Rightarrow B$ to infer B" (Modus Ponens)
$\forall$I: "if A is an axiom, so is $\forall xA$" (Generalization of axioms)

Thus, strong relevant logic system EcQ are defined as follows, where we use '$A \mid B$' to denote any choice of one from two axiom schemata A and B.

$$\mathrm{Ee} = \left\{\mathrm{E1}, \mathrm{E2} \mid \mathrm{E2}', \mathrm{E3} \mid \mathrm{E3}', \mathrm{E4} \mid \mathrm{E4}'\right\} + \Rightarrow\mathrm{E}$$
$$\mathrm{Ee} = \left\{\mathrm{E2}', \mathrm{E3}, \mathrm{E4}''\right\} + \Rightarrow\mathrm{E}$$
$$\mathrm{Ee} = \left\{\mathrm{E1}, \mathrm{E3}, \mathrm{E4}'''\right\} + \Rightarrow\mathrm{E}$$
$$\mathrm{Een} = \mathrm{Ee} + \{\mathrm{N1}, \mathrm{N2}, \mathrm{N3}\}$$
$$\mathrm{Ec} = \mathrm{Een} + \{\mathrm{C3} \sim \mathrm{C10}\}$$
$$\mathrm{EeQ} = \mathrm{Ee} + \{\mathrm{IQ1}, \mathrm{IQ3} \sim \mathrm{IQ5}\} + \forall\mathrm{I}$$
$$\mathrm{EenQ} = \mathrm{Een} + \{\mathrm{IQ1}, \mathrm{IQ3} \sim \mathrm{IQ5}\} + \forall\mathrm{I}$$
$$\mathrm{EcQ} = \mathrm{Ec} + \{\mathrm{IQ1} \sim \mathrm{IQ5}\} + \forall\mathrm{I}$$

Phase 1 Prepare logical fragments for various empirical theories

According to our methodology, preparation of logic fragments was done by two sub-phases as follows.

Phase 1.1 Define a semi-lattice of logic fragments

We defined a finite semi-lattice for EcQ logic fragments to deduce the logical theorems and prepare the logic fragments systematically. We showed a defined semi-lattice in Fig. 2.

Phase 1.2 Deduce logical theorems

We used FreeEnCal as a tool to prepare logic fragments. Based on the defined semi-lattice, at present we prepared logic fragments of EcQ. We did not prepare the logic fragments in which the degree of $\Rightarrow$ is 1, because those logic fragments are not practical for the deduction of empirical theorems in the following two case studies. Table 2 shows the numbers of logical theorem schemata of the prepared logic fragments. We used axioms of EcQ to deduce logic theorems except C5, N3, and C10. We did not use C5 and N3, because FreeEnCal provided them as the elimination rules. The axiom C10 was not used, because C10 can make the logical fragment enlarge too fast. Furthermore, we did not use IQ3 and IQ4 in the phase of preparation of logic fragments, because they are used for deducing empirical theorems but not logical theorems. Besides, we tried to deduce $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$, however, we cannot deduce all of the logic theorem schemata of the logic fragment because of the current limited memory space, at present 1,240,643 logical theorem schemata have been deduced.

5. The case study in NBG set theory

The purpose of the case study is to confirm the effectiveness of our methodology. NBG set theory is a choice for the first case study, because it is the foundation of mathematics fields. Therefore, if we can find some new and interesting

Table 3
Used definitions of NBG set theory.

Inputted notions	Meaning	Type
$\subseteq$	subclass	predicate
{ }	singleton set	function
V	class	individual constant
$<,>$	ordered pair	function
0	null class	individual constant
E	elementhood relation	individual constant
$\cup$	binary union	function
$+$	symmetric difference	function
restrict	restriction	function
P	power class	function
U	sum class	function
inverse	inverse	function
R	range	function
“	image	function
succ	successor	function
SUCC	successor set	individual constant
INDUCTIVE	inductive set	predicate
ω	infinity	individual constant
$\circ$	composition	function
SINGVAL	single-valued class	predicate
FUNCTION	function	predicate
regular	Regularity	function
‘	functional application	function
choice	universal choice	individual constant
ONEONE	one-to-one function	predicate
S	subset relation	individual constant
I	identity relation	individual constant
diag	diagonalization	function
cantor	Cantor class	function
OPERATION	operation	predicate
COMPATIBLE	compatible function	predicate
HOM	homomorphism	predicate

theorems in NBG set theory, it is possible that we can also use the methodology to find some unknown theorems in other mathematics fields.

Phase 2 Prepare empirical premises

Phase 2.1 Collect empirical premises

Quaife recorded the axioms, definitions, and more than 400 known theorems of NBG set theory in his book [12]. In our case study, we chose all of the axioms and definitions in Quaife's book as empirical premises. We showed all the inputted axioms as below and also showed used definitions in our case study in Table 3.

- Axiom A-1: Sets are classes (omitted because all objects are classes).
- Axiom A-2: Elements of classes are sets.
  $\forall x(x \subseteq V)$.
- Axiom A-3: Extensionality.
  $\forall x \forall y((x = y) \Rightarrow (x \subseteq y))$.
  $\forall x \forall y((x = y) \Rightarrow (y \subseteq x))$.
  $\forall x \forall y((x \subseteq y) \wedge (y \subseteq x) \Rightarrow (x = y))$.
- Axiom A-4: Existence of unordered pair.
  $\forall u \forall x \forall y((u \in \{x, y\}) \Rightarrow (u = x) \vee (u = y))$.
  $\forall x \forall y((x \in V) \Rightarrow (x \in \{x, y\}))$.
  $\forall x \forall y((y \in V) \Rightarrow (y \in \{x, y\}))$.
  $\forall x \forall y((\{x, y\} \in V))$.
- Axiom B-1: E (elementhood relation).
  $(E \subseteq V \times V)$.
  $\forall x \forall y((< x, y > \in E) \Rightarrow (x \in y))$.
  $\forall x \forall y((< x, y > \in (V \times V)) \wedge (x \in y) \Rightarrow (< x, y > \in E))$.
- Axiom B-2: $\cap$ (binary intersection).
  $\forall z \forall x \forall y((z \in (x \cap y)) \Rightarrow (z \in x))$.
  $\forall z \forall x \forall y((z \in (x \cap y)) \Rightarrow (z \in y))$.
  $\forall z \forall x \forall y((z \in x) \wedge (z \in y) \Rightarrow (z \in (x \cap y)))$.

- Axiom B-3: $\sim$ (complement).
  $\forall z \forall x(\neg((z \in\sim(x)) \wedge (z \in x)))$.
  $\forall z \forall x((z \in V) \Rightarrow (z \in\sim(x)) \vee (z \in x))$.
- Axiom B-4: D (domain).
  $\forall x \forall z(\neg((restrict(x, \{z\}, V) = 0) \wedge (z \in D(x))))$.
  $\forall x \forall z((z \in V) \Rightarrow (restrict(x, \{z\}, V) = 0) \vee (z \in D(x)))$.
- Axiom B-5: $\times$ (Cartesian product).
  $\forall u \forall v \forall x \forall y((< u, v > \in (x \times y)) \Rightarrow (u \in x))$.
  $\forall u \forall v \forall x \forall y((< u, v > \in (x \times y)) \Rightarrow (v \in y))$.
  $\forall u \forall v \forall x \forall y((u \in x) \wedge (v \in y) \Rightarrow (< u, v > \in (x \times y)))$.
- Axiom B-6: *inverse*.
  $\forall x(inverse(x) \subseteq (V \times V))$.
  $\forall u \forall v \forall x((< u, v > \in inverse(x)) \Rightarrow (< u, v > \in (V \times V)))$.
  $\forall u \forall v \forall x((< u, v > \in inverse(x)) \Rightarrow (< v, u > \in x))$.
  $\forall u \forall v \forall x((< u, v > \in (V \times V)) \wedge (< v, u > \in x) \Rightarrow (< u, v > \in inverse(x)))$.
- Axiom B-7: *rotate*.
  $\forall x(rotate(x) \subseteq ((V \times V) \times V))$.
  $\forall x \forall u \forall v \forall w((<< u, v >, w > \in rotate(x)) \Rightarrow (<< v, w >, u > \in x))$.
  $\forall x \forall u \forall v \forall w((<< v, w >, u > \in x) \wedge (<< u, v >, w > \in ((V \times V) \times V)) \Rightarrow (<< u, v >, w > \in rotate(x)))$.
- Axiom B-8: *flip*.
  $\forall x(flip(x) \subseteq ((V \times V) \times V))$.
  $\forall x \forall u \forall v \forall w((<< u, v >, w > \in flip(x)) \Rightarrow (<< v, u >, w > \in x))$.
  $\forall x \forall u \forall v \forall w((<< v, u >, w > \in x) \wedge (<< u, v >, w > \in ((V \times V) \times V)) \Rightarrow (<< u, v >, w > \in flip(x)))$.
- Axiom C-1: Infinity.
  $INDUCTIVE(\omega)$.
  $\forall y(INDUCTIVE(\omega) \Rightarrow (\omega \subseteq y))$.
  $(\omega \in V)$.
- Axiom C-2: U (sum class).
  $\forall x((x \in V) \Rightarrow (U(x) \in V))$.
- Axiom C-3: P (power class).
  $\forall u((u \in V) \Rightarrow (P(u) \in V))$.
- Axiom C-4: Replacement.
  $\forall x \forall xf(FUNCTION(xf) \wedge (x \in V) \Rightarrow ((xf``x) \in V))$.
- Axiom D: Regularity.
  $\forall x((x = 0) \vee (regular(x) \in x))$.
  $\forall x((x = 0) \vee ((regular(x) \cap x) = 0))$.
- Axiom E: Universal choice.
  $FUNCTION(choice)$.
  $\forall y((y \in V) \Rightarrow (y = 0) \vee ((choice`y) \in y))$.

Phase 2.2 Define the semi-lattice of fragments of empirical premises

To do ATF from the simple theorems to complex theorems, we defined a semi-lattice of abstract level fragments of collected empirical premises in NBG set theory. First, we summarized all of the predicate abstract levels of the collected definitions and axioms as shown in Table 4. Second, we summarized all of the function abstract levels of the collected definitions and axioms as shown in Table 5. Third, we summarized all of the abstract levels of collected definitions and axioms as shown in Table 6. Finally, according to the abstract level of collected definitions and axioms, we defined the semi-lattice of abstract level fragments of empirical premises in NBG set theory as shown in Fig. 3. In the figure, "NBG" means all of collected premises of NBG set theory.

Phase 3 Reason out empirical theorems

In the case study, we did not use Phase 3.4 of our methodology, because mathematical induction is not an axiom of NBG set theory.

Phase 3.1 Deduce empirical theorems

We used the reasoning engine FreeEnCal to deduce empirical theorems of NBG set theory automatically. We set the degrees of $\Rightarrow$ to 2, $\neg$ to 1, and $\wedge$ to 1 to deduce the empirical theorems of NBG set theory. We used all of the prepared logic fragments to deduce empirical theorems except the logic fragment $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$. We did not use $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$ because of the limited memory space. In detail, first we used the logic fragment $Th^{(\Rightarrow,2)}(EeQ)$ to deduce empirical theorems of NBG set theory. We inputted the fragments of empirical premises according to the defined semi-lattice shown in Fig. 3.

Table 4
Predicate abstract level in NBG set theory.

Predicate	Abstract from	Level
$\in$	none	1
$\subseteq$	$\in$	2
$=$	$\subseteq$	3
INDUCTIVE	$\in$, $\subseteq$	3
SINGVAL	$\subseteq$	3
FUNCTION	$\subseteq$, SINGVAL	4
ONEONE	FUNCTION	5
OPERATION	FUNCTION, $=$, $\subseteq$	5
COMPATIBLE	FUNCTION, $=$, $\subseteq$	5
HOM	OPERATION, COMPATIBLE, $=$, $\in$	6

Table 5
Function abstract level in NBG set theory.

Function	Abstract from	Level
$\{,\}$	none	1
$\cap$	none	1
$\sim$	none	1
$\{\ \}$	$\{,\}$	2
$\cup$	$\sim$, $\cap$	2
$+$	$\sim$, $\cap$	2
regular	$\cap$	2
$<,>$	$\{,\}$, $\{\ \}$	3
succ	$\cup$, $\{\ \}$	3
$\times$	$<,>$	4
restrict	$\cap$, $\times$	5
rotate	$<,>$, $\times$	5
flip	$<,>$, $\times$	5
D	restrict, $\{\ \}$	6
inverse	D, flip, $\times$	7
diag	$\sim$, D, $\cap$	7
U	D, restrict	7
R	D, inverse	8
“	R, restrict	9
P	“,$\sim$	10
$\circ$	“, $\{\ \}$, $\times$, $<,>$	10
‘	U, “, $\{\ \}$	10
cantor	D, diag, inverse, $\circ$, $\cap$	11

In detail, we first inputted the axiom A-2 and definition of $\subseteq$ to deduce the empirical theorems of NBG set theory such that all of deduced empirical theorems were below $(2,0)$ abstract level. Then, we came into the next phase to substitute and simplify terms and to abstract the empirical theorems and find interesting empirical theorems in the current abstract level. After that, we came back to this phase and put together the deduced theorems which are below $(2,0)$ abstract level with the axiom A-3 and C-1 to deduce the empirical theorems which are below $(3,0)$ abstract level. We used the method to deduce empirical theorems again and again such that we deduced theorems from simple to complex. After all of empirical premises of NBG set theory have been used, we used the bigger logic fragment such as $Th^{(\Rightarrow,3)}(EeQ)$ to deduce empirical theorems again. We used the logic fragments according to the defined semi-lattice shown in Fig. 2.

Phase 3.2 Substitute terms

In the case study, after we deduced the empirical theorems, we searched all of the $\forall x \forall y \ldots (A \Rightarrow B)$ style empirical theorems and C style empirical theorems. For example, the theorem $\forall x \forall y((x = y) \Rightarrow (x \subseteq y))$ is a $\forall x \forall y \ldots (A \Rightarrow B)$ style empirical theorem, and $\forall x(x = (x \cap x))$ is a C style empirical theorem. Then we tried to match the empirical theorem C with the A part of the $\forall x \forall y \ldots (A \Rightarrow B)$ empirical theorems. If we substitute the instance in A part, the two empirical theorems can deduce new theorems by using modus ponens, then we performed substitution of terms.

Phase 3.3 Simplify terms

In the case study, we did not found any $A = B$ style empirical theorems deduced from axioms and definitions of NBG set theory. Although there are some $A = B$ style definitions in NBG set theory, such as $\forall x(\{x, x\} = \{x\})$, we did not use those definitions as rules to simplified terms but use them as abstraction rule to abstract empirical theorems in Phase 4. Therefore, we did not simplify terms for any empirical theorem in the case study.

Table 6
The abstract level of axioms and definitions in NBG set theory.

Abstract level	Axiom and definition
(1, 1)	Axiom B2, B3
(1, 4)	Axiom B1, B5
(1, 7)	Axiom C2
(1, 10)	Axiom C3
(2, 0)	Axiom A2, Definition of $\subseteq$
(2, 5)	Axiom B7, B8
(2, 7)	Axiom B6
(2, 10)	Definition of $\circ$
(3, 0)	Axiom A3, C1
(3, 1)	Axiom A4,
(3, 2)	Definition of { }, $\cup$, $+$, Axiom D
(3, 3)	Definition of $<$, $>$, succ
(3, 5)	Definition of restrict
(3, 6)	Axiom B4,
(3, 7)	Definition of inverse, U, diag
(3, 8)	Definition of R
(3, 9)	Definition of “, INDUCTIVE
(3, 10)	Definition of SINGVAL, P, ‘
(3, 11)	Definition of cantor
(4, 4)	Definition of FUNCTION
(4, 9)	Axiom C4
(4, 10)	Axiom E
(5, 7)	Definition of ONEONE
(5, 8)	Definition of OPERATION, COMPATIBLE
(6, 10)	Definition of HOM

Phase 4 Abstract empirical theorems

According to our methodology, the abstraction phase was not performed only one time, but performed loop by loop with Phase 3 and Phase 5. After the empirical theorems by inputting the fragment $P_{(k,m)}$ of premises were deduced, we used the definitions and axioms from the fragment $P_{(k,m)}$ of premises as the abstraction rules. Then, we found interesting empirical theorems and deduced next higher abstract level empirical theorems. In detail, after we deduced the empirical theorems by inputting $P_{(3,0)}$ as premises, we got two abstraction rules according to axiom A-3 and definition of $\subseteq$. The first abstraction rule is to abstract sub-formula $((x \subseteq y) \wedge (y \subseteq x))$ to $(x = y)$ and the other is to abstract sub-formula $((u \in x) \Rightarrow (u \in y))$ to $(x \subseteq y)$. We used the two abstraction rule to the deduced empirical theorems. Besides, we did not delete the primitive empirical theorems before we abstracted in the case study, because they are still useful for reasoning out new empirical theorems.

Phase 5 Find interesting theorems

In the case study, by using our filtering methods, we found four known theorems in NBG set theory as shown below. We found the Theorem 1 after we inputted the fragment $P_{(5,7)}$ of empirical premises of NBG set theory. We found the Theorems 2–4 after we inputted the fragment $P_{(6,10)}$ of empirical premises of NBG set theory. In Table 7, we showed those theorems were found by using which logic fragments.

- Theorem 1: $\forall x ONEONE(x) \Rightarrow \forall x(x \subseteq (V \times V))$
- Theorem 2: $\forall x ONEONE(x) \Rightarrow \forall x((x \circ inverse(x)) \subseteq I)$
- Theorem 3: $\forall x \forall y \forall z HOM(z, x, y) \Rightarrow \forall x \forall z(D(D(x)) = D(z))$
- Theorem 4: $\forall x \forall y \forall z HOM(z, x, y) \Rightarrow \forall y \forall z(R(z) \subseteq D(D(y)))$

The ultimate goal of ATF is to find new and interesting theorems. However, we only rediscovered some simple known theorems but did not find new and interesting theorems in the case study. We consider that the main reason is involved in the restriction of the performance of our current equipment but not our methodology. In the case study, we only set the degree below 3 for logic connective $\Rightarrow$, and set degree 1 for logic connectives $\neg$ and $\wedge$. Besides, we did not input axiom $C10$ of EcQ which plays an important role in the reasoning process, because $C10$ will make the deduced theorems set enlarge so fast. In the case study, in spite of different logic fragments we used, we found same known theorems, which indicated us the logical theorems in our prepared logic fragments were not enough to deduce more interesting empirical theorems and we have to deduce higher degree logic fragments. On the other hand, in spite of the restriction of the performance of our current equipment and the restriction of prepared logical theorems, we have rediscovered several known theorems of NBG set theory by using our methodology. If we can prepare higher degree logic fragments to do ATF by using the methodology, it is hopeful to find new and interesting theorems in the future.

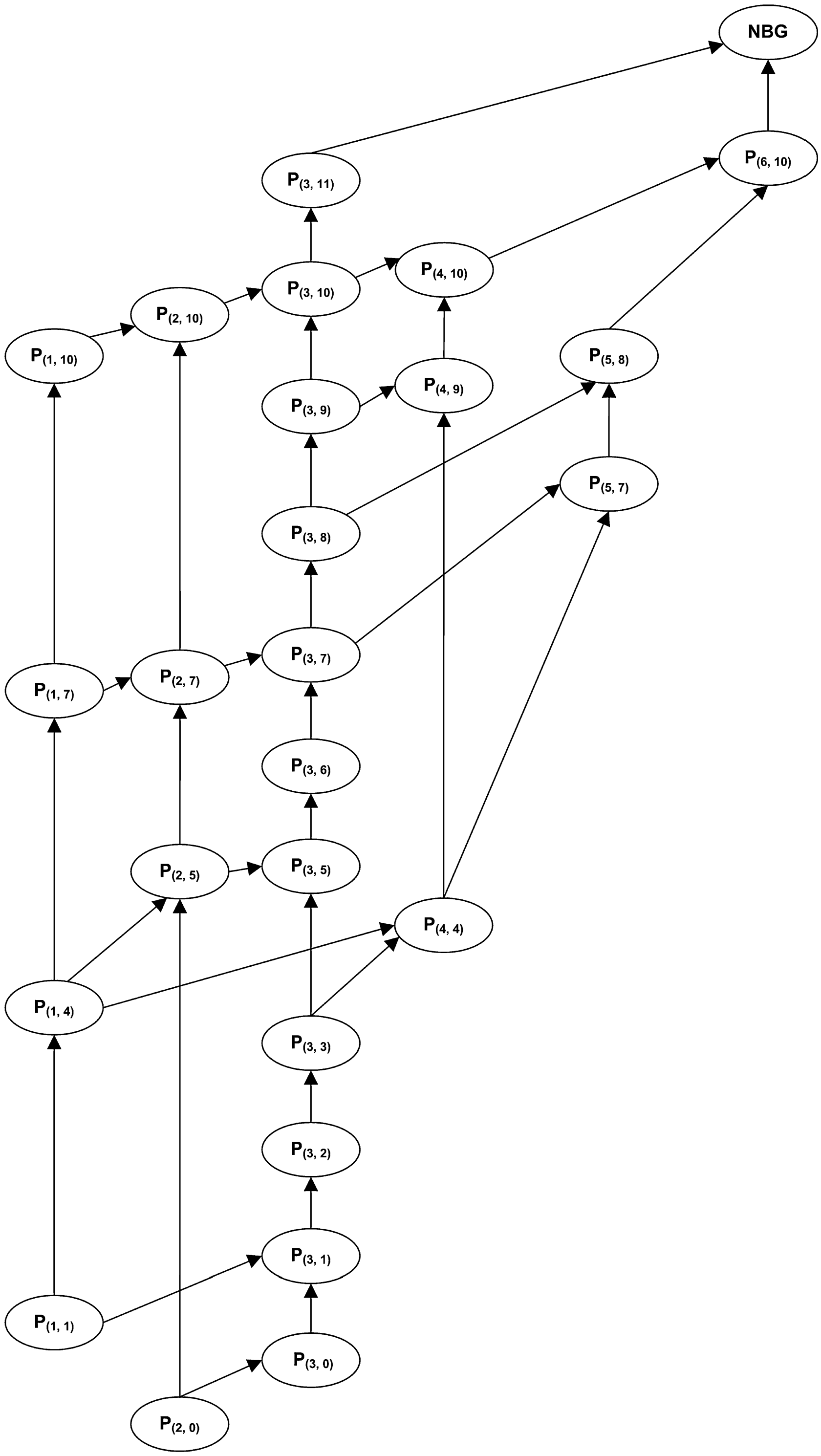

Fig. 3. The semi-lattice of fragments of premises in NBG set theory.

6. The case study in Peano's arithmetic

The purpose of this case study is to confirm the generality of our methodology. After we did a case study of NBG set theory, we have to use our methodology in another field to confirm the generality. The Peano axioms [12] which are a set of

Table 7
ATF by prepared logic fragments.

Used logic fragments	Found theorems
$Th^{(\Rightarrow,2)}(EeQ)$	Theorems 1–4
$Th^{(\Rightarrow,3)}(EeQ)$	Theorems 1–4
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	Theorems 1–4
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	Theorems 1–4
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	Theorems 1–4

Table 8
Used definitions of predicates in Peano's arithmetic.

Inputted notions	Meaning
$<$	ordering
DIV	divisibility
LD	linear difference predicate
PR	prime
On	on second argument
SUB	sublist
SET	set
SORTED	sorted list
PERM	perm
PRIMES	list of primes
PP	prime power predicate
PPOWERS	list of prime power
BSORTED	base sorted list
$\equiv$	congruence
INCONG	incongruent
COMPLETE	complete
CRS	complete residue system
RP	relative primality
RCOMPLETE	reduced complete
RCRS	reduced complete residue system

axioms for the natural numbers presented by the 19th century Italian mathematician Giuseppe Peano, are well formalized by mathematical logic. We chose Peano's arithmetic as the second field of our case study.

Phase 2 Prepare empirical premises

Phase 2.1 Collect empirical premises

Quaife recorded the axioms, definitions and more than 1,200 known theorems in his book [12]. In our case study, we chose all of the Peano's axioms recorded in Quaife's book as empirical premises (in Quaife's book, "$1+x$" means the successor of "x", and we used the function s() to represent the successor in our case study).

- Axiom 1: The successor function is one-to-one.
  $\forall x \forall y(((1+x)=(1+y)) \Rightarrow (x=y))$
- Axiom 2: Zero is not a successor.
  $\forall x(\neg((1+x)=0))$
- Axiom 3: First recursion equation for addition.
  $\forall x((0+x)=x)$
- Axiom 4: Second recursion equation for addition.
  $\forall x \forall y(((1+x)+y)=(1+(x+y)))$
- Axiom 5: First recursion equation for multiplication.
  $\forall x((0*x)=0)$
- Axiom 6: Second recursion equation for multiplication.
  $\forall x \forall y(((1+x)*y)=(y+(x*y)))$
- Axiom 7: The Mathematical Induction.
  $(P(0) \wedge \forall n(p(n) \Rightarrow p(1+n))) \Rightarrow \forall n(p(n))$

Besides of the seven axioms, we also used all the definitions of Peano's arithmetic recorded in Quaife's book as empirical premises. We recorded the definitions of predicates and functions we used in our case study in Tables 8 and 9.

Table 9
Used definitions of functions in Peano's arithmetic.

Inputted notions	Meaning
$-$	difference
min	minimum
max	maximum
mod	remainder
/	quotient
ld1	first linear difference coefficient
ld2	second linear difference coefficient
gcd	non-zero linear differences
gcd1	gcd coefficient
gcd2	gcd coefficient
lcm	least common multiple
lf	least factor
gf	great factor
!	factorial
$[\mid]$	pairing function
app	append
rev	reverse
head	head
tail	tail
at	nth tail
len	length
ht	components
card	cardinality
set	set function
merge	merge function
sort	sort function
del	delete function
$\sum$	sum of a list
const	list of length y of identical elements x
x^y	exponentiation
log	logarithm
$\prod$	product of a list
pfact	list of prime factors of a number
lpp	least prime power factor of a number
ppfact	prime power factorization
init	initial segment
modlist	mod of a list
times	times
red	reduction function
φ	Euler's φ function

Phase 2.2 Define the semi-lattice of fragments of empirical premises

To do ATF from the simple theorems to complex theorems, we defined a semi-lattice of abstract level fragments of collected empirical premises in Peano's arithmetic. First, we summarized all of the predicate abstract levels of the collected definitions and axioms as shown in Table 10. Second, we summarized all of the function abstract levels of the collected definitions and axioms as shown in Table 11. Third, we summarized all of the abstract levels of collected definitions and axioms as shown in Table 12. Finally, according to the abstract level of collected definitions and axioms, we defined the semi-lattice of abstract level fragments of empirical premises in Peano's arithmetic as shown in Fig. 4. In the figure, "Peano" means all of collected premises in Peano's arithmetic.

Phase 3 Reason out empirical theorems

Phase 3.1 Deduce empirical theorems

In the second case study, the reasoning engine FreeEnCal was chosen as the tool to deduce empirical theorems automatically. We set the degrees of $\Rightarrow$ to 2, $\neg$ to 1, and $\wedge$ to 1 to deduce the empirical theorems of NBG set theory. Same as the first case study, because of the limited memory space, we also used all of the prepared logic fragments except the logic fragment $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$ to deduce empirical theorems. The order of using logic fragments was according to the partial order of the defined semi-lattice of logic fragments shown in Fig. 2, that is the same as the case study of NBG set theory. After we chose one logic fragment, we inputted the empirical premises according to the partial order of the defined semi-lattice of fragments of collected premises shown in Fig. 4. In detail, first we inputted the axioms 1–6 of Peano's arithmetic (except mathematical induction) to deduce the empirical theorems such that all of the deduced theorems were below $(1, 1)$ abstract level. Then, we came into the next phase to substitute and simplify terms, induced empirical theorems, abstracted theorems and found interesting theorems from empirical theorems below $(1, 1)$ abstract level. After that, we came back to this phase and put together the reasoned out theorems in the last loop with the definitions of $-$, mod(), /, !, x^y to deduce

Table 10
Predicate abstract level in Peano's arithmetic.

Predicate	Abstract from	Level
$=$	none	1
$<$	$=$	2
DIV	$=$	2
LD	$=$	2
SET	$=$	2
SORTED	$=, <$	3
PERM	$=$	2
On	$=$	2
$\equiv$	$=$	2
PR	$<, =$	3
SUB	On	3
PP	$<, =$	3
BSORTED	$<, =$	3
INCONG	$<, =, \equiv$	3
COMPLETE	On, $=, \equiv$	3
CRS	PERM	3
RP	On, $=$	3
RCOMPLETE	On, $=, \equiv$	3
RCRS	PERM	3
PRIMES	$=$, PR	4
PPOWERS	PP, $=$	4

Table 11
Function abstract level in Peano's arithmetic.

Function	Abstract from	Level
s	none	1
$+$	none	1
$*$	none	1
gcd	none	1
lf	none	1
$-$	$+$	2
mod	$*$	2
/	$*$	2
!	$*$, s	2
x^y	$*$, s	2
log	/, s	3
min	$+, -$	3
max	$+, -$	3
ld1	$*, -$	3
ld2	$*, -$	3
lcm	gcd, $*$, /	3
gf	/, lf	3
[|]	$+, *$, /, s	3
head	[|]	4
tail	[|]	4
app	[|]	4
len	[|]	4
gcd1	gcd, ld1	4
gcd2	gcd, ld2	4
card	s, [|]	4
set	[|]	4
del	[|]	4
$\sum$	[|], $+$	4
const	[|], s	4
$\prod$	[|], $*$	4
pfact	lf, gf, [|]	4
lpp	lf, log	4
times	[|], $*$	4
red	gcd, [|]	4
modlist	mod, [|]	4
ppfact	[|], lpp, /	5
init	app, [|], s	5
rev	[|], app	5
at	s, tail	5
merge	[|], head, tail	5
sort	[|], merge	6
ht	head, at	6
φ	init, red, len	6

Table 12
The abstract level of axioms and definitions in Peano's arithmetic.

Abstract levels	Axioms and definitions
(1, 1)	Axioms of Peano's arithmetic
(1, 2)	Definition of $-$, mod, /, !, x^y
(1, 3)	Definition of min, max, ld1, ld2, lcm, gf, [|]
(1, 4)	Definition of gcd1, gcd2, app, len, del, $\sum$, const, $\prod$, lpp, modlist, times, red, head, tail
(1, 5)	Definition of rev, at, init
(1, 6)	Definition of ht, sort, φ
(2, 1)	Definition of gcd, lf
(2, 2)	Definition of $<$, DIV, $\equiv$
(2, 3)	Definition of LD, ON, log
(2, 4)	Definition of card, SET, set, PERM, pfact
(2, 5)	Definition of merge, ppfact
(3, 0)	Definition of SUB
(3, 1)	Definition of COMPLETE, RCOMPLETE, PR, RP
(3, 4)	Definition of PP, BSORTED, SORTED
(3, 5)	Definition of CRS, RCRS
(3, 6)	Definition of INCONG
(4, 4)	Definition of PRIMES, PPOWERS

empirical theorems whose abstract levels are below $(1, 2)$. We used the method to deduce empirical theorems again and again such that we deduced theorems from simple to complex.

Phase 3.2 Substitute terms

In the case study, we also use the method to substitute the terms same as the method we used in the first case study. Besides, we also substituted the instance 0 and s(0) to the deduced empirical theorems that form $\forall x A$, because we performed the mathematical induction in this case study.

Phase 3.3 Simplify terms

We collected the $A = B$ type empirical theorems from deduced empirical theorems as simplifying rules. We performed the collected simplifying rules automatically by FreeEnCal as elimination rules. For example, after we inputted the fragment $P_{(1,2)}$ of collected premises, the empirical theorem $\forall x((x - x) = 0)$ was deduced. We collected it as a simplifying rule and replaced all of the terms like $x - x$ with 0 in other deduced empirical theorems. By using the simplifying rule, the deduced empirical theorem $\min(0, 0) = (0 - (0 - 0))$ was simplified to $\min(0, 0) = 0$ in the case study. In the case study, we did not delete the primitive empirical theorems before we simplified, because they were still useful for reasoning out new empirical theorems.

Phase 3.4 Perform mathematical induction

In this phase, we used our method to generate proof target by observing the deduced empirical theorems and to prove the generated target. In Peano's arithmetic, the base element is 0, so we substituted 0 and s(0) as an instance to the deduced theorems that are like $\forall x A$ where A is a formula. Then we tried to find $B[x/0]$ and $B[x/\mathrm{s}(0)]$, and if we can find them in the deduced empirical theorems, we used mathematical induction to generate a hypothesis $\forall x B(x)$ as a proof target. For example, we substituted 0 to all of the deduced empirical theorems by inputting the fragment $P_{(1,1)}$ of empirical premises. Then, we found two empirical theorems $(0 + 0) = 0$ and $(\mathrm{s}(0) + 0) = \mathrm{s}(0 + 0)$. Because the empirical theorem $(0 + 0) = 0$ was deduced, we collected it as a simplifying rule and we simplified the term $0 + 0$ with 0 in the empirical theorems. Therefore, the empirical theorem $(\mathrm{s}(0) + 0) = \mathrm{s}(0)$ was found. Because we found the theorem $(0 + 0) = 0$ (can be seen as $B[x/0]$) and the theorem $(\mathrm{s}(0) + 0) = \mathrm{s}(0)$ (can be seen as $B[x/\mathrm{s}(0)]$), by using our method, we generated the proof target $\forall x((x + 0) = x)$. In our case study, besides of $\forall x((x + 0) = x)$, we also generated two proof targets $\forall x(\min(x, x) = x)$ and $\forall x(\max(x, x) = x)$. The next phase is to use the existing ATP system to prove those theorems, such as OTTER. Quaife used OTTER to prove those theorems and recorded them in his book [12], so we did not prove those known theorems again.

Phase 4 Abstract empirical theorems

Same as the case study of NBG set theory, we did not perform the abstraction phase one time, but performed the phase with Phase 3 and Phase 5 loop by loop. After we reasoned out the empirical theorems by inputting the fragment $P_{(k,m)}$ of premises, we chose the axioms and definitions from the $P_{(k,m)}$ of premises as abstraction rules. For example, after we reasoned out all of the theorems by inputting the fragment $P_{(2,2)}$ of premises, we changed all the sub-formula $((y \bmod x) = 0)$ to $DIV(x, y)$ according to the definition of $DIV()$. In the case study, we did not delete the primitive empirical theorems before we abstracted, because they are still useful for reasoning out new empirical theorems.

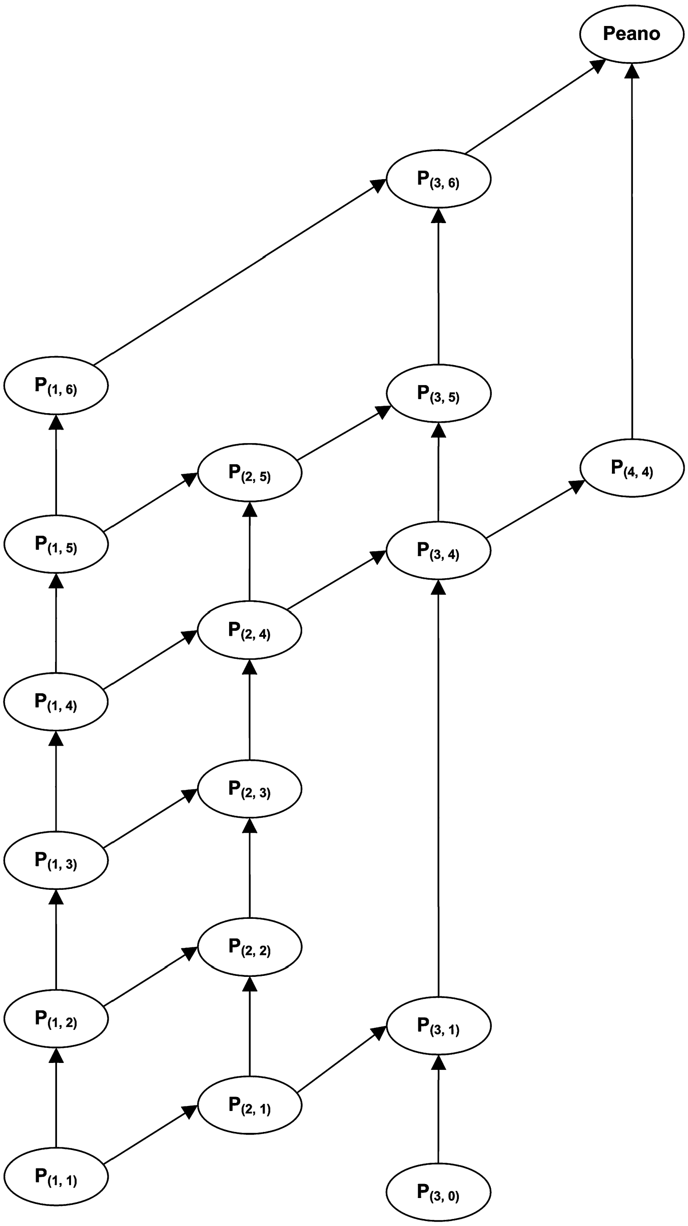

Fig. 4. The semi-lattice of fragments of premises in Peano's arithmetic.

Phase 5 Find interesting theorems

In the second case study, we also used the filtering methods to remove uninteresting empirical theorems, and from the rest of interesting candidates we found five known theorems in Peano's arithmetic as shown below. The first theorem was found after we inputted the fragment $P_{(1,1)}$ of collected premises. Theorem 2 was found after we inputted the fragment $P_{(1,2)}$ of collected premises. The third and fourth theorems were found after we inputted the fragment $P_{(1,3)}$ of collected premises. The fifth theorem was found after we inputted the fragment $P_{(2,1)}$ of collected premises. We found Theorem 1, Theorem 3 and Theorem 4 as proof targets, which has been explained in Phase 3.4. In Table 13, we showed those theorems were found by using which logic fragments.

- Theorem 1: $\forall x((x+0)=x)$
- Theorem 2: $\forall x((x-x)=0)$
- Theorem 3: $\forall x(\min(x,x)=x)$
- Theorem 4: $\forall x(\max(x,x)=x)$
- Theorem 5: $\forall x(\neg(1<\mathrm{lf}(x))\Rightarrow(lf(x)=x))$

Table 13
ATF by prepared logic fragments.

Used logic fragments	Found theorems
$Th^{(\Rightarrow,2)}(EeQ)$	Theorems 1–4
$Th^{(\Rightarrow,3)}(EeQ)$	Theorems 1–4
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	Theorems 1–4
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	Theorems 1–5
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	Theorems 1–4

In the second case study, we used the same logic fragments used in the first case study to do ATF and we found known theorems in the two different fields, which shows that our methodology holds generality. Furthermore, the result of the case study shows that bigger logic fragment is used, the possibility to deduce interesting theorems is bigger. Because our methodology provides a continuous and systematic way to prepare the logic fragments and use them to do ATF, if we do ATF by using our methodology continuously, it is possible to find a new and interesting theorem in the future.

7. Discussion

At present, we have not discovered any new and interesting theorems, but rediscovered some known theorems in our case studies. To find new theorems, we have to get more complex theorems rather than theorems we have reasoned out in our case study. A complex theorem is a theorem whose degree of nested logical connectives and/or functions is high. However, according to our methodology, it takes long execution time and needs huge amount of memory space, as obtained theorems become more complex. On the other hand, to find interesting theorems, a method to excavate interesting theorems is demanded, but our current methods are just to remove explicitly trivial theorems. To solve the ATF problem completely by using our methodology, the two problems should be solved.

Generating new predicates and functions from obtained empirical theorems is a way to get more complex theorems. By replacing the generated new predicates and functions with sub-formulas and nested functions in obtained empirical theorems, we can continue to reason out empirical theorems and keep the degree of nested logical connectives and functions low. However, in our current methodology, predicates and functions are defined by scientists before they start to do ATF. Thus, we should make an environment to automatically or semi-automatically provide candidates of new predicates and functions for scientists during doing ATF. From viewpoint of syntax, the environment extracts sub-formulas and nested functions in obtained empirical theorems according to filtering rules previously given by scientists who are doing ATF. From viewpoint of semantics, the scientist chooses meaningful sub-formulas and nested functions, and then defines new predicates and functions to abstract the sub-formulas and functions. The environment and the scientists work interactively. Epistemic programming approach [4] and its language EPLAS [9,13] have been proposed and are hopeful to construct such the environment, because EPLAS is designed to help scientific discovery by working with scientists interactively.

Excavating interesting theorems from obtained empirical theorems is the most difficult problem in ATF. The results of reasoning phase in our methodology can be classified into 4 classes: intermediates, explicitly uninteresting theorems, implicitly uninteresting or interesting theorems, and explicitly interesting theorems. Intermediates are results that are not closed formulas. Explicitly uninteresting theorems are closed formulas that are regarded as uninteresting according to some explicit criteria which have been accepted by scientists of the target field. Implicitly uninteresting theorems (or implicitly interesting theorems) are closed formulas that may be uninteresting (or interesting). Explicitly interesting theorems are closed formulas that are or will be regarded as interesting according to some explicit criteria that have been accepted by scientists of the target field. Explicitly interesting theorems can be classified into already known theorems and new theorems. Fig. 5 shows a partition of results of the reasoning phase from view point of degree of interesting. Ideal purpose of the finding phase in our methodology is to find only new and explicitly interesting theorems from the results, and give them to scientists of the target field. To achieve the purpose, we should find a method to reduce implicitly uninteresting or interesting theorems in the results, in other words, make "implicitly" into "explicitly". Under the consideration, the issues of excavation of interesting theorems in our methodology are as follows.

- How can we decide what theorems are explicitly uninteresting?
- How can we decide what theorems are explicitly interesting?
- How can we collect all of already known explicitly interesting theorems?
- How can we measure the degree of interesting of theorems?

8. Concluding remarks

We have presented our systematic and general methodology for ATF. We have shown our two case studies in NBG set theory and Peano's arithmetic in which we used our methodology to find several known theorems. We also proposed some future research directions for ATF.

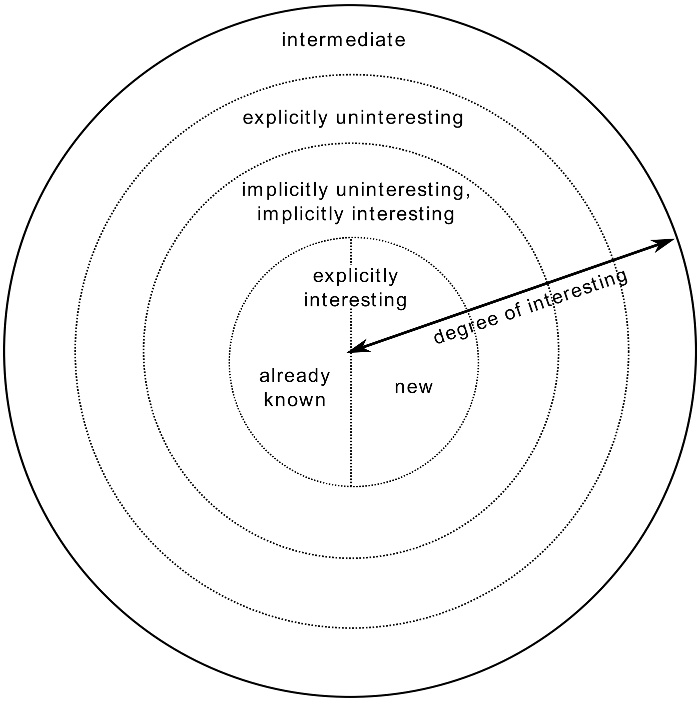

Fig. 5. Excavation problem of ATF.

There are many interesting and challenging research problems in our future works. First, to find more interesting theorems, we will deduce higher degree logic fragments and then deduce higher degree empirical theorems. Second, we will follow Cheng's epistemic programming approach [4] and use EPLAS [9,13] to provide an interactive environment for ATF, which can assist us to do ATF automatically or semi-automatically with our methodology. Finally, the semi-lattice model of formal theories [5,7] proposed by Cheng is aimed to do ATF in multi-fields, in which the axiomatic set theory can be seen as the minimum element and above it other fields can be established, such as number theory, graph theory, and combinatorics. We will use our methodology with Cheng's semi-lattice of formal theories to do ATF systematically in multi-fields in future.

References

[1] A.R. Anderson, N.D. Belnap Jr., Entailment: The Logic of Relevance and Necessity, vol. 1, Princeton University Press, 1975.
[2] A.R. Anderson, N.D. Belnap Jr., J.M. Dunn, Entailment: The Logic of Relevance and Necessity, vol. 2, Princeton University Press, 1992.
[3] J. Cheng, Entailment calculus as the logical basis of automated theorem finding in scientific discovery, in: Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium, AAAI Press – American Association for Artificial Intelligence, 1995, pp. 105–110.
[4] J. Cheng, A strong relevant logic model of epistemic processes in scientific discovery, in: Information Modelling and Knowledge Bases XI, in: Frontiers Artificial Intelligence Appl., vol. 61, IOS Press, 2000, pp. 136–159.
[5] J. Cheng, A semilattice model for the theory grid, in: Proc. 3rd International Conference on Semantics, Knowledge and Grid, IEEE Computer Society, 2007, pp. 152–157.
[6] J. Cheng, S. Nara, Y. Goto, FreeEnCal: a forward reasoning engine with general-purpose, in: The 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, in: LNCS (LNAI), vol. 4693, Springer, Heidelberg, 2007, pp. 444–452.
[7] J. Cheng, S. Nara, Y. Goto, T. Koh, A cooperative grid computing approach to automated theorem finding and automated problem proposing, in: The 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, in: LNCS (LNAI), vol. 4693, Springer, Heidelberg, 2007, pp. 840–851.
[8] B. Davey, H. Priestley, Introduction to Lattices and Order, 2nd edition, Cambridge University Press, 2002.
[9] W. Fang, I. Takahashi, Y. Goto, J. Cheng, Practical implementation of EPLAS: an epistemic programming language for all scientists, in: Proc. 10th International Conference on Machine Learning and Cybernetics, The IEEE Systems, Man, and Cybernetics Society, 2011, pp. 608–616.
[10] H. Gao, K. Shi, Y. Goto, J. Cheng, Automated theorem finding by forward deduction based on strong relevant logic: a case study in NBG set theory, in: Proc. 11th International Conference on Machine Learning and Cybernetics, The IEEE Systems, Man, and Cybernetics Society, 2012, pp. 1859–1865.
[11] H. Gao, K. Shi, Y. Goto, J. Cheng, Finding theorems in NBG set theory by automated forward deduction based on strong relevant logic, in: The 19th International Conference on Computing and Combinatorics, in: LNCS, vol. 7936, Springer, Heidelberg, 2013, pp. 697–704.
[12] A. Quaife, Automated Development of Fundamental Mathematical Theories, Kluwer Academic, 1992.
[13] I. Takahashi, S. Nara, Y. Goto, J. Cheng, EPLAS: an epistemic programming language for all scientists, in: The 7th International Conference on Computational Science, in: LNCS, vol. 4487, Springer, Heidelberg, 2007, pp. 406–413.
[14] L. Wos, Automated Reasoning: 33 Basic Research Problem, Prentice–Hall, 1988.
[15] L. Wos, The problem of automated theorem finding, J. Automat. Reason. 10 (1) (1993) 137–138.