Doctoral Dissertation

Automated Theorem Finding by Forward Reasoning Based on Strong Relevant Logic: Methodology and Case Studies

Hongbiao Gao

Graduate School of Science and Engineering,
Saitama University

Supervisor: Professor Jingde Cheng

March 2015

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. The problem of automated theorem finding is "What properties can be identified to permit an automated reasoning program to find new and interesting theorems, as opposed to proving conjectured theorems?". 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. 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.

The problem of automated theorem finding is still an open problem. Although there have been valuable works on the research of automated theorem proving, those works cannot be applied to the problem of automated theorem finding. On the other hand, a few works aimed to automated theorem discovery and automated theorem generation have been done. However, the problem of automated theorem finding which is to pursue properties to find new and interesting theorems is different from the automated theorem discovery and automated theorem generation. In fact, Wos's problem can be regarded as an attempt to find a systematic methodology in automated reasoning area, but the works on automated theorem discovery and automated theorem generation are not. The works on automated theorem discovery and automated theorem generation almost aimed to one certain mathematical field and current results of those works are only rediscovery of some simple known theorems in some certain fields, rather than finding new and interesting theorems.

Cheng proposed that forward reasoning based on strong relevant logic is a hopeful approach to solve the automated theorem finding problem. Reasoning is the process of drawing new conclusions from some premises which are already known facts and/or previously assumed hypotheses. 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. On the other hand, by using strong relevant logic as the fundamental logic to underlie reasoning for automated theorem finding, one can avoid paradoxical theorems in using classical mathematical logic, various conservative extensions of classical mathematical logic, and traditional relevant logics. However, no one showed how to use a systematic method by forward reasoning based on strong relevant logic to do automated theorem finding.

This thesis proposed a systematic methodology for automated theorem finding by forward reasoning based on strong relevant logic. The methodology consists of five phases. The first phase is to prepare logical fragments of strong relevant logic for various empirical theories. The second phase is to prepare empirical premises of a target empirical theory. The third phase is to reason out empirical theorems in the target empirical theory. The fourth phase is to abstract these empirical

theorems. The fifth phase is to find interesting theorems from the empirical theorems. The methodology holds generality so that we can use it to do automated theorem finding in various fields.

In order to show the effectiveness of our methodology, we did three case studies of automated theorem finding in three different mathematical fields by using our methodology. The first mathematical field is NBG set theory, the second one is Peano's arithmetic and the third one is graph theory. For each case study, we elaborated how to apply our methodology, showed the results and gave an evaluation. After we presented three case studies, we evaluated the methodology from viewpoint of generality.

This work has following contributions. The first contribution is that we proposed a systematic methodology for automated theorem finding based on the semilattice of formal theories in which the core is strong relevant logic, and the minimum element is the formal theory of axiomatic set theory, above it other formal theories can be established like number theory, graph theory, and lattice theory, so the methodology holds generality for various mathematical fields. The second contribution is that we proposed a method to do automated theorem finding based on the abstraction process of mathematical concept such that we can systematically find theorems from simple theorems to complex theorems. The third contribution is that we proposed a method to generate hypothesis by using forward reasoning approach by strong relevant logic and then combine automated theorem proving approach to systematically find those theorems proved by mathematical induction. The fourth contribution is that we performed three case studies of automated theorem finding in three different mathematical fields by using our methodology and clearly showed our method and results. Before our works, it is only in theory to use forward reasoning approach based on strong relevant logic to perform automated theorem finding in different mathematical fields, but our works showed the detail and systematic procedure of automated theorem finding clearly.

This thesis is organized as follows. Chapter 1 presents the background and purpose of this research. Chapter 2 explains the basis of the strong relevant logic and the terminology of automated theorem finding. Chapter 3 presents our systematic methodology for automated theorem finding. Chapter 4 presents the case study of preparation of logical fragments. Chapter 5 presents the case study of automated theorem finding in NBG set theory. Chapter 6 presents the case study of automated theorem finding in Peano's arithmetic. Chapter 7 presents the case study of automated theorem finding in graph theory. Chapter 8 gives a discussion about our methodology. Finally, concluding remarks are given in Chapter 9.

Acknowledgments

I would like to express special thanks first of all to my research supervisor Professor Jingde Cheng for his patience, invaluable guidance, and suggestions on all aspects of my academic life. In fact, Professor Jingde Cheng is not only a supervisor for my academic life, but also the most important tutor in my life. The things I learnt from him is not only limited in the academic field, but he taught me how to become a wise and good man.

I am very grateful to my thesis committee: Professor Norihiko Yoshida, Associate Professor Noriaki Yoshiura, Associate Professor Takeshi Koshiba, and Associate Professor Takashi Horiyama for their support, invaluable advice and comments to my research.

I am grateful to Assistant Professor Yuichi Goto for teaching me so much in my research and helping me in all respects. I would also like to thank other AISE lab members who have helped me in my research.

I also express my special thanks to Saitama University, and all of those teachers and staffs of the university who have helped me in my life studying abroad.

Finally, I would like to express my special thanks to my parents, who financed and support me to live and study in Japan. It would be impossible for me to pursue my doctoral degree in Japan without their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

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 [40, 41]. 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?" [40, 41]. 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 [4, 5]. 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 [4, 5].

The ATF problem is still an open problem until now [20]. Although there have been valuable works on the research of automated theorem proving (ATP for short), those works have nothing to do for the ATF problem [11]. On the other hand, a few works aimed to automated theorem discovery (ATD for short) [12] and automated theorem generation (ATG for short) [3, 13] have been done, however, the ATF problem which is to pursue properties to find new and interesting theorems is different from ATD and ATG. In fact, Wos's problem can be regarded as an attempt to find a systematic methodology in automated reasoning area, but the works on ATD and ATG are not. Those works on ATD and ATG almost aimed to one certain mathematical field. The earlier work of ATG is about a conceptual framework for discovery of theorems in geometry and a mechanism which systematically discovers such theorems [3]. After that, some works of ATD and ATG in geometry field were also done [14, 33, 35]. Besides of the geometry fields, automatic generation of classification theorems for finite algebras was performed by Colton, et al. [13]. McCasland, et al. [32] have also proposed a method for automated discovery of inductive theorems and performed some experiments in the natural numbers, positive natural numbers, and group theory. ATD in the field of game theory has been also performed in recent years [38]. Besides of the research of ATD or ATG on those certain fields, to deal with the problem about evaluation of "interesting" theorems, Puzis, et al. [34] described the techniques

which used filters and ranking to identify interesting theorems among the logical consequences. The works of ATD and ATG are ongoing, because the current works cannot find new theorems and can only rediscover some simple known theorems in some certain fields.

The approaches of current works of ATD and ATG are hard to solve the ATF problem generally [20]. First, all of the works are based on classical mathematical logic (CML for short) or its various conservative extensions. However, CML and its various conservative extensions are not suitable logic systems for underlying reasoning in ATF, because they have the well-known "implicational paradoxes" [5, 7]. Second, the works on ATD and ATG rely on generating hypotheses or conjectures as candidates of interesting theorems, and then use the method of ATP to prove them. However, those methods are not suitable for ATF. In fact, ATF is a kind of discovery which is a process to find out the unknown theorems. Therefore, we cannot have any target before we find a new theorem. The big problem of the current approaches is how to find those unknown and complex theorems which are hard to be generated as hypotheses or conjectures, because it is very difficult to automatically generate hypotheses and conjectures for those unknown and complex theorems. Third, all of the generating approaches of hypotheses or conjectures in the works depend on the specific knowledge of the certain fields. However, if we use the approaches to do ATF, then the methods of ATF are limited in one certain field, and are very hard to be modified to do ATF in other fields.

Cheng proposed a forward reasoning approach based on strong relevant logic to solve the ATF problem [4, 5]. ATF is a kind of discovery such that it cannot be solved by any automated theorem proving approach, and the reasoning is the only way to fit for ATF [4, 5]. 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 of 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 [7, 9]. On the other hand, not all logics can serve well as the fundamental logic underlying reasoning. In order to avoid the implicational paradoxes in CML or its various conservative extensions, 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" [7], and proposed strong relevant logics Tc, Ec, and Rc for relevant reasoning [7]. Tc, Ec, and Rc reject all the conjunction-implicational 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 [7]. The forward reasoning approach based on strong relevant logic is to use strong relevant

logic as the fundamental logic to underlie the reasoning in ATF.

The forward reasoning approach based on strong relevant logic is hopeful to solve the problem of ATF. First, by using strong relevant logic as the fundamental logic to underlie reasoning in ATF, we can avoid lots of boring theorems holding paradoxes deduced by using classical mathematical logic, because strong relevant logic is a paradox-free logic system [4, 5]. Second, by using Cheng's approach to do ATF, it is not necessary to generate hypotheses first and we can directly to reasoned out theorems. Third, Cheng's approach is hopeful to establish a systematic method for ATF in multi-fields, because Cheng's approach separates the empirical parts and logical parts to do ATF. In other words, Cheng's reasoning approach is independent from the specific knowledge of one certain field. Therefore, we can do ATF in different fields by using same fragments of theorems of strong relevant logics, the only difference is inputting empirical premises of different fields. To support Cheng's approach to do ATF in multi-fields, a semi-lattice model of formal theories was proposed [8], in which the formal theory of axiomatic set theory is seen as the minimum elements in the semi-lattice model, and other formal theories can be established above the axiomatic set theory level by level.

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 [24, 25]. 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 [24, 25]. To solve the ATF problem, it is necessary to establish a systematic and general methodology for ATF based on Cheng's approach.

1.2 Purpose and objectives

Our purpose is to propose a systematic methodology for ATF by forward reasoning based on strong relevant logic. The methodology holds generality so that we can use it to do ATF in various fields.

Our work involves following objectives. First, we propose the systematic methodology for ATF. Second, we show the effectiveness of our methodology by performing three case studies of ATF in three different mathematical fields [19, 21, 23]. The first field is NBG set theory, the second one is Peano's arithmetic, and the third one is graph theory. For each case study, we elaborated how to apply our methodology, showed results and gave a evaluation. After we presented three case studies, we evaluated the methodology from viewpoint of generality.

1.3 Structure of the thesis

This thesis is organized as follows. Chapter 1 presents the background and purpose of this research. Chapter 2 explains the basis of the strong relevant logic and the terminology of ATF. Chapter 3 presents our systematic methodology for ATF. Chapter 4 presents the case study of preparation of logical fragments. Chapter 5

presents the case study of ATF in NBG set theory. Chapter 6 presents the case study of ATF in Peano's arithmetic. Chapter 7 presents the case study of ATF in graph theory. Chapter 8 gives a discussion about our methodology. Finally, concluding remarks are given in Chapter 9.

Chapter 2

Basic notions and notations

2.1 Logic-based reasoning

Reasoning is the process of drawing new conclusions from some premises which are known facts and/or assumed hypotheses [9]. Therefore, reasoning can enlarging or extending somethings, or adding to what is already known or assumed. A logically valid reasoning is a reasoning such that its process of drawing new conclusions from premises is justified based on some logical criterion in order to obtain correct conclusions. Therefore, a reasoning may be valid on a logical criterion but invalid on another. The most essential and central concept in logic is the logical consequence relation that relates a given set of premises to those conclusions, which validly follow from the premises.

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 [7].

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 $Th^e_L(P) =_{df} \{A | P \vdash_L A \; and \; A \notin Th(L)\}$, $Th(L)$ and $Th^e_L(P)$ 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 [7].

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?"[5]

The notion of the degree [10] of a connective is defined as follows: Let θ be an arbitrary n-ary $(1 \leq n)$ 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 $\theta(a_1, a_2, ..., a_n)$ where $a_1, a_2, ..., a_n$ are formulas, then $D_\theta(A) = max\{D_\theta(a_1), D_\theta(a_2), ..., D_\theta(a_n)\} + 1$, (3)

if A is in the form $\sigma(a_1, a_2, ..., a_n)$ where σ is a connective different from θ and $a_1, a_2, ..., a_n$ are formulas, then $D_\theta(A) = max\{D_\theta(a_1), D_\theta(a_2), ..., D_\theta(a_n)\}$, and (4) if A is in the form QB 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 [10] is defined as follows: Let $\theta_1, \theta_2, ..., \theta_n$ be connectives of logic L and $k_1, k_2, ..., k_n$ be natural numbers, the fragment of L about $\theta_1, \theta_2, ..., \theta_n$ and their degrees $k_1, k_2, ..., k_n$, denoted by $Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$, 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 $D_{\theta_1}(A) \leq k_1, D_{\theta_2}(A) \leq k_2, ..., D_{\theta_n}(A) \leq k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$, (2) if A is the result of applying an inference rule of L to some members of $Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$ and $D_{\theta_1}(A) \leq k_1, D_{\theta_2}(A) \leq k_2, ..., D_{\theta_n}(A) \leq k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$, (3) Nothing else are in $Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$. 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 logical fragment $Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$ denoted by $T^{(\eta_1,j_1,...,\eta_s,j_s)}_{Th^{(\theta_1,k_1,...,\theta_n,k_n)}(L)}(P)$ is also similarly defined as the notion of a fragment of a logic.

2.2 Strong relevant logics

In the literature of mathematical, natural, social, and human sciences, it is probably difficult, if not impossible, to find a sentence form that is more generally used for describing various definitions, propositions, and theorems than the sentence form of 'if ... then ...'. In logic, a sentence in the form of 'if ... then ...' is usually called a conditional proposition or simply conditional which states that there exists a relation of sufficient condition between the 'if' part and the 'then' part of the sentence. Scientists always use conditionals in their descriptions of various definitions, propositions, and theorems to connect a concept, fact, situation or conclusion to its sufficient conditions. The major work of almost all scientists is to discover some sufficient condition relations between various phenomena, data, and laws in their research fields [7].

The notion abstracted from various conditionals is called "entailment". In an entailment there are two propositions which called the antecedent and the consequent. The truth and/or validity of an entailment depends not only on the truths of its antecedent and consequent but also essentially on a necessarily relevant, conditional, and/or causal relation between its antecedent and consequent.

A logical conditional that is considered to be universally true, in the sense of that logic, is also called an entailment of that logic. Indeed, the most intrinsic difference between various different logic systems is to regard what class of conditionals as entailments.

An obvious candidate for the logic to be used to underlie ATF is CML. However, CML was established in order to provide formal languages for describing the structures with which mathematicians work, and the methods of proof available to them; its principal aim is a precise and adequate understanding of the notion of mathematical proof. CML is not a suitable fundamental tool for ATF because

of the well-known "implicational paradox problem" [7].

In CML, the notion of conditional, which is intrinsically intensional but not truth-functional, is represented by the truth-functional extensional notion of material implication (denoted by $\rightarrow$ in this thesis) that is defined as $A \rightarrow B =_{df} \neg(A \wedge \neg B)$ or $A \rightarrow B =_{df} \neg A \vee B$, where $\wedge$, $\vee$, and $\neg$ denote the notion of conjunction, disjunction, and negation, respectively. However, the material implication is intrinsically different from the notion of conditional in meaning (semantics). As a result, for example, propositions "snow is white $\rightarrow$ 1+1=2", "snow is black $\rightarrow$ 1+1=2", and "snow is black $\rightarrow$ 1+1=3" are all right in logic. If we read "$\rightarrow$" as 'if... then..." then "if snow is white then 1+1=2", "if snow is black then 1+1=2", and "if snow is black then 1+1=3" are so strange in human logical thinking, because there is no necessarily relevant, conditional, and/or causal relation between the "if" part and the "then" part of each sentence [5]. Obviously, the notion of entailment in human logical thinking is intrinsically different from the notion of material implication in CML.

In order to avoid the "implicational paradox problem", traditional relevant logics T, E, and R were constructed [1, 2]. However, traditional relevant logics are also not suitable for ATF. Although traditional relevant logics have rejected the "implicational paradox problem", there are still some logical axioms or theorems in the logics which are not natural in the sense of entailment. Cheng named them "conjunction-implicational paradoxes" and "disjunction-implicational paradoxes". For example, "$(A \wedge B) \Rightarrow A$", "$A \Rightarrow (A \vee B)$", and so on [5]. Let us consider "$(A \wedge B) \Rightarrow A$", B in the antecedent may have no relevance with A in the consequent so that this formular are not natural in the sense of entailment. The propositions in this form such as "if snow is white and 1+1=2 then snow is white", "if snow is white and 1+1=3 then snow is white", "if snow is white and snow is not white then snow is white" and so on. Obviously, those propositions are not natural in the sense of entailment.

In order to avoid those paradoxes in traditional relevant logics, Cheng proposed strong relevant logics, named Tc, Ec, and Rc, for conditional relation representation and reasoning [7]. As a modification of T, E, and R, Tc, Ec and Rc reject all conjunction-implicational and disjunction-implicational paradoxes in T, E and R, respectively, so that they are free of not only implicational paradoxes in CML, but also conjunction-implicational and disjunction-implicational paradoxes in traditional relevant logics. What underlies the strong relevant logics is the strong relevance principle: If A is a theorem of Rc, Ec, or Tc, then every sentential variable in A occurs at least once as an antecedent part and at least once as a consequent part. Therefore, strong relevant logics are the most suitable candidates as the fundamental logic to underlie ATF [7]. Besides, Cheng also proposed predicate strong relevant logics, named TcQ, EcQ, and RcQ [7].

The logical connectives, axiom schemata, and inference rules of strong relevant logics are shown as follows:

Primitive logical connectives:

$\Rightarrow$: entailment

$\neg$: negation

$\wedge$: extensional conjunction

Defined logical connectives:

$\otimes$: intensional conjunction, $A \otimes B =_{df} \neg(A \Rightarrow \neg B)$

$\oplus$: intensional disjunction, $A \oplus B =_{df} \neg A \Rightarrow B$

$\Leftrightarrow$: intensional equivalence, $A \Leftrightarrow B =_{df} (A \Rightarrow B) \otimes (B \Rightarrow A)$

$\vee$: extensional disjunction, $A \vee B =_{df} \neg(\neg A \wedge \neg B)$

$\rightarrow$: material implication, $A \rightarrow B =_{df} \neg(A \wedge \neg B)$ or $\neg A \vee B$

$\leftrightarrow$: extensional equivalence, $A \leftrightarrow B =_{df} (A \rightarrow B) \wedge (B \rightarrow A)$

Quantifiers:

$\forall$: universal quantifier

$\exists$: existential quantifier

These quantifiers are not independent and can be defined as follows:

$$\begin{aligned} \forall x A &=_{df} \neg\exists x \neg A, \\ \exists x A &=_{df} \neg\forall x \neg A. \end{aligned}$$

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"': $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (((A \Rightarrow C) \Rightarrow D) \Rightarrow D))$

E5: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow (B \Rightarrow (A \Rightarrow C))$

E5': $A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$

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 $LA =_{df} (A \Rightarrow A) \Rightarrow A$

D1: $A \Rightarrow (A \vee B)$

D2: $B \Rightarrow (A \vee B)$

D3: $((A \Rightarrow C) \wedge (B \Rightarrow C)) \Rightarrow ((A \vee B) \Rightarrow C)$

DCD: $(A \wedge (B \vee C)) \Rightarrow ((A \wedge B) \vee C)$

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: $(\forall xA \wedge \forall xB) \Rightarrow \forall x(A \wedge B)$

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)

$\wedge$I: "from A and B infer $A \wedge B$" (Adjunction)

$\forall$I: "if A is an axiom, so is $\forall xA$" (Generalization of axioms)

Thus, various relevant logic systems may now defined as follows, where we use 'A | B' to denote any choice of one from two axiom schemata A and B.

Te = {E1, E2, E2', E3| E3" } + $\Rightarrow$E

Ee = {E1, E2 | E2', E3 | E3', E4 | E4'} + $\Rightarrow$E

Ee = {E2', E3, E4"} + $\Rightarrow$E

Ee = {E1, E3, E4"'} + $\Rightarrow$E

Re = {E1, E2 | E2', E3 | E3', E5 | E5'} + $\Rightarrow$E

Ten = Te + {N1, N2, N3 }

Een = Ee + {N1, N2, N3 }

Ren = Re + {N2, N3 }

T = Ten + {C1 $\sim$ C3, D1 $\sim$ D3, DCD } + $\wedge$I

E = Een + {C1 $\sim$ C4, D1 $\sim$ D3, DCD } + $\wedge$I

R = Ren + {C1 $\sim$ C3, D1 $\sim$ D3, DCD } + $\wedge$I

Tc = Ten + {C3, C5 $\sim$ C10}

$$
\begin{aligned}
\text{Ec} &= \text{Een} + \{\text{C3} \sim \text{C10}\} \\
\text{Rc} &= \text{Ren} + \{\text{C3, C5} \sim \text{C10}\} \\
\text{TeQ} &= \text{Te} + \{\text{IQ1, IQ3} \sim \text{IQ4 }\} + \forall\text{I} \\
\text{EeQ} &= \text{Ee} + \{\text{IQ1, IQ3} \sim \text{IQ5 }\} + \forall\text{I} \\
\text{ReQ} &= \text{Re} + \{\text{IQ1, IQ3} \sim \text{IQ4 }\} + \forall\text{I} \\
\text{TenQ} &= \text{Ten} + \{\text{IQ1, IQ3} \sim \text{IQ4 }\} + \forall\text{I} \\
\text{EenQ} &= \text{Een} + \{\text{IQ1, IQ3} \sim \text{IQ5 }\} + \forall\text{I} \\
\text{RenQ} &= \text{Ren} + \{\text{IQ1, IQ3} \sim \text{IQ4 }\} + \forall\text{I} \\
\text{TcQ} &= \text{Tc} + \{\text{IQ1} \sim \text{IQ4 }\} + \forall\text{I} \\
\text{EcQ} &= \text{Ec} + \{\text{IQ1} \sim \text{IQ5 }\} + \forall\text{I} \\
\text{RcQ} &= \text{Rc} + \{\text{IQ1} \sim \text{IQ4 }\} + \forall\text{I}
\end{aligned}
$$

Here, Te, Ee, and Re are the purely implicational fragments of T, E, and R, respectively. Ten, Een, and Ren are the implication-negation fragments of T, E, and R, respectively; TcQ, EcQ, and RcQ are predicate strong relevant logics proposed by Cheng [7].

2.3 Forward deduction engine

Reasoning can be classified into three forms, deduction, induction and abduction. Deduction is the process of deducing or drawing conclusions from some general principles already known or assumed. Induction is the process of inferring some general laws or principles from the observation of particular instances. Abduction is the process whereby a surprising fact is made explicable by the application to it of a suitable proposition. The deduction guarantees that the conclusions deduced in the process are true if all premises are true, but induction and abduction do not. Thus, automated forward deduction is an indispensable component for above computing systems with purposes of prediction and/or discovery. Automated forward deduction is a process of deducing new and unknown conclusions automatically by applying inference rules to premises and previously deduced conclusions repeatedly until some previously specified condition is satisfied [16]. All of backward reasoning engines, including backward deduction engines, cannot use to automatically discover new theorems [10].

Cheng has proposed an automated forward deduction system for general-purpose entailment calculus, named "EnCal" [6]. EnCal supports automated forward deduction based on strong relevant logics as well as other logics, but the logic systems and application fields which EnCal can supports is limited and EnCal cannot perform inductive and abductive reasoning. Based on the experiences of development and applications of EnCal, FreeEnCal [10, 27] has been designed and developed which can interpret and perform inference rules defined and given by its users,

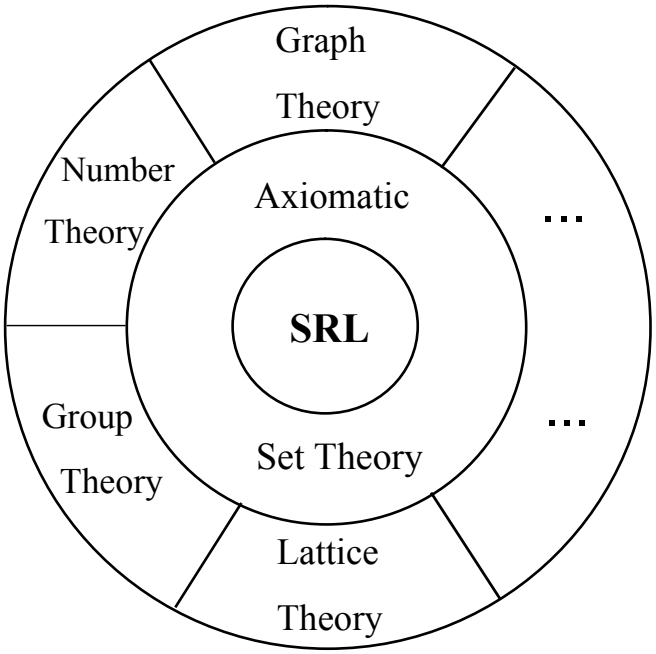

Figure 2.1: The semi-lattice model of formal theories

can deal with various logic systems including classical mathematical logic, its various classical conservative extensions, and various philosophical logics in principle [26, 30], draw empirical theorems of various formal theories constructed based on various logic systems, and perform deductive, inductive, and abductive reasoning automatically [10]. FreeEnCal [10] 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 $Th^{(\theta_1,k_1,\theta_2,k_2,...,\theta_n,k_n)}(L)$. FreeEnCal can also reason out in principle all empirical theorems of $T^{(\eta_1,j_1,...,\eta_s,j_s)}_{Th^{(\theta_1,k_1,...,\theta_n,k_n)}(L)}(P)$ from $Th^{(\theta_1,k_1,...,\theta_n,k_n)}(L)$ and P with inference rules of L.

2.4 The semi-lattice model of formal theories

Cheng has proposed a semi-lattice model of formal theories [8], which can support forward reasoning approach based on strong relevant logic to do ATF in multi-fields. The core of the model is strong relevant logic (SRL) and the formal theory of axiomatic set theory is seen as the minimum element in the semi-lattice, and other formal theories can be established above the axiomatic set theory as shown in Figure 2.1.

On the other hand, theorem finding processes in mathematics must include some concept/notion abstraction processes. In any mathematical field, definitions and axioms mean simple concepts. Mathematicians continue to define more complex concepts by using previously given definitions and axioms, and already defined

concepts. Then, mathematicians think, assume, prove propositions by using the defined complex concepts. After that, they obtain new theorems. For example, predicate "$\in$" is the most basic predicate in the set theory. Mathematicians define predicate "$\subseteq$" which is a higher level predicate than "$\in$", and abstract from "$\in$" by the definition of "$\subseteq$": $\forall x \forall y(\forall u((u \in x) \Rightarrow (u \in y)) \Leftrightarrow (x \subseteq y))$. In addition, they define predicate "$=$" which is a higher level predicate than "$\subseteq$", and abstracts from "$\subseteq$" by the axiom: $\forall x \forall y(((x \subseteq y) \wedge (y \subseteq x)) \Leftrightarrow (x = y))$. Moreover, mathematicians think, assume, and prove own beliefs by using simple representation rather than complex representation if both representation show same meaning. For example, after mathematicians defined "$=$", they will think, assume, and prove propositions by using $\forall x \forall y(x = y)$ rather than $\forall x \forall y((x \subseteq y) \wedge (y \subseteq x))$. In theorem finding process, a complex representation is replaced with a simple representation, but not preserved, after mathematicians define the simple representation as the complex representation. To do ATF by abstraction of mathematical concepts based on Cheng's semi-lattice model of formal theories holds generality for ATF. The definitions of abstraction level are shown as below.

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) $pal(X) = max(pal(Y_1), pal(Y_2), ..., pal(Y_n)) + 1$ if a predicate X is defined by other predicates $Y_1, Y_2, ..., Y_n$ in the target field where n is a natural number. A predicate X is called k-level predicate, if $pal(X) = k$. If $pal(X) < pal(Y)$, we call 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) $fal(f) = max(fal(g_1), fal(g_2), ..., fal(g_n)) + 1$ if a function f is defined by other functions $g_1, g_2, ..., g_n$ in the target field where n is a natural number. A function f is k-level function, if $fal(f) = k$. If $fal(f) < fal(g)$, we call 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) $pal(A) = max(pal(Q_1), pal(Q_2), ..., pal(Q_n))$ where Q_i is a predicate and occurs in A $(1 \leq i \leq n)$, or $pal(A) = 0$, if there is not any predicate in A, (3) $fal(A) = max(fal(g_1), fal(g_2), ..., fal(g_n))$ where g_i is a function and occurs in A $(1 \leq i \leq n)$, or $fal(A) = 0$, if there is not any function in A. A formula A is (k, m)-level formula, if $lfal(A) = (k, m)$. (k, m)-level formula is also called k-level predicate formula, or m-level function formula.

(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 number $(0 \leq j \leq k$ and $0 \leq n \leq m)$.

If two theorems are same theorems when we use the most primitive predicate and function of the target field to represent them, then we defined them as "equivalent theorems".

Chapter 3

A systematic methodology for automated theorem finding

3.1 Overview

We proposed a systematic methodology for ATF by forward reasoning based on strong relevant logic. The methodology is aimed to do ATF in various mathematical fields based on the proposed semi-lattice model of formal theories. In the target field of ATF, we defined the abstract levels of predicates and functions and the whole procedure of the methodology can be seen as reasoning, abstracting and finding theorems systematically so that we can do ATF from simple theorems to complex theorems. The methodology holds generality so that we can use it to do ATF in various fields.

3.2 Systematic methodology

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. Figure 3.1 shows the procedure flow of the systematic methodology.

Phase 1 Prepare logical fragments for various empirical theories

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

Phase 1.1 Define a semi-lattice of logical fragments

In this phase, we define a semi-lattice [15] of logical fragments of EcQ such that we can prepare logical fragments systematically [8, 11]. EcQ is a suitable logic system for reasoning in ATF, because EcQ is the logic system of strict and

relevant implication [7]. To define the semi-lattice of logical fragments of EcQ, we use the elements of the semi-lattice to represent the different logical fragments respectively, and use the partial order of the semi-lattice to represent the inclusion relation between two logical fragments [8]. Firstly, we define the minimum element of the semi-lattice. All of the logical 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 logical 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 logical fragments of EeQ in this order: first the logical fragment $Th^{(\Rightarrow,1)}(EeQ)$, then $Th^{(\Rightarrow,2)}(EeQ)$, finally the logical fragment $Th^{(\Rightarrow,3)}(EeQ)$ and so on. A defined semi-lattice of logical fragments of EcQ is shown in Figure 3.2.

Phase 1.2 Deduce logical theorems

In this phase, we deduce logic theorems according to the partial order of the defined semi-lattice of logical 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 [5, 7, 8]. We set the degree of $\Rightarrow$ below 4, and set the degree of $\neg$ and $\wedge$ below 2. We can start to deduce the logical fragments from $Th^{(\Rightarrow,1)}(EeQ)$, but not start from $Th^{(\Rightarrow,1)}(Ee)$, because ATF in all of mathematical fields need to use logical fragments of predicate strong relevant logic. In detail, first we input all the axioms of EeQ and set the degree of $\Rightarrow$ by 1 to deduce the logical 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 $\Rightarrow$ by 2 to deduce $Th^{(\Rightarrow,2)}(EeQ)$. Thirdly, we input the deduced logical theorems of $Th^{(\Rightarrow,2)}(EeQ)$ as new premises to deduce $Th^{(\Rightarrow,3)}(EeQ)$. After that, we combine the axioms of $N1 - N3$ of $EenQ$ with deduced logical theorems in the logical fragments $Th^{(\Rightarrow,k)}(EeQ)$ $(1 \leq k \leq 3)$ and set the degree of $\neg$ by 1 to deduce the logical fragments $Th^{(\Rightarrow,k,\neg,1)}(EenQ)$ $(1 \leq k \leq 3)$. Finally we combine the axioms $C3 - C10$ of EcQ with deduced logic theorems in $Th^{(\Rightarrow,k,\neg,1)}(EenQ)$ $(1 \leq k \leq 3)$ and set the degree of $\wedge$ by 1 to deduce the logical fragments $Th^{(\Rightarrow,k,\neg,1,\wedge,1)}(EcQ)$ $(1 \leq k \leq 3)$.

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 empirical 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.

Phase 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 Figure 3.1 such that we can do ATF systematically according to the partial order of defined semi-lattice of logical fragments and the partial order of defined semi-lattice of fragments of collected premises. In detail, we firstly choose the prepared minimum logical fragment to do ATF. After we chose logical 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 logical 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 of premises, we choose the bigger logical 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 logical fragments, the ATF process should be continued until we can find new and interesting theorems or we used all of the prepared logical 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 logical fragment, inference rule, and set the degree of each logic connective to deduce empirical theorems. We choose logical fragment according to the partial order of defined semi-lattice of logical 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 by substitution can deduce new empirical theorems with other deduced theorems by using modus ponens, and then we perform 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 [39]. If we succeed to prove the target, we adopt the target as an empirical theorem.

Phase 4 Abstract empirical theorems

In this phase, we perform the k-level abstraction according to the current level (k-level, $k > 0$) in the defined semi-lattice of abstract level fragments of empirical premises [22].

To perform the k-level abstraction, we define i-level abstraction rule r_i. An i-level abstraction rule r_i is defined as $r_i =< p_i, w_j >$ where p_i is i-level predicate/function, w_j is a j-level predicate/function formula, i, j, and k are natural numbers, and $(1 \leq j < i \leq k)$. we define i-level abstraction rule r_i according to definitions of predicates/functions in a certain mathematical field. For example, if an i-level predicate p is defined as $\forall x_1 \ldots \forall x_n(p(x_1, \ldots, x_n) \Leftrightarrow B)$ where B is j-level predicate formula, i, j, and n are natural numbers, and $(1 \leq j < i)$, then an i-level abstraction rule is $< p(x_1, \ldots, x_n), B >$. Then, the procedure of k-level abstraction is as follows.

1. S is a set of all i-level abstraction rules.
2. $r_i \in S$ if $S \neq \emptyset$.
3. if a theorem A includes w_j of r_i, then replace all w_j in A with p_i of r_i.
4. $S \leftarrow S - \{r_i\}$
5. Go back to 3.2.

Note that the k-level abstraction conclude a process of an explicitly epistemic contraction and a process of an explicitly epistemic expansion by predicate/function abstraction. An explicitly epistemic contraction by predicate/function abstraction is an operation to remove a theorem from the current set of obtained theorems, if the theorem can be k-level abstraction. An explicitly epistemic expansion by predicate/function abstraction is an operation to add a theorem obtained by k-level abstraction to a source theorem, if the abstracted theorem does not exist

Table 3.1: 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%

in the current set of obtained theorems. After phase 4, each theorem in the current set of obtained theorems is checked whether the explicitly epistemic contraction and expansion can be applied to the theorem or not. By the explicitly epistemic contraction and expansion, redundant equivalent theorems can be deleted in the current set of obtained 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 [25].

First, to conveniently analyze empirical theorems, we remove the quantifiers of all the empirical theorems and called 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 [25].

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 empirical theorems of each abstract level, logical connectives of interesting theorems must be low degree, because it is very hard for human's brain to understand those higher degree empirical theorems. For example, we have analyzed more than 400 known theorems of NBG set theory recorded in Quaife's book [39], and recorded our analysis results in Table 3.1. We can find that the logical connectives of those known theorems are almost lower than degree 3.

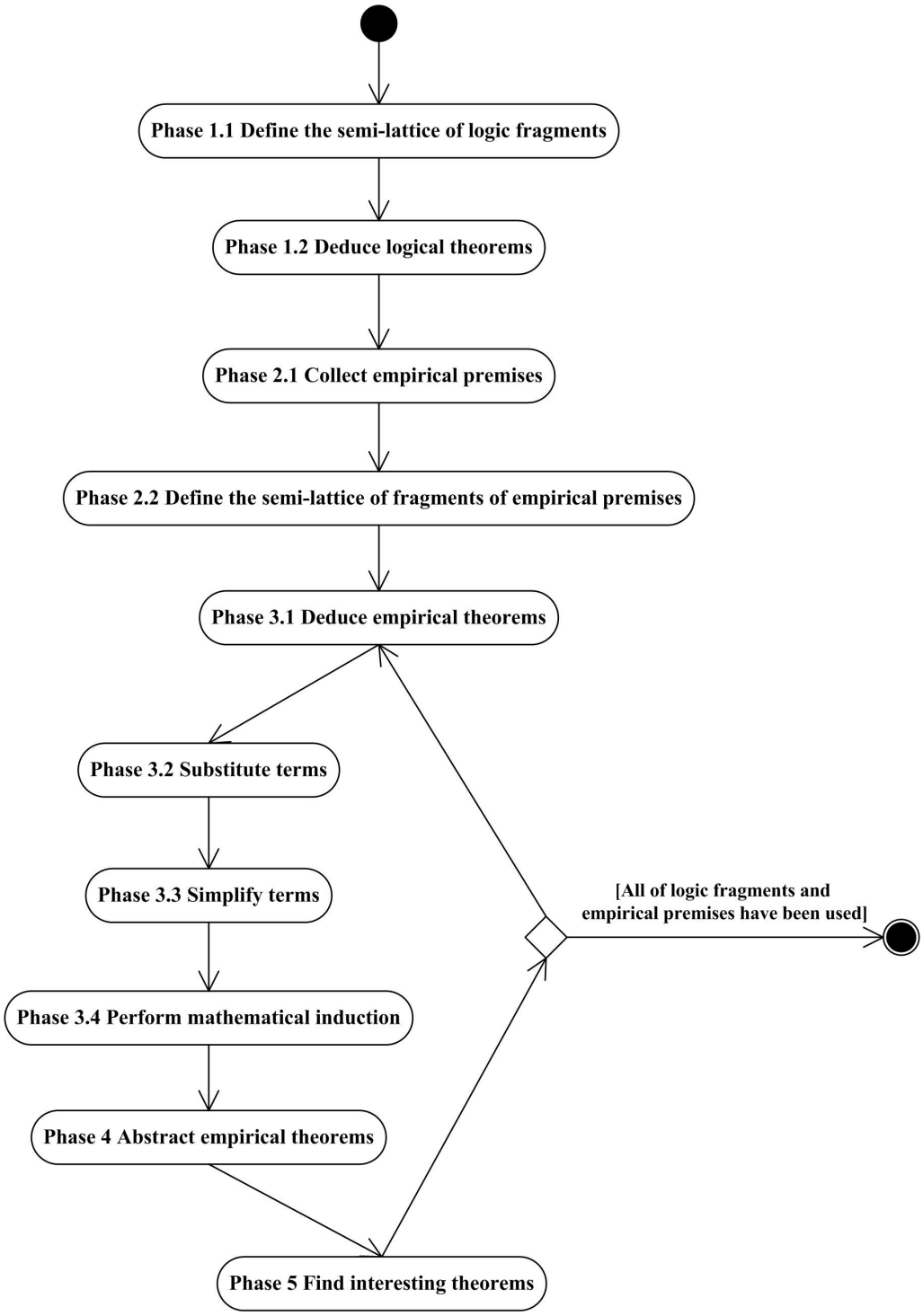

Figure 3.1: The activity diagram of the systematic method for ATF

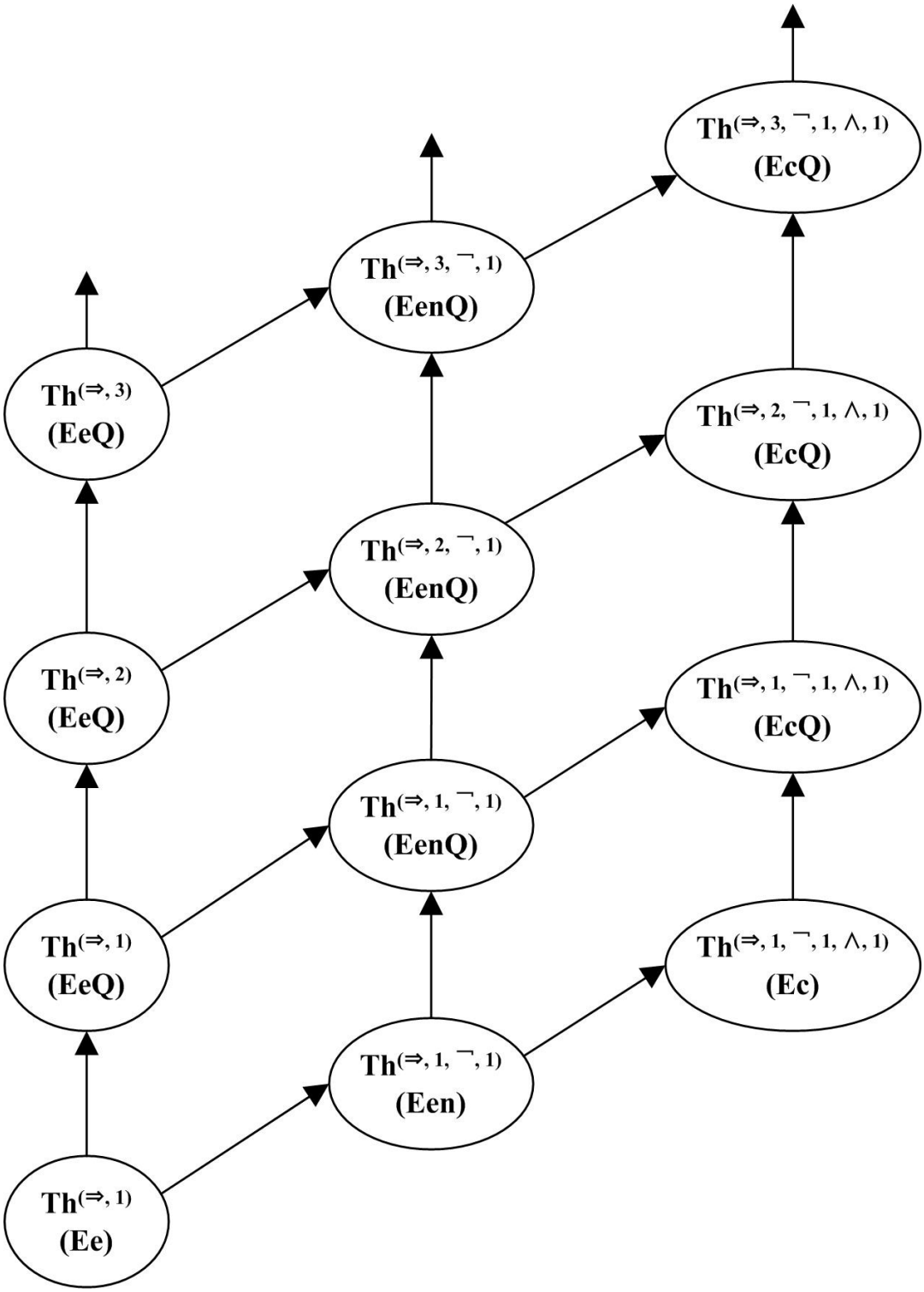

Figure 3.2: The semi-lattice of logical fragments of EcQ

Chapter 4

Preparation of logical fragments

In our case studies, we chose strong relevant logic system EcQ [7] as fundamental logic system, and we used the forward reasoning engine FreeEnCal to deduce the logical theorems automatically. The prepared logical fragments of strong relevant logics EcQ are independent from any target field and the prepared logical fragments can be reused for ATF in the three case stuides. Therefore, we show the first phase of following three case studies by one chapter.

Phase 1 Prepare logical fragments for various empirical theories

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

Phase 1.1 Define a semi-lattice of logical fragments

We defined a semi-lattice of logical fragments of EcQ to deduce the logical theorems and prepare the logical fragments systematically. We showed the defined semi-lattice in Figure 3.2.

Phase 1.2 Deduce logical theorems

According to the defined semi-lattice of logical fragments of EcQ, we used the inference rule of generalization of axioms to add one quantifier and used FreeEnCal to prepare logical fragments of EcQ automatically. We used axioms of EcQ to deduce logical 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 logical fragments, because they are used for deducing empirical theorems but not logical theorems. Table 4.1 shows the numbers of logical theorem schemata of the prepared logical fragments.

Then, we added quantifiers for those prepared logical fragments according to the empirical premises of the target fields so that we can used them to reason out empirical theorems in the empirical fields of ATF. The quantifiers of the collected empirical premises of NBG set theory and Peano's arithmetic are below 4, and the quantifiers of the collected empirical premises of graph theory are below 6, therefore

Table 4.1: Prepared logical fragments

logical fragments	Logical theorem schemata	Prefix notation	Infix notation
$Th^{(\Rightarrow,1)}(Ee)$	10	$Ee1.txt$	$Ee1\_lf.txt$
$Th^{(\Rightarrow,2)}(Ee)$	10	$Ee2.txt$	$Ee2\_lf.txt$
$Th^{(\Rightarrow,3)}(Ee)$	13	$Ee3.txt$	$Ee3\_lf.txt$
$Th^{(\Rightarrow,4)}(Ee)$	17589	$Ee4.txt$	$Ee4\_lf.txt$
$Th^{(\Rightarrow,1,\neg,1)}(Een)$	12	$Een21.txt$	$Een21\_lf.txt$
$Th^{(\Rightarrow,2,\neg,1)}(Een)$	12	$Een21.txt$	$Een21\_lf.txt$
$Th^{(\Rightarrow,3,\neg,1)}(Een)$	40	$Een31.txt$	$Een31\_lf.txt$
$Th^{(\Rightarrow,4,\neg,1)}(Een)$	1139516+	$Een41.txt$	$Een41\_lf.txt$
$Th^{(\Rightarrow,1,\neg,1,\wedge 1)}(Ec)$	18	$Ec111.txt$	$Ec111\_lf.txt$
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(Ec)$	23	$Ec211.txt$	$Ec211\_lf.txt$
$Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(Ec)$	2013	$Ec311.txt$	$Ec311\_lf.txt$
$Th^{(\Rightarrow,1)}(EeQ)$	21	$EeQ1.txt$	$EeQ1\_lf.txt$
$Th^{(\Rightarrow,2)}(EeQ)$	21	$EeQ2.txt$	$EeQ2\_lf.txt$
$Th^{(\Rightarrow,3)}(EeQ)$	64	$EeQ3.txt$	$EeQ3\_lf.txt$
$Th^{(\Rightarrow,4)}(EeQ)$	543403	$EeQ4.txt$	$EeQ4\_lf.txt$
$Th^{(\Rightarrow,1,\neg,1)}(EenQ)$	25	$EenQ21.txt$	$EenQ21\_lf.txt$
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	27	$EenQ21.txt$	$EenQ21\_lf.txt$
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	191	$EenQ31.txt$	$EenQ31\_lf.txt$
$Th^{(\Rightarrow,1,\neg,1,\wedge 1)}(EcQ)$	39	$EcQ111.txt$	$EcQ111\_lf.txt$
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	65	$EcQ211.txt$	$EcQ211\_lf.txt$
$Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$	42563	$EcQ311.txt$	$EcQ311\_lf.txt$

Table 4.2: Prepared logical fragments by adding quantifiers

Logical fragments	Logical theorem schemata below 4 quantifiers	Logical theorem schemata below 6 quantifiers
$Th^{(\Rightarrow,2)}(EeQ)$	60	91
$Th^{(\Rightarrow,3)}(EeQ)$	970	3199
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	90	147
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	3098	10714
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	311	575
$Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$	1,240,643+	1,240,643+

we added quantifiers for prepared logical fragments below 4 and 6 respectively. In the step, we did not prepare the logical fragments whose degree of $\Rightarrow$ is 1, because those logical fragments are not practical for the deducing empirical theorems. Besides, we tried to deduce $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$ after we added four quantifiers, however, we cannot deduce all of the logic theorem schemata of the logical fragment because of the current limited memory space. Table 4.2 showed the prepared logical fragments which were added quantifiers.

Chapter 5

Case study: Automated theorem finding in NBG set theory

5.1 Overview

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 almost all mathematics can be formulated in the language of set theory [28, 36], the set theory has been regarded as the ultimate proving ground for automated theorem proving programs. This is also true in ATF. Furthermore, mathematics can provide the language in which the natural sciences aspire to describe and analyze the universe. There is a natural link between mathematics and the natural sciences. Thus, if we can achieve ATF in NBG set theory, it is also hopeful that we can use the approach in other mathematical fields even other natural science fields to achieve ATF.

On the other hand, in the foundations of mathematics, NBG set theory is one kind of axiomatic set theory that is a conservative extension of the canonical axiomatic set theory ZFC [29]. But not like other axiomatic set theory, NBG set theory is one axiom system for set theory that can in turn be expressed within the language of the first-order predicate calculus, which can be programmed on a digital computer. That is the reason why we did the case study in NBG set theory rather than other axiomatic set theory.

5.2 Perform the methodology

We did ATF based on the proposed methodology phase by phase. We showed each phase how we did ATF as below.

Phase 1 Prepare logical fragments for various empirical theories

The first phase to prepare various logical fragments of strong relevant logics has been showed in the Chapter 4. The prepared logical fragments can be reused in different empirical fields for ATF, so here we did not explained this phase again.

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 [39]. 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 5.1.

- 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.
  $\forall x \forall x f(FUNCTION(xf) \wedge (x \in V) \Rightarrow ((xf\text{“}x) \in V))$.

- 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.
  $FUNCTION(choice)$.
  $\forall y((y \in V) \Rightarrow (y = 0) \vee ((choice\text{‘}y) \in y))$.

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 5.2. Second, we summarized all of the function abstract levels of the collected definitions and axioms as shown in Table 5.3. Third, we summarized all of the abstract levels of collected definitions and axioms as shown in Table 5.4. 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 Figure 5.1. In the figure, “NBG” means all of collected premises of NBG set theory.

Phase 3 Reason out empirical theorems

Table 5.1: 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

Table 5.2: 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 5.3: 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

Table 5.4: 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

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 logical fragments to deduce empirical theorems except the logical 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 logical 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 Figure 5.1. 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 combined 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 logical fragment such as $Th^{(\Rightarrow,3)}(EeQ)$ to deduce empirical theorems again. We used the logical fragments according to the defined semi-lattice shown in Figure 3.2.

Phase 3.2 Substitute terms

In the case study, after we deduced the empirical theorems, we searched all of the $\forall x \forall y ...(A \Rightarrow B)$ style empirical theorems and C style empirical theorems. For example, the theorem $\forall x \forall y((x = y) \Rightarrow (x \subseteq y))$ is a $\forall x \forall y ...(A \Rightarrow B)$ 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 $\forall x \forall y ...(A \Rightarrow B)$ 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.

Phase 4 Abstract empirical theorems

Table 5.5: ATF in NBG set theory by prepared logical fragments

Used logical fragments	Obtained empirical theorems	Core empirical theorems	Filtered results
$Th^{(\Rightarrow,2)}(EeQ)$	199	87	78
$Th^{(\Rightarrow,3)}(EeQ)$	968	128	80
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	211	91	82
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	4754	480	363
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	415	178	129

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 defined the abstraction rules according to the definitions and axioms of the fragment $P_{(k,m)}$ and performed abstraction automatically by using elimination rules provided by FreeEnCal. 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 $(x \subseteq y)$. After we used the two abstraction rules to abstract the empirical theorems, we enter into the next phase to find interesting theorems in this abstract level.

Phase 5 Find interesting theorems

In the phase, we used our filtering methods to filter uninteresting theorems step by step and then see the filtered results as candidates of interesting theorems. First, we removed the quantifiers of all of the obtained empirical theorems to get "core empirical theorems". Second, we divided all of core empirical theorems into sub-formula and used FreeEnCal to check whether or not core empirical theorems and its sub-formulas are the instances of deduced logical theorem schemata automatically. If a core empirical theorem includes a tautological sub-formula (or itself is a tautological formula), we filtered it as uninteresting theorem. In the case study, all of empirical theorems were obtained as low degree, therefore, it is not necessary to filter high degree empirical theorems. We showed the obtained empirical theorems, core empirical theorems and filtered results in Table 5.5.

5.3 Case study for explicitly epistemic contraction by predicate abstracton

Purpose of the case study is to confirm the effectiveness of explicitly epistemic contraction in Phase 4 of our methodology. By introducing explicitly epistemic contraction by predicate/function abstraction to the methodology, we can expect two positive effects, but it may be a side effect. The first positive effect is that we

Table 5.6: The abstract level of axioms and definitions in NBG set theory

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

can reduce execution time and amount of used memory space of automated forward reasoning for ATF, because by unifing those equivalent theorems, lots of intermediate results in the process of reasoning of ATF have been reduced. The second positive effect is that we can save the cost of excavation of new and interesting theorems from obtained theorems, because lots of redundant equivalent theorems have been removed, and the set of obtained theorems have been contracted. On the other hand, the explicitly epistemic contraction by predicate/function abstraction may also lead to a side effect, that is some new and interesting theorems may not be deduced due to the removed equivalent theorems. Some new and interesting theorems may be deduced from the removed equivalent theorems or the theorems deduced from the removed equivalent theorems. Therefore, we have to investigate the effectiveness of the explicitly epistemic contraction approach by a case study of ATF. Because the processes of k-level function abstraction is similar to k-level predicate abstraction in Phase 4, we only focus on the explicitly epistemic contraction by predicate abstraction in the case study.

Method

We tried to investigate two things. First is how amount of redundant results the explicitly epistemic contraction by predicate abstraction can reduce. Second is whether the explicitly epistemic contraction has a side effect or not. To investigate the two things, we did two case studies by using our methodology. One is not using the explicitly epistemic contraction by predicate abstracton, “case study 1” for short. The other one is using the explicitly epistemic contraction by predicate abstraction, “case study 2” for short. In both case studies, we used same logical fragment and empirical premises. The logic fragment we used is $Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$, which has been prepared. The empirical premises are all axioms and definitions of NBG set theory in Quaife’s book. We summarized all of the $(1, m)$-level to $(6, m)$-level definitions and axioms of NBG set theory as shown in Table 5.6.

In the both of case studies, we did 6 loops from phase 3 to phase 5 of our methodology, because of the highest abstract level of predicate is 6 in the case study. In phase 3, when we were doing i-th loop, we used the (i, m)-level fragments of the axioms and definitions, and results obtained in last loop as empirical

Table 5.7: The number of theorems included intermediate results

	Case study 1	Case study 2
6th loop	1,855	1,516

Table 5.8: The number of theorems of (i, m)-level theorems $(4 \leq i \leq 6)$

	Case study 1	Case study 2
6th loop, $(6, m)$-level theorems	20	20
5th loop, $(5, m)$-level theorems	43	37
4th loop, $(4, m)$-level theorems	21	16

premises, and obtained empirical theorems of NBG set theory by using FreeEnCal with setting the degree $\Rightarrow$ to 2, $\neg$ to 1, and $\wedge$ to 1. Note that output of FreeEnCal includes not only empirical theorems of NBG set theory, but also intermediate results that are used for reasoning the empirical theorems. In phase 4, when we were doing i-th loop, we applied i-level abstraction to each of obtained empirical theorems and intermediate results. In case study 1, both a theorem/intermediate result and a result of i-level abstraction of the theorem/intermediate result are preserved. However, in case study 2, only a result of i-level abstraction of the theorem/intermediate result is preserved because of the explicitly epistemic contraction by predicate abstraction.

In both case studies, we counted the number of obtained results (including intermediate results) after 6th loop for checking how amount of redundant results the explicitly epistemic contraction by predicate abstraction can reduce. To investigate whether the explicitly epistemic contraction has a side effect or not, we counted the number of (i, m)-level theorems obtained after i-th loop $(4 \leq i \leq 6)$. Moreover, we counted the number of each level theorems obtained after each loop in case study 1.

Result

We showed the results of the two case studies. First, we counted the number of theorems included intermediate results of the two case stuies in Table 5.7. To investigate the side effect, we counted the number of (i, m)-level $(4 \leq i \leq 6)$ theorems in i-th loop $(4 \leq i \leq 6)$ as shown in Table 5.8, and we also counted the theorems in i-th loop $(1 \leq i \leq 6)$ of case study 1 as shown in Table 5.9.

We analyzed the results from the two aspect. First one is for the two expected positive effects. The current results show that the number of obtained theorems and the intermediate results have been indeed reduced by the proposed approach as shown Table 5.7, that means the approach can reduce a lot of redundant results and reduce the cost for excavation of new and interesting theorems. Second investigation is for the side effect. The higher abstract level theorems are more possible to become new and interesting theorems. Based on the opinion, we counted the highest abstract level from 4th loop to 6th loop as shown in Table 5.8. We found

Table 5.9: The number of theorems in case study 1

	$(1,m)$	$(2,m)$	$(3,m)$	$(4,m)$	$(5,m)$	$(6,m)$
1st loop	111					
2nd loop	220	18				
3rd loop	228	19	62			
4th loop	228	57	98	21		
5th loop	228	57	98	41	43	
6th loop	228	57	98	41	43	20

almost all of the highest abstract level theroem in each loop were not removed by our approach, such as the highest abstract level theorems in 6th loop. Although some theorems cannot be obtained, some of these theorems are removed as equivalent theorems, so it is not side effect. For example, the six highest level theorems cannot be obtained in 5th loop in the case study 2, but four theorems are removed as equivalent theorems. Some theorems cannot be deduced due to the removed equivalent theorems, however, we consider that the possibility of those theorems found as new and interesting theorems is low, because the removed theorems cannot successively impact on the deduction of the higher abstract level theorems as shown in Table 5.9. Therefore, the experiment results shows our approach is effectiveness.

5.4 Evaluation

The results of the case study show that our methodology is effective. First, from the viewpoint of systematic method, it is sure that the case study of ATF in NBG set theory were performed systematically in each phase. Besides, the results of case study for explicitly epistemic contraction by predicate abstraction showed the effectiveness of the abstraction approach in our methodology. We can conclude that the approach is useful for ATF by forward reasoning in many mathematical fields. From the viewpoint of finding new and interesting theorems, our method provides the filtering method to filter most of uninteresting theorems based on syntax, especially more empirical theorems are obtained the effectiveness of the filtering method is more obvious, for example, near 90% empirical theorems reasoned by $Th^{(\Rightarrow,3)}(EeQ)$ and $Th^{(\Rightarrow,3,\neg,1)}(EenQ)$ can be removed automatically such that the scientists can find interesting theorems from the filtered results based on semantics by accepatable time.

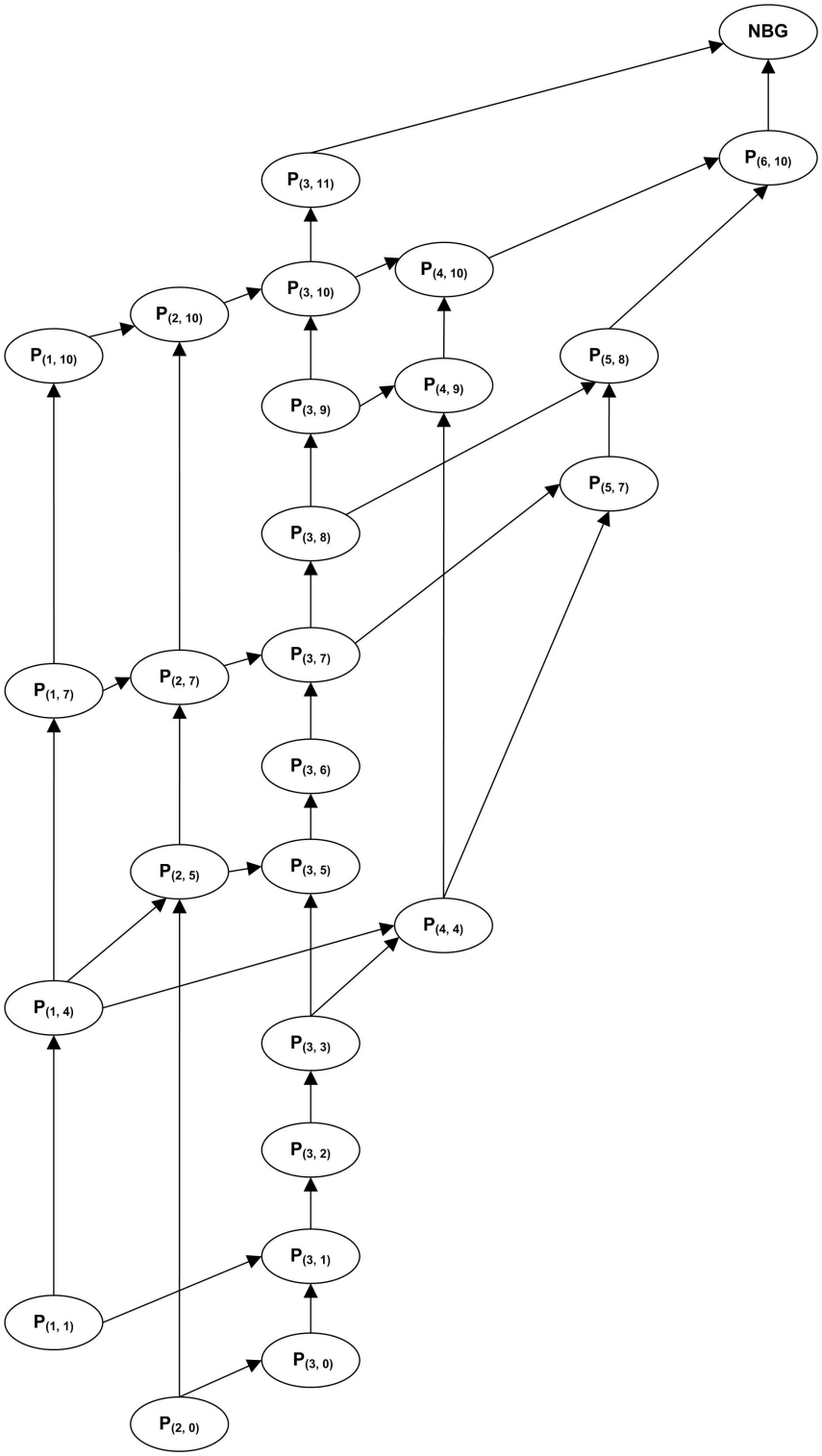

Figure 5.1: The semi-lattice of fragments of premises in NBG set theory

Chapter 6

Case study: Automated theorem finding in Peano's arithmetic

6.1 Overview

The purpose of this case study is to confirm the effectiveness and 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 [31, 39] which are a set of 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.

6.2 Perform the methodology

To confirm the generality of the proposed methodology, we performed ATF in Peano's arithmetic by using the proposed methodology as same as the first case study. We showed in each phase how we did ATF as below.

Phase 1 Prepare logical fragments for various empirical theories

The first phase to prepare various logical fragments of strong relevant logics has been showed in the Chapter 4. The prepared logical fragments can be reused in different empirical fields for ATF, so here we did not explained this phase again.

Phase 2 Prepare empirical premises

Phase 2.1 Collect empirical premises

Quaife recorded the axioms, definitions and more than 1200 known theorems in his book [39]. 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.
  $\forall x \forall y(((1 + x) = (1 + y)) \Rightarrow (x = y))$

Table 6.1: 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

- 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 Table 6.1 and Table 6.2.

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

Table 6.2: 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
[|]	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

Table 6.3: Predicate abstract level in Peano's arithmetic

Predicate	Abstract from	Level
$=$	none	1
$<$	$=$	2
DIV	$=$	2
LD	$=$	2
SET	$=$	2
PERM	$=$	2
On	$=$	2
$\equiv$	$=$	2
SORTED	$<$, $=$	3
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

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 6.3. Second, we summarized all of the function abstract levels of the collected definitions and axioms as shown in Table 6.4. Third, we summarized all of the abstract levels of collected definitions and axioms as shown in Table 6.5. 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 Figure 6.1. 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. As same as the first case study, because of the limited memory space, we also used all of the prepared logical fragments except the logical fragment $Th^{(\Rightarrow,3,\neg,1,\wedge 1)}(EcQ)$ to deduce empirical theorems. The order of using logical fragments was according to the partial order of the defined semi-lattice of logical fragments shown in Figure

Table 6.4: 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 6.5: 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

3.2, that is the same as the case study of NBG set theory. After we chose one logical fragment, we inputted the empirical premises according to the partial order of the defined semi-lattice of fragments of collected premises shown in Figure 6.1. In detail, first we inputted the axiom 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 combined the reasoned out theorems in the last loop with the definitions of $-$, $mod()$, /, !, x^y to deduce empirical theorems whose abstract level 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, 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. 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 $\forall xA$ where A is a formula. Then we tried to find $B[x/0]$ and $B[x/s(0)]$, and if we can find them in the deduced empirical theorems, we used mathematical induction to generate a hypothesis $\forall xB(x)$ 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 $(s(0) + 0) = s(0 + 0)$. 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 $(s(0) + 0) = s(0)$ was found. Because we found the theorem$(0 + 0) = 0$ (can be seen as $B[x/0]$) and the theorem $(s(0) + 0) = s(0)$ (can be seen as $B[x/s(0)]$), 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 [39], so we did not prove those known theorems again.

Phase 4 Abstract empirical theorems

As 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 defined the abstraction rules according to the axioms and definitions of the $P_{(k,m)}$, and performed abstraction automatically by using elimination rules provided by FreeEnCal. In this case study, we have defined and used 54 abstraction rules to abstract empirical theorems. 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 $((y \ mod \ x) = 0)$ to $DIV(x, y)$ according to the definition of DIV. After we used the abstraction rule to abstract the empirical theorems, we entered into the next phase to find interesting theorems in this abstract level.

Phase 5 Find interesting theorems

In the phase, we used our filtering methods to filter uninteresting theorems step by step and then see the filtered results as candidates of interesting theorems. The processes of the phase is same as the case study of NBG set theory

Table 6.6: ATF in Peano's arithmetic by prepared logical fragments

Used logical fragments	Obtained empirical theorems	Core empirical theorems	Filtered results
$Th^{(\Rightarrow,2)}(EeQ)$	290	145	129
$Th^{(\Rightarrow,3)}(EeQ)$	1216	224	139
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	328	168	152
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	4662	821	553
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	488	214	186

such that we do not elaborate them again. We showed the obtained empirical theorems, core empirical theorems and filtered results in Table 6.6. By using our methodology, some proved theorems by using OTTER indeed can be found by our methodology, such as the theorem $\forall x((x-x)=0)$. Besides, some known theorems were found as proved targets by using our methodology, such as $\forall x((x+0)=x)$ and $\forall x(min(x,x)=x)$, which were proved by using OTTER. In the filtered results, some interesing candidates found by our forward reasoning approach was not recorded in Quaife's book, such as $\forall x(\neg(1 < \text{lf(x)}) \Rightarrow (lf(x)=x))$, and to evaluate the interestingness for those empirical theorems is not a easy work, so our method is to provide those candidates to mathematicians and make mathematicians to evaluate the interestingness for those empirical theorems found by our filtering methods.

6.3 Evaluation

The case study shows that our methodology is effective and holds generality, because we used the same logical fragments used in the case study of NBG set theory and perform same methodology, we can also find some interesting candidates of empirical theorems, even some known theorems. Besides, the result of the case study shows that bigger logical fragment is used, the possibility to deduce interesting theorems is bigger, such as the empirical theorem $\forall x(\neg(1 < \text{lf(x)}) \Rightarrow (lf(x)=x))$ which was reasoned out by using logical fragment $Th^{(\Rightarrow,3,\neg,1)}(EenQ)$, but it cannot be reasoned out by other four lower degree logical fragments in the case study. Because our methodology provides a continuous and systematic way to prepare the logical 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.

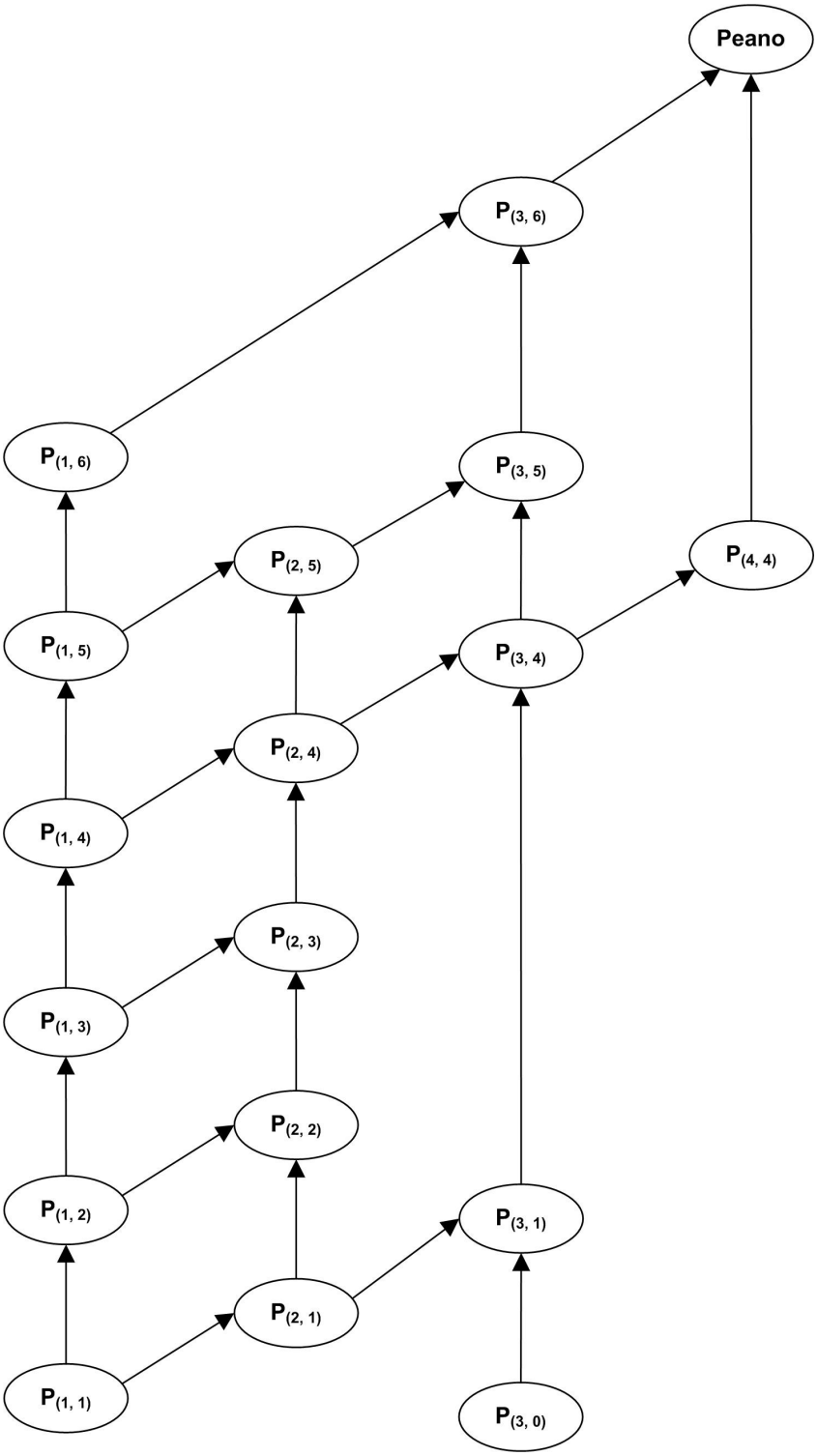

Figure 6.1: The semi-lattice of fragments of premises in Peano's arithmetic

Chapter 7

Case study: Automated theorem finding in graph theory

7.1 Overview

The purpose of this case study is to confirm the effectiveness and generality of our methodology. After we did a case study of NBG set theory and Peano's arithmetic, we also use our methodology in another field to confirm the generality. We chose graph theory as the third target field of ATF, because graph theory [17] can be established above the axiomatic set theory such that we can confirm how to perform ATF based on the semi-lattice model of formal theories. If we can also perform ATF well in graph theory by using the methodology, it means that we can also do ATF in other mathematical fields by using our methodology, because almost all of mathematical fields can be established above axiomatic set theory. Besides, graph theory can be used to model many types of relations and processes in physical, biological, social and information systems and many practical problems can be represented by graphs theory. Therefore, if the ATF in graph theory is succeeded, it will attract lots of researchers' attention and the ATF by computer program will be a new trend in other fields.

7.2 Perform the methodology

To confirm the generality of the proposed methodology, we performed ATF in graph theory by using the proposed methodology. The only difference between this case study and the last case study in Peano's arithmetic is that we also used the empirical theorems of NBG set theory as empirical premises in this case study, because the graph theory is established above axiomatic set theory. We showed in each phase how we did ATF in graph theory as below.

Phase 1 Prepare logical fragments for various empirical theories

The first phase to prepare various logical fragments of strong relevant logics has been showed in the Chapter 4. The prepared logical fragments can be reused in different empirical fields for ATF, so here we did not explained this phase again.

Phase 2 Prepare empirical premises

Phase 2.1 Collect empirical premises

The second phase is to prepare empirical premises of graph theory. Diestel recorded definitions of graph theory in his book [17]. In the case study, we chose more than 20 definitions in Diestel's book as empirical premises of the case study and formalized them by using predicates and functions in the first case study of NBG set theory. The definitions of graph theory formalized by us are shown as below.

- Definition of graph
  $G(V, E) =< V, E >$
- Definition of empty graph
  $0 = G(0, 0)$
- Definition of incident edge
  $\forall x \forall y \forall v((\{x, y\} \in E) \wedge (v \in \{x, y\}) \Leftrightarrow (\{x, y\} = incidentedge(v)))$
- Definition of loop
  $\forall x(loop(x) = \{x\})$
- Definition of isomorphism
  $\forall w \forall v \forall v' \forall e \forall e'((G(v, e) = G(v', e')) \Leftrightarrow (w \in e \Rightarrow isomorphism(w) \in e')$
- Definition of $\cap_g$
  $\forall x \forall x' \forall y \forall y'(G(x, y) \cap_g G(x', y') = G(x \cap x', y \cap y'))$
- Definition of $\cup_g$
  $\forall x \forall x' \forall y \forall y'(G(x, y) \cup_g G(x', y') = G(x \cup x', y \cup y'))$
- Definition of subgraph
  $\forall x \forall x' \forall y \forall y'(Sub(G(x, y), G'(x', y')) \Leftrightarrow ((x \subseteq x') \wedge (y \subseteq y') \wedge (x' \subseteq V) \wedge (y' \subseteq E)))$
- Definition of induced subgraph
  $\forall v \forall x \forall x' \forall y \forall y'(InducedSub(G(x, y), G'(x', y')) \Leftrightarrow Sub(G(x, y), G'(x', y')) \wedge ((v \in (x \cap x')) \Rightarrow (incidentedge(v) \in y)))$
- Definition of super graph
  $\forall x \forall x' \forall y \forall y'(Sup(G'(x', y'), G(x, y)) \Leftrightarrow Sub(G(x, y), G'(x', y'))$
- Definition of simple graph
  $\forall x \forall m \forall n \forall v \forall e(SimpleGraph(G(v, e)) \Leftrightarrow (v \subseteq V) \wedge (e \subseteq E) \wedge ((x \in v) \Rightarrow \neg(loop(x) \in e)) \wedge ((m \in e) \wedge (n \in e) \Rightarrow \neg(m = n)))$
- Definition of adjacent
  $\forall x \forall y(Adjacent(x, y) \Leftrightarrow (\{x, y\} \in E))$

- Definition of complete graph
  $\forall x \forall y \forall v \forall e(CompleteGraph(G(v,e)) \Leftrightarrow SimpleGraph(G(v,e)) \wedge ((x \in v) \wedge (y \in v) \Rightarrow Adjacent(x,y)))$

- Definition of disjoint
  $\forall x \forall x' \forall y \forall y'(Disjoint(G(x,y), G(x',y')) \Leftrightarrow (G(x,y) \cap_g G(x',y') = 0))$

- Definition of $-$
  $\forall x \forall y \forall u(G(x,y) - u = G(\sim (x \cap u), \sim (y \cap incidentedge(u)))$

- Definition of connected graph
  $\forall u \forall x \forall y \forall z \forall v \forall e(ConnectedGraph(G(v,e)) \Leftrightarrow ((G(u,x) \cup_g G(y,z) = G(v,e)) \Rightarrow \neg Disjoint(G(u,x), G(y,z)))$

- Definition of path
  $\forall e \forall v \forall x \forall m \forall n \forall p((path(v,e) = G(v,e) \Leftrightarrow (ConnectedGraph(G(v,e)) \wedge ((m \in e) \wedge (n \in e) \wedge (p \in e) \wedge \neg(m = n) \wedge \neg(n = p) \wedge \neg(m = p) \wedge (x \in m) \wedge (x \in n)) \Rightarrow \neg(x \in p)))$

- Definition of cycle
  $\forall v \forall e \forall x \forall m \exists n((cycle(v,e) = G(v,e)) \Leftrightarrow ((path(v,e) = G(v,e)) \wedge ((m \in e) \wedge (x \in m) \Rightarrow (x \in n) \wedge (n \in e) \wedge \neg(m = n)))$

- Definition of connectivity
  $\forall x \forall y \forall e \forall v(Connect(x,y) \Leftrightarrow ((x \in v) \wedge (y \in v) \Rightarrow (\{v\} \in path(v,e))))$

- Definition of forest
  $\forall v \forall v' \forall e \forall e'(Forest(G(v,e)) \Leftrightarrow \neg Sub(cycle(v',e'), G(v,e)))$

- Definition of tree
  $\forall v \forall e(Tree(G(v,e)) \Leftrightarrow ConnectedGraph(G(v,e)) \wedge Forest(G(v,e)))$

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

Then, we defined a semi-lattice of abstract level fragments of formalized empirical premises according to the proposed methodology. In detail, first we summarized all of the predicate abstract levels of the formalized definitions of graph theory as shown in Table 7.1. Second, we summarized all of the function abstract levels of the formalized definitions of graph theory as shown in Table 7.2. Third, we summarized all of the abstract levels of formalized definitions of graph theory as shown in Table 7.3. Finally, according to the abstract level of formalized definitions, we defined the semi-lattice of abstract level fragments of empirical premises in graph theory as shown in Figure 7.1. In the figure, "NBG" means all of empirical premises of NBG set theory and "Graph" means all of the collected premises of graph theory. In this case study NBG set theory is seen as the minimum element of the semi-lattice, which is different from the last two case studies, because graph theory is established above axiomatic set theory.

Phase 3 Reason out empirical theorems

Table 7.1: Predicate abstract level in graph theory

Predicate	Abstract from	Level
Adjacent	$\in$	2
Connect	$\in$	2
Sub	$\subseteq$	3
Forest	Sub	4
InducedSub	Sub, $\in$	4
Sup	Sub	4
SimpleGraph	$\in$, $\subseteq$, $=$	4
Disjoint	$=$	4
CompleteGraph	SimpleGraph, $\in$, Adjacent	5
ConnectedGraph	Disjoint, $=$	5
Tree	ConnectedGraph, Forest	6

Table 7.2: Function abstract level in graph theory

Function	Abstract from	Level
incidentedge	unordered pair	2
loop	singleton	3
G	ordered pair	4
path	G	5
$\cap_g$	$\cap$, G	5
$\cup_g$	$\cup$, G	5
isomorphism	G	5
$-$	G, $\cap$, incidentedge, $\sim$	5
cycle	G, path	6

Table 7.3: The abstract level of definitions in graph theory

Abstract levels	Definitions
(2, 1)	Definition of adjacent
(2, 5)	Definition of connectivity
(3, 2)	Definition of incident edge
(3, 3)	Definition of loop
(3, 4)	Definition of graph, empty graph, subgraph
(3, 5)	Definition of isomorphism, $\cap_g$, $\cup_g$, $-$
(3, 6)	Definition of cycle
(4, 4)	Definition of super graph, simple graph, induced subgraph
(4, 5)	Definition of disjoint
(4, 6)	Definition of forest
(5, 4)	Definition of complete graph
(5, 5)	Definition of path, connected graph
(6, 4)	Definition of tree

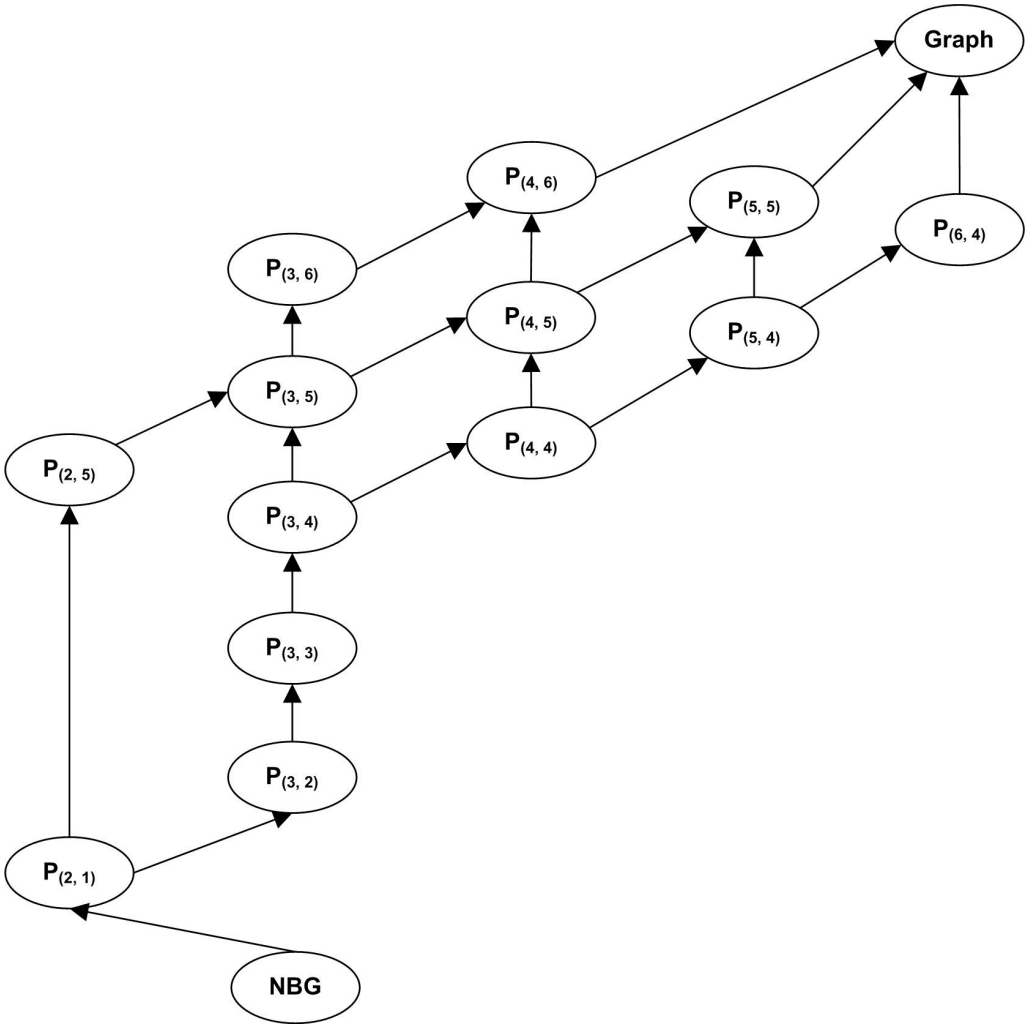

Figure 7.1: The semi-lattice of fragments of premises in graph theory

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

Phase 3.1 Deduce empirical theorems

We used the reasoning engine FreeEnCal to deduce empirical theorems of graph theory automatically. We set the degrees of $\Rightarrow$ to 2, $\neg$ to 1, and $\wedge$ to 1 to deduce the empirical theorems of graph theory. We used all of the prepared logical fragments to deduce empirical theorems except the logical 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 logical 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 Figure 7.1. In detail, we first inputted all of obtained empirical theorems of NBG set theory and the definition of adjacent to deduce the empirical theorems. Then, we came into the next phase to substitute and simplify terms and to abstract the empirical theorems to the current abstract level and find

interesting empirical theorems. After that, we came back to this phase and combined the obtained empirical theorems of graph theory which are $(2,1)$ abstract level with the the definition of incident edge to deduce the empirical theorems of graph theory which are $(3,2)$ abstract level. We used the method to deduce empirical theorems again and again according to the defined semi-lattice of fragments of empirical premises of graph theory. After all of empirical premises of graph theory have been used, we used the bigger logical fragment such as $Th^{(\Rightarrow,3)}(EeQ)$ to deduce empirical theorems again. We used the logical fragments from small to big systematically according to the defined semi-lattice shown in Figure 3.2.

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.

Phase 3.3 Simplify terms

In the case study, we did not found any $A = B$ style empirical theorems of graph theory. Therefore, we did not simplify terms for any empirical theorem in the case study.

Phase 4 Abstract empirical theorems

As same as the case study of NBG set theory and Peano's arithmetic, 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 defined the abstraction rules according to the axioms and definitions of the $P_{(k,m)}$, and performed abstraction automatically by using elimination rules provided by FreeEnCal. For example, after we reasoned out all of the theorems by inputting the fragment $P_{(2,1)}$ of premises, we changed all the sub-formula $(\{x,y\} \in E)$ to $adjacent(x,y)$ according to the definition of *adjacent*. After we used the abstraction rule to abstract the empirical theorems, we entered into the next phase to find interesting theorems in this abstract level.

Phase 5 Find interesting theorems

In the phase, we used our filtering methods to filter uninteresting theorems step by step and then see the filtered results as candidates of interesting theorems. The processes of the phase is same as the case study of NBG set theory such that we do not elaborate them again. We showed the obtained empirical theorems, core empirical theorems and filtered results in Table 7.4.

Table 7.4: ATF in graph theory by prepared logical fragments

Used logical fragments	Obtained empirical theorems	Core empirical theorems	Filtered results
$Th^{(\Rightarrow,2)}(EeQ)$	1138	102	83
$Th^{(\Rightarrow,3)}(EeQ)$	1662	133	93
$Th^{(\Rightarrow,2,\neg,1)}(EenQ)$	1139	103	84
$Th^{(\Rightarrow,3,\neg,1)}(EenQ)$	2885	288	216
$Th^{(\Rightarrow,2,\neg,1,\wedge 1)}(EcQ)$	1216	122	97

7.3 Evaluation

The case study shows that our methodology is effective and holds generality. From the viewpoint of effectiveness, it is sure that the case study of ATF in graph theory were performed systematically in each phase. All of those empirical theorems are obtained by using forward reasoning method and our filtering method can filter most of uninteresting theorems based on syntax, more than 90% empirical theorems reasoned out by all of the five prepared logical fragments can be removed as uninteresting theorems automatically such that the scientists can find interesting theorems from the filtered results based on semantics by accepatable time. From the viewpoint of generality, the case study shows that our methodology support Cheng's semi-lattice model of formal theories, that is we define the (k, m)-fragment of empirical premises of axiomatic set theory and we extend them to other mathematical fields. We consider finding process of theorems must include some concept abstraction processes in other mathematical fields and almost all of mathematical fields can be represented by axiomatic set theory, so we can conclude that our methodology is suitable for ATF in other mathematical fields, such as group theory, lattice theory and number theory.

Chapter 8

Discussion

The three case studies shows that our methodology is effective. First, from the viewpoint of systematic method, it is sure that we systematically performed three case studies of ATF in each phase by using our methodology. We systematically prepared and used logical fragments of strong relevant logic from small to big by defined semi-lattice of logical fragments, systematically reasoned out empirical theorems from simple to complex by defined semi-lattice of fragments of empirical premises, and systematically filtered uninteresting theorems step by step. Second, from the viewpoint of soundness of our methodology, indeed there is not any paradoxical theorem obtained by using our methodology. Because our methodology choose strong relevant logic as fundamental logic tool to underlie the forward reasoning.

Based on the performed three case studies, we can conclude our methodology is a general methodology for ATF in mathematical fields. In three performed case studies, we prepared logical fragments in one time and then we can performed ATF in three different mathematical fields by using those logical fragments. Besides, we defined the semi-lattice of fragments of empirical premises based on abstract level in three different mathematical fields well, such as the defined independent semi-lattice in the Peano's arithmetic and the defined semi-lattice in graph theory which extends the defined semi-lattice of NBG set theory. Because almost all of mathematical fields can be represented by axiomatic set theory and have abstraction processes of mathematical concepts, we can extend the defined semi-lattice of NBG set theory into other mathematical fields like the method used in our third case study. Finally, the performed three case studies showed that our filtering method can filter uninteresting theorems generally in three different mathematical fields, and we can also conclude that the filtering methods is useful in other mathematical fields.

On the other hand, the ultimate goal of ATF is to find new and interesting theorems, but we did not find new and interesting theorems in three case studies. 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.

To find new theorems, we have to get more complex theorems than the theorems reasoned out in our case studies. A complex theorem is a theorem whose degree of nested logical connectives and/or functions are 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 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 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 scientists choose 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 [7] and its language EPLAS [18, 37] 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. Figure 8.1 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.

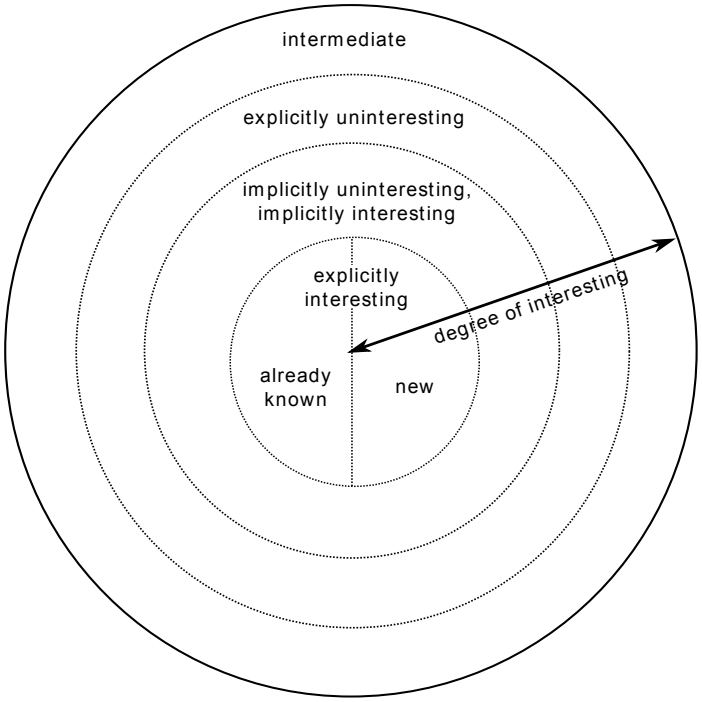

Figure 8.1: Excavation problem of ATF

- 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?

Chapter 9

Conclusions

9.1 Contributions

This work has following contributions. The first contribution is that we proposed a systematic methodology for automated theorem finding based on the semi-lattice of formal theories in which the core is strong relevant logic, and the minimum element is axiomatic set theory, above it other formal theories can be established like number theory, graph theory, and lattice theory, so the methodology holds generality for various mathematical fields. The second contribution is that we proposed a method to do automated theorem finding based on the abstraction process of mathematical concept such that we can systematically find theorems from simple theorems to complex theorems. The third contribution is that we proposed a method to generate hypothesis by using forward reasoning approach by strong relevant logic and then combine automated theorem proving approach to systematically find those theorems proved by mathematical induction. The fourth contribution is that we performed three case studies of automated theorem finding in three different mathematical fields by using our methodology and clearly showed our method and results. Before our works, it is only in theory to use forward reasoning approach based on strong relevant logic to perform automated theorem finding in different mathematical fields, but our works showed the detail and systematic procedure of automated theorem finding clearly.

9.2 Future works

There are many interesting and challenging research problems in our future works. First, to find more interesting theorems, we will deduce higher degree logical fragments and then deduce higher degree empirical theorems. Second, we will follow Cheng's epistemic programming approach [7] and use EPLAS [18, 37] 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 [8, 11] proposed by Cheng is aimed to do ATF in multi-fields, in our case studies we have perform ATF in the axiomatic set theory, number theory and graph theory. We will use our methodology with the semi-lattice of formal

theories to do ATF systematically in more fields like combinatorics, lattice theory, and group theory in the future.

Publications

Refereed papers published in journals or books (first author)

- Hongbiao Gao, Kai Shi, Yuichi Goto, and Jingde Cheng, "Finding Theorems in NBG Set Theory by Automated Forward Deduction Based on Strong Relevant Logic," D.-Z. Du and G. Zhang (Eds.), "Computing and Combinatorics, The 19th Annual International Conference, COCOON 2013, Hangzhou, China, June 21-23, 2013, Proceedings," Lecture Notes in Computer Science, Vol. 7936, pp 697-704, Springer, Heidelberg, June, 2013.
- Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "Research on Automated Theorem Finding: Current State and Future Directions," J. J. Park, Y. Pan, C. Kim, and Y. Yan (Eds.), "Future Information Technology, The 9th FTRA International Conference, FutureTech 2014, Zhangjiajie, China, May 28-31, 2014, Proceedings," Lecture Notes in Electrical Engineering, Vol. 309, pp 105-110, Springer, Heidelberg, May, 2014.
- Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "A Systematic Methodology for Automated Theorem Finding," Theoretical Computer Science 554, pp 2-21, Elsevier, October, 2014.
- Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "Explicitly Epistemic Contraction by Predicate Abstraction in Automated Theorem Finding: A Case Study in NBG Set Theory," N. T. Nguyen, et al. (Eds.), "Intelligent Information and Database Systems, The 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part I," Lecture Notes in Artificial Intelligence, Vol. 9011, pp. 593-602, Springer, March, 2015.
- Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "Automated Theorem Finding by Forward Reasoning Based on Strong Relevant Logic: A Case Study in Graph Theory," "Future Information Technology, The 10th FTRA International Conference, FutureTech 2015, Hanoi, Vietnam, May 18-20, 2015, Proceedings," Lecture Notes in Electrical Engineering, Springer, Heidelberg, May, 2015. (to appear)

Refereed papers published in international conference proceedings (first author)

- Hongbiao Gao, Kai Shi, Yuichi Goto, and Jingde Cheng, "Automated Theorem Finding by Forward Deduction Based on Strong Relevant Logic: A Case Study in NBG Set Theory," Proceedings of the 11th International Conference on Machine Learning and Cybernetics, ICMLC 2012, Xian, China, pp 1859-1865, The IEEE Systems, Man, and Cybernetics Society, July, 2012.
- Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "Automated Theorem Finding by Forward Deduction Based on The Semi-lattice Model of Formal Theory: A Case Study in NBG Set Theory," Proceedings of the 9th International Conference on Semantics, Knowledge and Grid, SKG 2013, Beijing, China, pp 22 - 29, IEEE Computer Society Press, October, 2013. (Acceptance rate: less than 30%)

Refereed papers published in journals or books (co-author)

- Shunsuke Nanaumi, Kazunori Wagatsuma, Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "A Bidirectional Transformation Supporting Tool for Formalization with Logical Formulas," N. T. Nguyen, et al. (Eds.), "Intelligent Information and Database Systems, The 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part I," Lecture Notes in Artificial Intelligence, Vol. 9011, pp. 634-643, Springer, March, 2015.

- Shunsuke Nanaumi, Kazunori Wagatsuma, Hongbiao Gao, Yuichi Goto, and Jingde Cheng, "Development of a Bidirectional Transformation Supporting Tool for Formalization with Logical Formulas and Its Application," "Future Information Technology, The 10th FTRA International Conference, FutureTech 2015, Hanoi, Vietnam, May 18-20, 2015, Proceedings," Lecture Notes in Electrical Engineering, Springer, Heidelberg, May, 2015. (to appear)

Refereed papers published in international conference proceedings (co-author)

- Yuichi Goto, Hongbiao Gao, Takahiro Tsuji, and Jingde Cheng, "Practical Usage of FreeEnCal: an Automated Forward Reasoning Engine for General-Purpose," Proceedings of the 11th International Conference on Machine Learning and Cybernetics, ICMLC 2012, Xian, China, pp. 1878-1883, The IEEE Systems, Man, and Cybernetics Society, July, 2012.

Bibliography

[1] A. R. Anderson and 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, and J. M. Dunn: Entailment: The Logic of Relevance and Necessity. vol. 2, Princeton University Press, 1992.

[3] R. Bagai, V. Shanbhogue, J. M. Zytkow, and S. C. Chou: Automatic Theorem Generation in Plane Geometry. Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Computer Science, vol. 689, Springer, Heidelberg, pp. 415-424, 1993.

[4] J. Cheng: A Relevant Logic Approach to Automated Theorem Finding. Proceedings of the Workshop on Automated Theorem Proving attached to International Symposium on Fifth Generation Computer Systems, pp. 8-15, 1994.

[5] J. Cheng: Entailment Calculus as the Logical Basis of Automated Theorem Finding in Scientific Discovery. Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium, AAAI Press-American Association for Artificial Intelligence, pp. 105-110, 1995.

[6] J. Cheng: EnCal: An Automated Forward Deduction System for General-Purpose Entailment Calculus. Advanced IT Tools, Proceedings of the 14th WCC, Canberra, Chapman & Hall, pp. 507-517, 1996.

[7] J. Cheng: A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery. Information Modelling and Knowledge Bases XI, Frontiers in Artificial Intelligence and Applications, vol. 61, IOS Press, pp. 136-159, 2000.

[8] J. Cheng: A Semilattice Model for the Theory Grid. Proceedings of the 3rd International Conference on Semantics, Knowledge and Grid, IEEE Computer Society, pp. 152-157, 2007.

[9] J. Cheng: Representing, Managing, and Reasoning about Mathematical Knowledge Based on Strong Relevant Logic. Proceedings of the 7th International Conference on Machine Learning and Cybernetics, the IEEE Systems, Man, and Cybernetics Society, pp. 299-306, July 2008.

[10] J. Cheng, S. Nara, and Y. Goto: FreeEnCal: A Forward Reasoning Engine with General-Purpose. Proceedings of the 11th International Conference on

Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, vol. 4693, Springer, Heidelberg, pp. 444-452, 2007.

[11] J. Cheng, S. Nara, Y. Goto, and T. Koh: A Cooperative Grid Computing Approach to Automated Theorem Finding and Automated Problem Proposing. Proceedings of the 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, vol. 4693, Springer, Heidelberg, pp. 840-851, 2007.

[12] S. Colton: Automated Theorem Discovery: A Future Direction for Theorem Provers. Proceedings of the 1st Automated Reasoning: International Joint Conference, Workshop on Future Directions in Automated Reasoning, pp. 38-47, 2001.

[13] S. Colton, A. Meier, V. Sorge, and R. McCasland: Automatic Generation of Classification Theorems for Finite Algebras. Automated Reasoning, Lecture Notes in Computer Science, vol. 3097, Springer, Heidelberg, pp. 400-414, 2004.

[14] G. Dalzotto and T. Recio: On Protocols for the Automated Discovery of Theorems in Elementary Geometry. Journal of Automated Reasoning 43.2, pp. 203-236, 2009.

[15] B. Davey and H. Priestley: Introduction to Lattices and Order, 2nd Edition. Cambridge University Press, 2002.

[16] M. Davis: The Early History of Automated Deduction. Handbook of Automated Reasoning, vol. 1, Elsevier Science, pp. 3-15, 2001.

[17] R. Diestel: Graph Theory. Springer, 1997.

[18] W. Fang, I. Takahashi, Y. Goto, and J. Cheng: Practical Implementation of EPLAS: An Epistemic Programming Language for All Scientists. Proceedings of the 10th International Conference on Machine Learning and Cybernetics, the IEEE Systems, Man, and Cybernetics Society, pp. 608-616, 2011.

[19] H. Gao, Y. Goto, and J. Cheng: Automated Theorem Finding by Forward Deduction Based on the Semi-lattice Model of Formal Theory: A Case Study in NBG Set Theory. Proceedings of the 9th International Conference on Semantics, Knowledge and Grid, IEEE Computer Society Press, pp. 21-28, 2013.

[20] H. Gao, Y. Goto, and J. Cheng: Research on Automated Theorem Finding: Current State and Future Directions. J. J. Park, Y. Pan, C. Kim, and Y. Yan (Eds.), Future Information Technology, The 9th FTRA International Conference, FutureTech 2014, Zhangjiajie, China, May 28-31, 2014, Proceedings, Lecture Notes in Electrical Engineering, vol. 309, Springer, Heidelberg, pp. 105-110, 2014.

[21] H. Gao, Y. Goto, and J. Cheng: A Systematic Methodology for Automated Theorem Finding. Theoretical Computer Science 554, Elsevier, pp. 2-21, 2014.

[22] H. Gao, Y. Goto, and J. Cheng: Explicitly Epistemic Contraction by Predicate Abstraction in Automated Theorem Finding: A Case Study in NBG Set Theory. N. T. Nguyen, et al. (Eds.), Intelligent Information and Database Systems, The 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part I, Lecture Notes in Artificial Intelligence, vol. 9011, Springer, pp. 593-602, 2015.

[23] H. Gao, Y. Goto, and J. Cheng: Automated Theorem Finding by Forward Reasoning Based on Strong Relevant Logic: A Case Study in Graph Theory. Future Information Technology, The 10th FTRA International Conference, FutureTech 2015, Hanoi, Vietnam, May 18-20, 2015, Proceedings, Lecture Notes in Electrical Engineering, Springer, Heidelberg, 2015. (to appear)

[24] H. Gao, K. Shi, Y. Goto, and J. Cheng: Automated Theorem Finding by Forward Deduction Based on Strong Relevant Logic: A Case Study in NBG Set Theory. Proceedings of the 11th International Conference on Machine Learning and Cybernetics, ICMLC 2012, Xian, China,The IEEE Systems, Man, and Cybernetics Society, pp. 1859-1865, 2012.

[25] H. Gao, K. Shi, Y. Goto, and J. Cheng: Finding Theorems in NBG Set Theory by Automated Forward Deduction Based on Strong Relevant Logic. D.-Z. Du and G. Zhang (Eds.), Computing and Combinatorics, The 19th Annual International Conference, COCOON 2013, Hangzhou, China, June 21-23, 2013, Proceedings, Lecture Notes in Computer Science, vol. 7936, Springer, Heidelberg, pp. 697-704, 2013.

[26] L. Goble: The Blackwell Guide to Philosophical Logic. Oxford, 2001.

[27] Y. Goto, H. Gao, T. Tsuji, and J. Cheng: Practical Usage of FreeEnCal: An Automated Forward Reasoning Engine for General-Purpose. Proceedings of the 11th International Conference on Machine Learning and Cybernetics, ICMLC 2012, Xian, China, The IEEE Systems, Man, and Cybernetics Society, pp. 1878-1883, 2012.

[28] P. R. Halmos: Naive Set theory. Springer Science & Business Media, 1960.

[29] K. Hrbacek and T. Jech: Introduction to Set Theory, Third Edition. CRC Press, 1999.

[30] D. Jacquette: A Companion to Philosophical Logic. Oxford, 2002.

[31] R. Kaye: Models of Peano Arithmetic. Clarendon Press, 1991.

[32] R. McCasland, A. Bundy, and S. Autexier: Automated Discovery of Inductive Theorems. Journal of Studies in Logic, Grammar and Rhetoric 10.23, pp. 135-149, 2007.

[33] A. Montes and T. Recio: Automatic Discovery of Geometry Theorems Using Minimal Canonical Comprehensive Grobner Systems. Proceedings of the 6th

International Workshop on Automated Deduction in Geometry, Lecture Notes in Computer Science, vol. 4869, Springer, Heidelberg, pp. 113-138, 2007.

[34] Y. Puzis, Y. Gao, and G. Sutcliffe: Automated Generation of Interesting Theorems. Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, AAAI press-the Association for the Advancement of Artificial Intelligence, pp. 49-54, 2006.

[35] T. Recio and M. Z. Velez: Automatic Discovery of Theorems in Elementary Geometry. Journal of Automated Reasoning 23.1, pp. 63-82, 1999.

[36] P. Suppes: Axiomatic Set Theory. Courier Dover, 2012.

[37] I. Takahashi, S. Nara, Y. Goto, and J. Cheng: EPLAS: An Epistemic Programming Language for All Scientists. Proceedings of the 7th International Conference on Computational Science, Lecture Notes in Computer Science, vol. 4487, Springer, Heidelberg, pp. 406-413, 2007.

[38] P. Tang and F. Lin: Discovering Theorems in Game Theory: Two-Person Games with Unique Pure Nash Equilibrium Payoffs. Artificial Intelligence 175.14, pp. 2010-2020, 2011.

[39] A. Quaife: Automated Development of Fundamental Mathematical Theories. Kluwer Academic, 1992.

[40] L. Wos: Automated Reasoning: 33 Basic Research Problem. Prentice-Hall, 1988.

[41] L. Wos: The Problem of Automated Theorem Finding. Journal of Automated Reasoning 10.1, pp. 137-138, 1993.