

教育用オープンソースソフトウェア
GeoGebraの証明機能実装に関する研究

甲南大学総合研究所

叢書 137

教育用オープンソースソフトウェア GeoGebra の証明機能実装 に関する研究

高橋正(甲南大学 知能情報学部) 篠田有史(甲南大学 共通教育センター)

要旨

GeoGebra は、オーストリアのリンツ大学 (Johannes Kepler University of Linz) の Markus Hohenwarter 教授を中心とするグループにより開発が進められている幾何学、代数学、解析学を結び付けた動的な数学ソフトウェアである。ソースコードはGPLの下でリリースされており、オープンソースの(無料で使え、プログラムの開発が可能な) ソフトウェアである。

GeoGebra は、近年、証明機能の開発が行われている。この証明機能とは、コンピュータを用いた自動定理証明 (Automated Theorem Proving(ATP)) で、自動推論(AR) の中でも最も成功している分野でありコンピュータプログラムによって数学的定理の証明を発見することである。基礎となる論理によって、定理の妥当性を決定する問題は簡単なものから不可能なものまで様々である。

GeoGebraの証明機能については、日本では利用者が少なく、その使用方法に関しても日本語の文献がほとんどない状態である。日本では、数学ソフトウェアを効果的に用いた数学教育に関する実践及びその教育研究が遅れている。欧米では数学ソフトウェアを数学教育の道具として使用し、ソフトウェアを効果的に用いた数学教育の次の目標として数学ソフトウェアを用いた証明教育を掲げている。日本では、数学ソフトウェアは数学や工学分野研究者のエキスパートシステムであると見られ、研究者が研究の一貫として自動証明ソフトウェアを使用している。本研究の目的は、日本における数学ソフトウェアを用いた証明教育として、GeoGebraのProving機能を用いたICT環境の構築に関し、以下の順序で研究を進める。

- ・ The Gap of Complete Proof and Incomplete Proof in Mathematics Education using Technology

- Teaching and Learning by using GeoGebra
- GeoGebra のビルドについて
- GeoGebra の Proving 機能を用いた授業のための ICT 環境の構築

§1. はじめに

(以下の部分は、[1] からの引用である。)

自動定理証明とは、自動推論において最も成功している分野であり、コンピュータプログラムによって数学的定理に対する証明を発見することである。基礎となる論理によって、定理の妥当性を決定する問題は簡単なものから不可能なものまで様々である。

現在の定理証明プログラムは、以下のように大別される。

- 論理式を入力すれば、自動的に証明する (ものによっては証明も出力する)、いわゆる prover
- 証明に用いる定理や規則を対話的に入力する、いわゆる proof assistant
この proof assistant として、Isabelle/HOL、Coq が公開されている。
- 証明全体を入力すれば、その正しさを検証してくれる、いわゆる proof checker

本研究の対象とする GeoGebra は、prover のタイプである。自動化の度合いと扱える定理のレベルとは相反する要求であり、手放しで証明できるのは、比較的少ない前提で扱えるもののみであり、それなりの内容の証明には、相応の人間の入力が必要となる。

([1] からの引用終了。)

§2. The Gap of Complete Proof and Incomplete Proof in Mathematics Education using Technology

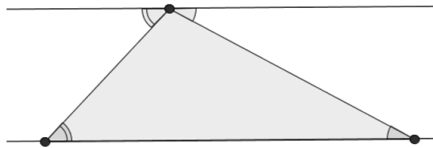
(以下の部分は、[2] からの引用である。)

2.1 Complete Proof and Incomplete Proof in Mathematics Education

We illustrate the distinction between a complete proof and an incomplete proof for the learner in mathematics education using dynamic geometric software (DGS).

The study of the elements of plane geometry yields the best opportunity

to grasp the idea of a rigorous proof. Let us take as an example the complete proof of the following theorem: For any triangle, the sum of the three angles is equal to two right angles, which is an inalienable mental property for most of us. Assume a line through a vertex A parallel to the side BC of a triangle. Then the angles of the triangle at B and at C are equal to certain angles at A, since alternate angles are equal in general. The three angles of the triangle are equal to three angles with a common vertex A forming a straight angle, that is, two right angles, and so the theorem is proved (G. Polya. (1954)) [3].



Incomplete proofs ought to be carefully distinguished from complete proofs; to confuse one with the other is bad, but to pass off one as the other is worse. However, incomplete proofs may be useful when employed in their proper place and in good taste. Their purpose is not to replace complete proofs, which they never could, but to lend interest and coherence to the presentation of a fact.

As the example of a complete proof and an incomplete proof of the above, we show a typical example using DGS as below.

Example 1. (Example of a complete proof)

We draw a quadrangle ABCD and take the four middle point P, Q, R, S of the sides AB, BC, CD, DA, respectively. Then the problem is to show what kind of quadrangle PQRS is.

We consider the segment PS and the diagonal BD, which is a supplementary element, and apply the midpoint connector theorem. The segment PS and the diagonal BD are parallel to each other. Similarly, the segment QR and the diagonal BD are parallel, and so the segments PS and QR must be parallel by transitive inference. Therefore, the quadrilateral PQRS satisfies the

definition of a parallelogram.

Example 2. (Example of an incomplete proof)

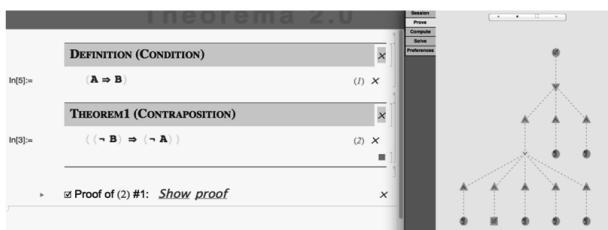
The angle of circumference is half of the central angle. This famous theorem is often shown to a learner by performing many measurements.

In this example of using DGS, the theorem is not proved with technology, rather by the teacher only showing cases. However, then learners' interest in the theorem is excited, and learners incise the property of the properties of the angle of circumference. This is an effective way for letting learners recognize the objective. In addition, this example is a typical one for showing how an incomplete proof is different from a complete proof.

2.2 Automated proof

Automated theorem proving (ATP) is a subfield of automated reasoning and mathematical logic, focusing on the proving of mathematical theorems using computer programs. The Theorema system (THEOREMA) [4] allows algorithms to be expressed using the predicate logic of the Mathematica programming language. We can use this "Theorema language" to express propositions from set theory. We use only the basic operations defined earlier. The syntax, including all symbols, is interpreted unambiguously by the Theorema parser into Theorema's internal representation.

In logic, if the statement is true, then its contrapositive is true. We can prove it by using THEOREMA. Is this proof complete proof? We must think about it in a viewpoint of the mathematics education.



([2] からの引用終了。)

§ 3. Teaching and Learning by using GeoGebra

(以下の部分は、[5] からの引用である。)

3.1 Gaining knowledge and solving problems

Human modeling can elucidate how humans make decisions and how learning ability can be improved. Techniques that have been used widely in this field can be classified into those that integrate a control theory model [25], a probability theory model [6], a will determinism model [11], an information processing model [7], or an artificial intelligence model [16], and those integrating all of these models [23]. The rapid development in research on computer technology, cognitive psychology, and artificial intelligence has led to a shift from behavioristic models [21], which ignore the psychological processes of behavior, to information processing models [14] and artificial intelligence models.

However, these models are still limited with respect to human cognition and while promising new research is being conducted, such research [8] has not been collated into a theoretical system with specific methods. When considering a human model, the model must be viewed as a process model, a knowledge model, and a control model. At present, the models used to describe human information processing include a 7-step user action model, a 3-level human action model, a human cognition model, and a knowledge model. Unfortunately, there is currently no optimal model of understanding in cognitive science. When constructing a new model of human cognition, the effects of various factors should be accounted for. Attempting to construct a model based on the consideration of as many factors as possible would

complicate the model with too many hypotheses and parameters, making it difficult to test.

Therefore, the human model proposed in the present work is based on the three-level model of Rasmussen [13] arose from research on human actions in large-scale systems such as an atomic energy plant or an aircraft. We can consider that such a model has been applied to understanding in cognitive science. Thus, this model was used to position problem-solving strategy and clarify cognitive changes. Almost all human cognitive activities arise from an interactive process between information from the outside world and the knowledge possessed by the individual. Information “from the outside world” is affected by the cultural background and social customs. “Knowledge possessed by the individual” is influenced by the experiences of the individual. Despite this complicated and continuous interactive reaction between external and internal factors, there are environments and stimuli common to all humans; common human reactions exist. Internal knowledge includes not only linguistic and analytical intellectual knowledge, but also non-linguistic and comprehensive intellectual knowledge. Humans seem to use both types of knowledge to understand, learn, and act. Determining whether such human activities can be accurately evaluated and included in a theoretical framework is the key to developing a user interface that can elicit the full intellectual potential of humans.

If we want to conduct effective cognitive science that identifies the source of understanding and creativity, rather than simply utilize unconscious human actions to reach into the depths of human psychological activities, then we must establish a framework that explains the communication between the external and internal sources of knowledge. Our proposed human model of understanding in cognitive science is based on a three-level model of human action. Human researching begins with physical processes, followed by visual processes, and finally intuitive (symbolic) processes. In rule-based actions, classifying behavior occurs in response to problems.

This makes efficient search possible, by indicating what should be done next based on the present situation. To examine how the classification is expressed and constructed, subjects are given several problems to solve for the present study. If the calculation procedure is incorrect when the behavior classification is being constructed, the results will not be predictable. What required identification was how easy it was to reconstruct the classification and what conditions facilitate the identification of the actions that were the cause of the error. In order to assess the degree of freedom achieved, a measurement method was required for identifying the necessary conditions for moving from a rule-based activity to a skill-based activity. Knowledge-based behavior results from cognition and interpretation of external conditions and the construction of a psychological model that uses skill- and rule-based behavior as a solution process.

This requires knowledge of how humans solve problems. Information must be organized and recorded for cognition and understanding. Receiving a message means that the message is reconstructed. In order to do this, one must have the resources with which to conduct this reconstruction. An agent should be able to function appropriately with human common sense and the ability to learn. The agent should always consider safety and accuracy and also possess the ability to explain what is dangerous. The issues in designing such an agent and its actual use must be identified. Mistakes in knowledge-based behavior due to illusions or uncertain memory can lead to the inability to choose or the lack of knowledge of an operation procedure. To integrate these skill-, rule- and knowledge-based behaviors, the three-level human action model was painstakingly reconstructed in order to establish a useful human model, which is used for the learner in cognitive science.

Gaining knowledge and solving problems is two side of the same coin. Reasoning is a reduction of the complex to the simple. Mathematics is the science of "explanation" (reducing everything to "plain things"). By using

theorem prover, we can reduce the proofs of complicated theorems to simple steps. The development of theorem prover is an exciting study of mathematics and computer science. Creating the mathematical technology (creating the algorithm) proceed by thinking and applying the results. There are many computer algebra systems for proving mathematical theorems in automatic mode. In this paper, we will show the structure of GeoGebra.

3.2. Human Model

According to the three-level human behavior model of Rasmussen, automatic human actions can be classified into the three levels of skill-, rule- and knowledge-based actions (Fig. 3-1, [20]).

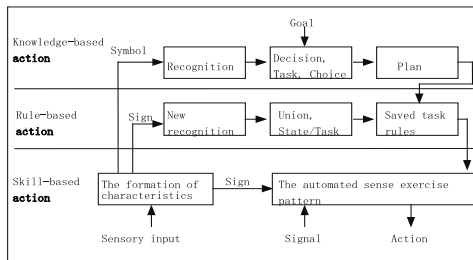


Fig 3-1: Three-Level Model of Human Action

Skill-based action: These actions occur at the stage when some intention is formed. However, the actions are automated and are executed without the control of an action pattern.

Rule-based action: Actions that utilize a previously acquired rule in order to realize a specified purpose. The learner constructs and executes a series of actions.

Knowledge-based action: The learner recognizes the situation, manipulates a psychological model and considers possible strategies.

A skill-based action is a response that occurs in less than 1 second [17]. A chain of skill-based actions is a rule-based action. Thinking about how to solve a problem is a knowledge-based action. Skill-based actions are

performed smoothly without intentional control. Rule-based actions require a great deal of repetitive practice in order to be transferred to the skill-based level. First, the external conditions must be recognized then the rules for composing the act are combined with the conditions required to carry out the behavior. Knowledge-based actions require the recognition of external conditions, the interpretation of these conditions, the construction of a psychological model for considering solutions, planning and finally, the use of the other two behavior levels to carry out the action. This is a process model in which mastery of behavior requiring thought is internalized to the point where it can be carried out unconsciously. Mistakes can be explained as omitted steps, or for example, as pushing the wrong nearby button in smoothly carried out skill-based actions. In the case of knowledge-based actions, illusion can lead to error. In the present study, this process was analyzed using Rasmussen's three-level human behavior model in order to identify what functions are essential to facilitating smooth action and learning. Behavior used to learn about problems and how to solve them is classified in detail according to the three-level model. Humans act by classifying issues and their relationships by consciously combining them. Humans control themselves by constantly observing, thinking about, evaluating and integrating their behavior in order to achieve accuracy, continuity, consistency and normality [13]. Classified factors can be separated into the same three levels as the general actions.

3.3 Strategy

In cognitive science, humans use strategies to solve problems. Strategies are used as knowledge to plan solutions and decide procedures. When these procedures, in general or for the most part, obtain the correct answer, the procedure is called a heuristic; however, such heuristics do not always result in a correct solution. The study of factors related to the early stages of strategy is based on observing the results of an individual's

problem-solving method. Following the initial stage, the individual passes through a series of stages and then reaches a solution. However, consensus has not been reached regarding the meaning of these stages. For example, Wallas identified the following four stages involved in problem solving in the book “The Art of Thought” [24]:

- I. Preparations (collecting information)
- II. Warming up (Separating from the problem)
- III. Flash (Sudden insight)
- IV. Verification (Reaching a solution)

Duncker’s observation that “A person resolves the problem into a goal for the lower rank, and then he tries to solve the goal for the lower rank” [9] could not be verified. At this time, not much could be inferred about human accomplishments and an individual’s thought processes could not be described clearly. Subsequently, cognitive psychology came to the forefront and Polya clarified a method for teaching problem-solving strategies in [18]. Ernst and Newell and later Newell and Simon, extracted strategies using the following method [15]. A man involved in solving a problem was directed to communicate the contents of his decision-making process and his thoughts through speech. The strategy used by the man was extracted from the records of his speech. Newell designed the General Problem Solver (GPS) computer program based on the problem-solving methods that were clearly and accurately described. This GPS successfully solved the following 11 problems, which are all different kinds of problems.

- I. Missionary and cannibals (Three missionaries and three cannibals must cross a river using a boat that only two people can ride at a time)
- II. The integral calculus (The application of an integral formula)
- III. The tower of Hanoi (Different size disks must be moved and placed on three sticks)
- IV. The proof of the theorem (The theorem to solve a function calculation)
- V. Father and sons (A father and 2 sons cross a river on a boat

with a 200-pound-capacity)

- VI. Subject of the monkey (There is a banana at a height that a monkey can not reach)
- VII. Three coins (Two coins can be overturned at a time, and by the third attempt all coins must be showing the same side)
- VIII. Sentence analysis (Distinction of parts of speech)
- IX. Water pot problem (Using five gallon and eight gallon water tanks two gallons of water must be measured)
- X. Character line complete problem (The rule must be detected from a row of characters)
- XI. The bridge of Kernihisberg (When returning to the city, one passes seven cities and seven bridges, but passes each bridge only once)

For a computer to solve a problem, the expression of the problem solution strategy must be accurately described as follows:

- (1) The problem must be expressed as a problem space.
- (2) The problem space consists of the possible operations available to change the premise conditions of the problem, the final conditions of the problem and the middle conditions.
- (3) The problem space has a passage that can be used (dead end).
- (4) Solutions to the problem are to be sought from the viewpoint of problem-solving goals. They are not carried out from a trial and error perspective.

Strategies are used even when human beings solve mathematical problems. Recognition knowledge and experience are used as “doing it like this is effective in this case”. The ability to rapidly reference knowledge is required for strategies based on experience. Furthermore, the recognition of thoughts and feelings controls. The famous book by the mathematician Polya, “How to solve it [18], showed the processes of mathematical problem solving; however, one can not learn how to use heuristics in problem solving just by reading a book. In researching problem solving, there are two contrasting concepts. The first emphasizes insight, flash and senses, while

the second emphasizes experiential knowledge. The former concept employs a strong tendency to perceive that strategies of thought are learned through the experience of problem solving. In other words, it is assumed that an intuitive feelings and specific technical abilities can be acquired. In the latter concept, it is assumed that problem solving ability arises from the accumulation of rules inherent to the domain provided by an individual problem. Such differences depend on the problem's nature, domain and level, and the type of person involved in the learning process. In addition, it is difficult to establish clear boundary lines between these two concepts. In problem solving, experiential knowledge plays a large role. Heuristics are general ideas or algorithms (a procedure providing the correct solution) and are widely used. Heuristics are equal to "the logic of a thought". Examples of extremely general strategies are "try to draw a figure if you come across a difficult problem" and "search for similar problems that you have experience with". There are also concrete strategies we are familiar with, such as "A problem requiring the comparison of quantities requires two differences and a transform formula" and "try to make clauses that differ next to each other for number sum sequence problems" [10]. Therefore, the kinds of strategies that students actually use were examined. The present research explores the differences in the learning of students who adopt the strategy of physical principles without knowledge of the usual learning methods for mathematics and students who adopt the natural strategy of mathematics, which proceeds with logical, progressive thought. These differences are assembled in the student's brain, which determines how these strategies are to be used. Finally, a learner independently searches for heuristics. It is desirable to understand this process, in order to recognize the meaning of cultivating natural mathematical comprehension and thinking power. However, currently, there are many cases in which the learner does not consciously study problems from the perspective that "there are various learning methods and heuristics".

3.4 Theorem Prover

The theorem prover is being used frequently in mathematics teaching. Although good teaching examples and experiences exist, the efficient and successful use is not self-evident, yet. A subtle relationship exists between paper-and-pencil techniques, theorem prover and conceptual understanding. The nature of theorem prover is different from that of paper-and-pencil techniques. Mathematics are presented as a primarily mental activity that involves the construction of mathematical objects and relations. Using the theorem prover requires insight into procedures as well as into the concepts involved.

In cognitive science, use of a computer with the theorem prover can correct the weakness in mathematical thinking. We can clearly understand mathematical concepts and can minimize the burden of operation opportunities. Computer software using a theorem prover have bring about changes in mathematical thinking. Therefore, learners can concentrate on mathematical problems.

GeoGebra: GeoGebra is an interactive geometry, algebra, statistics and calculus application, intended for learning and teaching mathematics and science from primary school to university level. GeoGebra is an interactive mathematics software program for learning and teaching mathematics and science from primary school up to university level. Constructions can be made with points, vectors, segments, lines, polygons, conic sections, inequalities, implicit polynomials and functions. All of them can be changed dynamically afterwards. Elements can be entered and modified directly via mouse and touch or through the Input Bar. GeoGebra has the ability to use variables for numbers, vectors and points, find derivatives and integrals of functions and has a full complement of commands. Teachers and students can use GeoGebra to make conjectures and to understand how to prove geometric theorems. GeoGebra [26] is a good platform for experimentation which supports the development of mathematical concepts

and the abilities to explain geometric properties. By using GeoGebra, we can aim at consolidating students' understandings on the geometrical concepts through experiencing the process of exploring, conjecturing, verifying, justifying and proving is described.

3.5 Knowledge Base in Cognitive Science

When researchers use the theorem prover for the acquisition of knowledge or skills, we must consider a “tool” to be a “symbol device”. A symbol device exists between the researchers and the research subject. Operation activity occurs between a symbol device and the researching subject. In cognitive science, two difficulties exist, one in the interaction between the researcher and the symbol device, and one in the interaction between the symbol device and the research subject. Therefore, we must overcome these difficulties in order to effectively utilize the theorem prover in cognitive science. Moreover, we must assess the benefits of considering the integration of the theorem prover from the perspective of the relationship between mathematical knowledge and mathematical concepts. When theorem provers are used in mathematical studies, researchers achieve a result through their efforts. Then, the researchers must investigate whether conceptual problems exist or whether they simply do not appreciate how the theorem prover works. By using a theorem prover effectively, researchers become aware of numerous mathematical ideas. This is made possible by incorporating the results of research in cognitive science. In carrying out a seven-phase model of human action, “the formation of a series of intentions or actions” must be performed smoothly [22]. The effective use of a theorem prover in cognitive science is influenced by the contents of mathematical thought, and research and understanding of mathematics can further influence general idea formation. The theorem prover influences the “perception - interpretation - evaluation” phases of evaluation. The foundations of this model were studied by Rasmussen as the three-level control model of individuals' actions [22]. We can use

the theorem prover as a material object that is available for the assessment of human activity. The use of the theorem prover can establish automatic and routine procedures. Controlling this automation is essential, especially in research on thought processes. There are three methods for creating a theorem proof (by hand, by mind and with a computer). A researcher's point of view of cognitive science considers the relationship between the brain and mind as the relationship between hardware and software in a computer. According to this point of view, the science of the mind is a special science, the science of thought.

We must distinguish between a tool and an instrument. Recent studies have given us theoretical tools for improved understanding of human/machine interactions. The psychological component is defined through the notion of a scheme. Operations rely on implicit knowledge, specifically, concepts that are implicitly considered. This knowledge guides the action. A scheme is itself the product of an assimilation activity, in which the computer and the available theorem prover play major roles. An instrument is a mixed entity, with a material component (a tool, or a part of a tool activated in order to realize a specific type of task) and a psychological component (the schemes organizing the activity of the subject). The theorem prover can act on several levels as an instrument in cognitive science:

- The first level (the level of the default system);
- The second level of an instrument or a set of instruments;
- The third level (a meta one) of the relationship of a subject with an instrument or a set of instruments.

These three levels correspond to tools in different levels of cognitive science:

- Primary tools, the concept of the tool for initial research;
- Second-level tools, representations and action modes utilizing first level tools;
- Third-level tools, for trained researchers.

In the three-level model of human behavior, operations and strategies can

be identified and considered in relation to human thought processes in order to facilitate error-free problem solving. In consideration of surface features and conditions, similar problems can be recognized and suitable problem-solving methods can be identified. In addition, it was found that contents of the subconscious can be raised to the knowledge-based action level in order to support the expression process and the achievement of efficient functioning.

The technology of theorem prover automated reasoning. The ultimate goal of mathematics is technology. To do mathematics is gaining knowledge and solving problems by reasoning. Theorem prover is a powerful tool for researching mathematics. Researchers should appreciate the possibility of sharing cognitive level with such technology (Fig. 3-2).

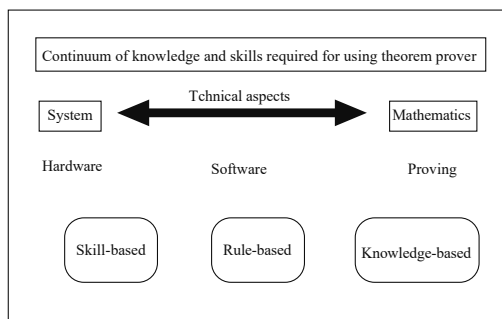


Fig 3-2: The continuum of knowledge and skills required for using theorem prover

Researchers can use theorem prover as a 'partner'. However, researchers must study not to reach for the computer in an unthinking manner. Researchers need to be aware that algebraic expressions using theorem prover syntax may be time consuming. If an expression is to be used repeatedly in the proof of a theorem then this initial time spent on theorem prover entry may be worthwhile but researchers will require guidance in

making these choices. In contrast, strategic use requires the researcher to think about the mathematics and the tool. The value of theorem prover will be realized once researching with the tool simulates rather than impedes mathematical thinking. To reach this stage, researchers must encourage to develop the habits of discerning and strategic use of theorem prover along with the elements of algebraic insight required. In this way, theorem prover gives many opportunities for rich mathematical researching. Technology in mathematics is the fulfilment of the aspiration of mathematics in the computer age. By recent advances in computational logic, computational algebra and software technology, the automation of reasoning promises to become practical feasible and useful for mathematics research. ([3] からの引用終了。)

§ 4. GeoGebra のビルドについて

(以下の部分は、[27] からの引用である。)

GeoGebraの公式サイトにビルドの方法が書かれており、それを参考にビルドを行うことが一般的な方法である。以下にビルドに必要な部分の概略を示す。

(公式サイトに記載されている方法の概要)

Eclipse IDE for Java Developers (Mars, 4.5)を <https://eclipse.org/> からダウンロードする。

GWT for Eclipse plugin for 4.4をSDKなしで必要なコンポーネントのみをダウンロードする。

https://developers.google.com/eclipse/docs/getting_started

Subclipse Eclipse plugin (<http://subclipse.tigris.org/>をダウンロードする。Gradle Integration for Eclipse (4.4) 3.6.4.RELEASEを http://marketplace.eclipse.org/content/gradle-integration-eclipse-44#V_B_gPGOM140 にアクセスしてインストールをクリックする。ここでGradle IDEのみを選択する。

gwt-2.7.0.zipを<http://www.gwtproject.org/download.html>からダウンロードし解凍して、Window → Preferences → Google → Web Toolkit (MacのEclipseでは → Preferences...)から追加する。

空のワークスペースで、File → Import → SVN → Checkout Projects from SVN → Create a new を選択し、<http://dev.geogebra.org/svn/trunk/>を入力してNextをクリックする。そしてGeoGebraを選択してNextをクリックし、Checkout as a project in the workspaceを選択し Project nameをGeoGebraにしてFinishをクリックする。GeoGebraのプロジェクトをPackage Explorer上でDeleteをクリックし、ワークスペースから削除する(ここでディスクからプロジェクトを削除しないこと)。

File → Import → Gradle → Gradle projectで、ワークスペースからGeoGebraを選択し、Build modelをクリックする。ここで、input3Dとinput3D/impl以外を選択し、input3D/stubからチェックを外す。Run beforeの with optionがcleanEclipse eclipseであることを確認し(変更しないこと)、Finishをクリックする。

src/main/javaのorg.geogebra.desktopにある GeoGebra3D.javaを選択し、右クリックしてRun As → Java Applicationを選択する。

上述したソフトウェアの概要は以下の通りである。

- ・ Eclipseとは、IBMが開発した統合開発環境(IDE)であり、多くのプログラミングに言語対応、充実したソース補完などの機能を有する高機能なオープンソースのソフトウェアである。
- ・ GWTとは、Google Web Toolkitのことであり、Googleの提供するサービズを利用するために必要なソフトウェアである。
- ・ Gradleとは、ライブラリの依存関係などを管理しながらコンパイルを行うビルドツールである。
- ・ SVNとは、プログラムのソースコードの管理システムSubversionを扱えるソフトウェアであり、変更履歴や変更点などを管理することができる。

しかし上述した方法の場合、かなりの時間がかかる部分があり、説明不足である点があるため、部分的に別の方法を採用するなどを行った。

公式サイトの方法においての問題点はいくつかあるが、GeoGebraを <http://dev.geogebra.org/trunk>

からEclipseを用いてcheckoutするときに時間がかかりかかってしまう点があげられる。そのままcheckoutすると、数GBにも及ぶ膨大な数のファイルがダウンロードされてしまう。しかし、GeoGebraの開発で用いられてきた更新履歴などを含むbranchやtagが大半であり、開発に関しては、ほとんど必要としないことが多い。

以下、公式サイトの方法に対する追記すべき点および変更点について述べる。

(我々の環境は、MacBook Air 2016Early メモリ4GBであり、EclipseはLuna (4.4)、Java8である。)

- ・ (公式 $\text{\ref{f1}}$ について) EclipseのHelp -> Install New Softwareを選択し、<https://dl.google.com/eclipse/plugin/4.4>と入力して、SDKsを展開してGoogle Web Toolkit SDK 2.7.0 をインストールする。

- ・ (公式 $\text{\ref{f2}}$ について) Gradleをインストールするには、EclipseのHelp -> Eclipse Market Placeを選択し、「Gradle」と入力して検索を行うと、「Gradle IDE Pack 3.8.x+1.0x」が表示されるので、それをインストールする。

- ・ SVNを扱えるソフトウェアを用いて

<http://dev.geogebra.org/svn/trunk/geogebra> をcheckoutする。

- ・ このままビルドしようとする、

cannot invoke method setNativeLibraryLocation() on null project と表示されるので、checkoutする。フォルダ名

File -> Import -> Gradle -> Gradle (STS) projectで、上の処理でcheckoutしたフォルダを選択し、Build modelをクリックする。

ここで、input3Dとinput3D/impl以外を選択し、input3D/stubからチェックを外し、Finishをクリックする。

- ・ (追加) JavaFXを使うために、Eclipseの設定を変更する。Project Explorerでdesktopフォルダを選択、メニューバーからProject ->

Properties → Java build pathをクリックし、LibraryのJRE System Libraryの中にあるAccess Rule をチェックし、edit をクリックする。

その画面でaddをクリック、Resolutionをaccessible、rule patternにjavafx/**と入力しOKをクリックする。

もう一つの方法として上記の手順2までを実行した後以下の手順を行うことでもビルド可能である。

- ・ (追加) EclipseのHelp → Eclipse Market Placeを選択し、「Subclipse」と入力して検索を行うと、「Subclipse 1.10.13」が表示されるので、それをインストールする。
 - ・ Project nameの設定後、Revisionを49522(5.0.280)に設定してからFinishをクリックする。
 - ・ (追加) Package Explorer上に表示されたファイルの次の2箇所を書き換える。
 geogebra/build.gradle の40行目を `gradleVersion = '2.13'`
 geogebra/gradle/wrapper/gradle-wrapper.properties の 6 行目を `distributionUrl=https\textbackslash ://services.gradle.org/distributions/gradle-2.13-bin.zip` にする。
 - ・ 公式ページの方法を実行する。
 - ・ File → Import → Gradle(STS) → Gradle projectで、ワークスペースからGeoGebraを選択し、Build modelをクリックする。ここで、input3Dとinput3D/impl以外を選択し、input3D/stubからチェックを外す。Run beforeのwith optionがcleanEclipse eclipseであることを確認し(変更しないこと)、Finishをクリックする。
 - ・ (追加) Package Explorer上でdesktopを選択し、メニューバーよりproject → Properties → Java Compiler → Errors/Warnings の中のDeprecated and restricted API を展開し、Forbidden reference (access rules):のボックスを“Error”から“Warning”に変更する。
 - ・ 公式ページの方法を実行する。
- 上記の方法は公式サイトの掲載事項のファイルサイズよりかなり小さい現実的である。
- ([27] からの引用終了。)

§5. GeoGebra の Proving 機能を用いた授業のための ICT 環境の構築

(このセクションは、[28, 29] に基づき再構成したものである。)

5.1 ICT環境の構築の概要

幾何学の入門者からは苦手意識を持たれてしまう一方で論理展開の本質を学ぶことができる「証明問題」に関連する内容は、GeoGebra を活用する取り組みの中で、明示的に、あるいは非明示的に取り組みがなされつつある。GeoGebra のオフィシャル Web サイトからも、Proving 機能についての紹介にアクセスすることができ、作図のみならず、幾何学問題の証明についても、GeoGebra のプロジェクトが野心的に取り組んでいることが示唆される。コンピュータを活用した幾何学の証明を、どのように教育に役立てることができるかという題材は、非常にチャレンジングかつ興味深いものであると考えられる。すなわち、どのような題材で、どのように提供される証明機能を組み合わせ、どのような教示のもとで学習者の試行錯誤をひきだして、数学的な知的活動の体験へと昇華させるか、という問題である。ただし、この問題に取り組む上では、そもそもそのような学習体験は意味あるものになりうるのか、という議論が必要である。上記のような議論は、机上の検討だけではなく、授業の中で実証的に研究されるべき性質のものである。そのためには GeoGebra の Proving 機能を利用して学習するという体験を円滑に提供するための ICT 環境を構築し、実証につなげていくことが必要である。そして、この仕組みの実現のためには、まず、GeoGebra がどのような証明に関する機能を提供できているか、という部分についての調査が必要である。調査の上で証明に関する機能に介入し、体験に役立つようにアレンジした画面出力を得る方が可能であるかについても調査が必要である。さらに、体験に役立つアレンジを施したアプリケーションを、円滑に学習者に提供する仕組みも必要である。そこで、本研究においては、以下の 4 つの事項に分けての検討を行い、GeoGebra の Proving 機能を用いた授業のための ICT 環境の構築について考察する。第一の事項は、証明機能を持ったソフトウェアはどのように教育に役立てることができるかという、証明機能の利用法略について検討である。第二の事項は、GeoGebra の証明機能について検討し、どのような構造で処理が行われているかを明らかにする。第三の事項は、GeoGebra の証明機能に介入し、最初に検討した証明機能をもったソフトウェアに求められる幾何学証明の実現のため、どのような工夫が可能であるかを検討する。最後に第四の事項として、検討した枠組みを学習者にどのように提供するかを考察

した。

5.2 証明機能を持ったソフトウェアの教育への活用方略

ポリアは著書「いかにして問題を解くか」の中で、完全な証明と、不完全ではあるが有効な証明の概念を提案している[31]。この中でポリアは、不完全な証明と完全な証明は別のものであり、不完全な証明は完全な証明の代替にならないことを述べ、完全な証明については、論理的体系とし、一方で、証明を放棄して表面的な実利だけを求めるものを、料理本的体系としている。そのうえで、

厄介すぎる証明と料理本のレベルとのどちらをとろうかという、矛盾を解決するいちばんよい方法は、不完全な証明をうまく利用することである。

厳格な倫理家にとっては不完全な証明は証明ではない。そうして不完全な証明と完全な証明とは、はっきり区別しておかなければならない。不完全な証明は適当な場処で上手につかえば役に立つ。その目的は完全な証明の代りをする事（それは不可能である）ではなくて、書いてあることに興味と関連性をもたせることである。

と述べ、不完全な証明であっても、有効に機能する部分があるのではないかと、という提言を行っている。

そこで、ユーザーフレンドリーな図形描画機能を有するソフトウェアである GeoGebra について、作図の中で、不完全であっても有効な証明を使って学ぶというアプローチを提案することができれば、興味をわかせる「そうらしい」と学習者の心の中に刻み込むことができるものと考えられる。このような、GeoGebra の新しい活用の方向性を探る上では、GeoGebra の持つ証明機能がどのように動作しているのかを明らかにする必要がある。また、GeoGebra で扱うことができる証明に関する例題を明らかにし、教材としてまとめていくことが必要になる。ここで、不完全であるが有効な証明の例題を検討する上では、GeoGebra が作図ソフトであるという部分を有効活用することが切り口のの一つとして考えられる。GeoGebra の作図をベースとした証明を考えた場合、例えば 3 つの点 A、B、C で三角形を描いたとき、この三角形 ABC が 1 本の線に潰れていないことは、三角形 ABC を目にして学習者にとっては明らかなことではある一方、数学的にみると、A、B、C に適切な条件を付け加えなければ三角形であることは保障できない。GeoGebra の作図に基づいた証明のユーザー体験を考えると数学的に満足

できるレベルで表現されたロジックによって成り立つ完全な証明と同等の厳密さを扱うことは困難であると予想される。しかし、ここで、「この図では正しい」、「今回の作図では並行」、といった視点で評価をすることができれば、「諸々の条件が作図のような条件の場合」という表現によって、条件の定義を大幅に単純化することができる。これこそが有用な理解への刺激、すなわち、不完全であるが有効な証明となりうるのではないかと考えられる。

完全な証明は、それを理解する側の知識を前提とする。前提とする数学の知識が、完全な証明を成立させる必要条件となっている。これらのことを勘案すると、Prover ソフトウェアを用いた際の機能として、「この図においては正しい」との表示を得ることで、前提となる数学の知識を動員することが可能になれば、「Prover ソフトウェアを用いた結果+学習者の数学の知識」によって、完全な証明を完成させることが可能になる。

5.3 GeoGebra の Prover の機能

5.3.1 GeoGebra を利用した研究開発の概要

2019年10月25日現在、GeoGebraはWebアプリケーションとして、ユーザーに公開されている。このWebアプリケーションとしてのGeoGebraは、多数の研究者が関わり、専門の技術者によってオンラインに公開されてメンテナンスを受けているものである。今回我々が研究開発の対象とするのは、このWebアプリケーションの元となったGeoGebraClassicと呼ばれるアプリケーションである。もともと、GeoGebraはWebアプリケーションとして開発されておらず、最初は実行ファイルをダウンロードしてコンピュータにインストールしたうえで利用する形態で配布が行われていた。この形態用のソースコードについては、オフィシャルサイトで公開がなされている。

そこで我々は、このソースコードをインターネット経由で入手し、研究開発環境を構築することとした。研究に着手した2017年の6月の段階においては、オフィシャルサイトで公開されていたが、その段階でのGeoGebraのソースコードをビルドする方法については福井らの研究[30]を参考にした。

本研究では、オペレーティングシステム：Windows10 (64ビット日本語版)、Eclipse (Version: Luna SR2 (4.4.2))、JRE (JavaSE-1.6)を主な開発環境として利用し、処々のプログラム・開発ツールについては福井らが[30]にて紹介しているものを規準に、オフィシャルサイトの情報も加味してビルドした。

5.3.2 GeoGebra のプログラムの概要

Java のプログラムとして動作が可能になった状態で、GeoGebra のプログラムの構成について述べる。

GeoGebra は多くのクラスから構成されるプログラムで、フォルダ構造から大きく 3 つに分かれているものと考えられる。基盤となる common (と base)、そして、web と desktop である。今回は、開発の主体として desktop に含まれているプログラム群を中心の検討を行った。いわゆるコンピュータの画面上でアプリケーションとして動作する GeoGebra については、

```
/desktop/src/main/java/geogebra/GeoGebra3D.java
```

にメインクラスが存在する。この中の

```
Org.geogebra.desktop.GeoGebra3D.main(cmdArgs);
```

である。

動作の基幹となる部分については common フォルダに格納されている。例えば、後述する証明機能部分については、

```
/common/src/main/java/org/geogebra/common/kernel/prover
```

 フォルダに証明のための類型を管理するためのファイルと考えられる java ファイルが格納されている。

GeoGebra は大規模なプログラムで、例えば、GeoGebra Ver 5.0.366 のソースコードについてプロジェクトフォルダに含まれる java ファイル数は 5673 個であった。これらはビルドに必要な、GeoGebra 関連の外部のソースを含んでいないので、ビルドに用いられるファイルの数という意味ではさらに数は膨大になる。このようなプログラムを、GeoGebra は複数の研究者・開発者が関わるオープンソースの形で開発がなされており、その全容の把握は容易ではない。他方、今回我々が目指すのは、教育の形態としての在り方の検討であり、GeoGebra のプログラム全体を把握する必要は必ずしもなく、証明を用いる教育の鍵となる証明機能部分についてどのようなアプローチが可能であるかを検討することが最も重要となる。そこで本研究では、証明機能に使われている情報を手掛かりに証明部分に関する調査を出発点とする形で調査を行い、必要十分な拡張ができるかどうかを検討した。

5.3.3 GeoGebra の証明機能部分の動作

前述の議論からは、証明機能については、有効で不完全な証明と、完全な証明との間を、いかに GeoGebra で結び付けていくか、その具体的な例題はなにか、といった部分を取り扱うに足る機能を有している必要があることが明らかになった。一方で、GeoGebra の証明機能については、どのような機能があり、どのように動作しているかを明らかにしなければ、教示方略を議論するための土台になる例題を作成できない。

GeoGebra の証明機構は Prover と呼ばれ、オフィシャルサイトでは Ver. 5 より利用が可能になったというアナウンスが行われていた[32]。そこで本研究では、どのように証明問題にアプローチするべきかを検討するために、Prover の証明機能の状況を明らかにすることを目標とした。具体的には、現状のバージョンでどのようなことが可能であるか、教示に求められるような機能拡張はどのようにしたら実現できるのか、という大きく分けて2つの方向性である。

研究プロジェクトでは、2017年11月にこの部分を重点的に検討したため、報告としてはこの時点の Prover の動作状況について述べる。2017年11月10日の段階で、GeoGebra wiki には、Theorem Proving のセクションが存在しており、このセクションの最終更新は2015年の6月であった[32]。このセクションには、Prover の動作はまだ開発途中であり制約があること、4つのタイプの Prover が実装されていることが記載されていた。Recio, Botana, Puresymbolic, OpenGeoProver の4つの Prover である。また、それぞれの Prover には機能制約があると記載されているものの、その機能制約の説明リンクをたどっても、情報が表示されない状態であった。

そこで、Proving 機能がどのように動作しているのかを明らかにするために、2017年11月10日現在の最新版であった Ver 5.0.396 について動作を検証することとした。また、このときの対象は Relation 関数と連携して行われる Proving とした。ここでは、図 5-1 に示す図形を描画し、Prover が動作するプログラムを入力した。

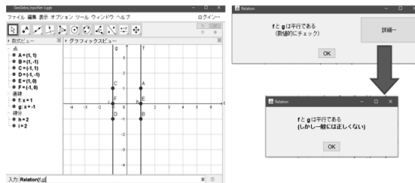


図 5-1. 基本図形とそれに対する Relation[f, g] を実行した結果

図 5-1 に、基礎的な図形の描画とそれに対する $\text{Relation}[f, g]$ を実行した結果を示す。ここでは、原点の周りに 4 つの点を配置し、縦に直線を引いて、直線と直線の関係、点と直線の関係、線分と線分の関係という 3 つの関係を調査することとした。動作結果を表 5-2 に示す。 $\text{Relation}[f, g]$ を実行すると、GeoGebra は自動で f と g が平行である、という関係性の調査結果を出力する。ここで詳細ボタンをクリックすると、Prover が動作し、評価の結果が表示される。入力に対して得られた結果は、次の表 5-2 のようになった

表 5-2. Ver 5.0.396 における振る舞い

構文	詳細における画面出力
$\text{Relation}[f, g]$	f と g は並行である (しかし一般には正しくない)
$\text{Realaion}[A, f]$	A は f 上にある (常に正しい)
$\text{Relation}[AB, CD]$	$\text{Distance}[A, B]$ と $\text{Distance}[C, D]$ は等しい (しかし一般には正しくない)

他方、図 5-1 や表 5-2 で得られた「しかし一般には正しくない」という表現が、どのように得られているのかを明らかにすることができれば、GeoGebra の証明問題への活用方略へのヒントが得られるものと考えた。しかし標準の状態では、これ以上の情報を得ることはできず、内部で行われた処理を伺い知ることはできなかった。

この証明結果については、先行する取り組みで古いバージョンの GeoGebra を使った事例があった。表 5-3 に、2017 年初頭の段階における Prover の動作結果を示す。

表 5-3 の段階では、各点の座標に 0 が含まれるかどうかで、証明結果に変化が現れる等、不安定な部分が見受けられたものの、今回においてはそのような現象は確認できなかった。また OpenGeoProver の動作において、表 5-3 では「恐らく一般に正しい」といった出力があったが、今回はそのような出力は確認できなかった。このことから Prover の振る舞いについて、継続的に開発が行われていると推察した。

表 5-3. 2017 年初頭における Prover の振る舞い([30]より)

座標と式の条件	(A) マウスで点を配置	(B) 入力バーから座標を入力
A=(-3,3),B=(-3,-3) C=(3,-3),D=(3,3) f=Line[A,B],g=Line[C,D]	Relation[f,g]: ①,R Relation[A,f]: ②,B Relation[AB,CD]: ①,B	Relation[f,g]: ①,R Relation[A,f]: ②,B Relation[AB,CD]: ③,O
A=(-3,3),B=(-3,0) C=(3,0),D=(3,3) f=Line[A,B],g=Line[C,D]	Relation[f,g]: ①,R Relation[A,f]: ②,B Relation[AB,CD]: ①,B	Relation[f,g]: ①,R Relation[f,g]: ③,O Relation[AB,CD]: ③,O
A=(-3,3),B=(-3,0) f=Line[A,B] 上を作成後に C=(-3,-3)	Relation[f,A]: ②,B Relation[f,B]: ②,B Relation[f,C]: ①,B	Relation[f,A]: ③,O Relation[f,B]: ③,O Relation[f,C]: ①,O

R : Recio B : Botana O : OpenGeoProver_WU

① : しかし一般には正しくない ② : 常に正しい ③ : 恐らく一般に正しい

5.4 GeoGebra の証明機能への介入

5.3 において、Prover.java について呼び出しと処理が行われていること、および得られた出力について確認を行った。そこで、具体的にプログラム内部を調査し、Prover の状況をさらに掘り下げて調査した。さらに、5.2 で検討した証明機構に求められるリクエストを達成するための実験を行なった。この節で紹介するプログラムについては、2017 年 6 月の時点で配信されていた Ver 5.0.366 をベースとした。

5.4.1 UI のダイアログからの処理内容の検討

まず、UI ダイアログの情報から内部処理でどのように Prover が用いられているかを確認することとした。次に、画面に表示される UI のダイアログを、状況に応じて、書き換えることができるかどうか検討を行った。

手がかりとして用いるのは、5.3 で確認した、「しかし一般には正しくない」という表現である。この表現がどのようにプログラムの中に格納され、どのタイミングでどこから発せられているかを調査することで、今回求められている Prover の介入を行うためのヒントが得られると考えられるからである。

検討の結果、表示される日本語の情報は、Unicode エンコードがなされて格納されていることが確認できた。すなわち、UI の出力に見られる日本語に対して Unicode エンコードを施し、プログラムコード全体に対して検索を行うと、該当する日本語が使用されている部分を確認することができる。今回は、「しかし一般には正しくない」という表現を対象に調査を行った。この語を Unicode に変換すると、

```
¥u3057¥u304b¥u3057¥u4e...
```

という表現で開始する文字列となる。この文字列について検索を行うと、プログラム内の幾つかの部分で用いられていることが確認できた。Ver 5. 0. 366 において、日本語が含まれているファイルとそのパスは、次の 2 つであった。

```
/desktop/src/nonfree/resources/org/geogebra/desktop/properties/menu_ja.  
.properties  
/web/src/nonfree/resources/org/geogebra/web/pub/js/properties_keys_ja.  
js
```

この 2 つのファイルの中には、次のような表記が存在しており、対応する日本語とその元となったプログラムで用いられている文字列とを対応して確認することができた。“ButNotGenerallyTrue”: “(¥u3057¥u304b¥u3057¥u4e…この表現から、「しかし一般には正しくない」という表現は、プログラムの中で、ButNotGenerallyTrue というパラメータで管理されていることが確認できた。

すなわち、ButNotGenerallyTrue に処理結果等を格納する箇所が、今回、我々が目的としている証明処理が行われている箇所ということである。そのため、このパラメータが用いられている箇所を中心に、いくつかコンソールに出力を得るためのコメント文を挿入し、動作の確認を行なった。

また、“ButNotGenerallyTrue”を手掛かりにソースコードを調べることで、Prover 関連のファイルを格納しているフォルダも確認することができた。

例えば、図 5-4 に示すように、

```
/common/src/main/java/org/geogebra/common/kernel/prover  
に、多数の java ファイルの存在を確認した。
```

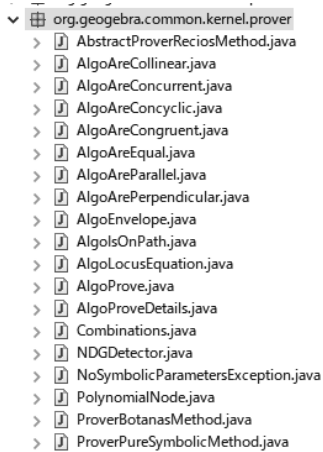


図 5-4. Prover 関連の java ファイルの例

図 5-4 に記載されたファイル名からは、例えば平行であるかを検討する、或いは等しいかどうかを検討する、といった用途に合わせてクラスが編成されていることが分かる。ただし、それらがどのように呼び出されて処理を実行するのか、また証明の中で実際に使われているのかそうでないのかといった部分については明らかにできなかった。他方、数千個の java ファイルからなるプログラムについて、デバッグ機能だけで該当部分の動作を捕捉して理解するのは困難であった。そのため、本研究ではプログラムの各所に動作状況を出力するためのコメント文を差し込み、該当する機能が起動されるかどうかも含めて検討を行なった。

5.4.2 コメントの挿入

Eclipse 上で GeoGebra をソースコードからビルドして動作させると、Eclipse のコンソールに状態出力を得ることができる。これは Prover に関しても同様で、動作に関する内部情報が得られるものの、どのように動作しているのかは一見して把握することが困難である。そこで、ここでは、自由にコメントをさしはさむ方法を確認した。コメントは、次の構文をさしはさむことでコンソールへ出力を行うことができる

```
Log.debug("表示する文字");
```

この構文により、動作しているメソッドの情報、該当する Log.debug の行数等を付加した形で、出力を得ることができる。次に、その出力例を示す。

```
0:08:36.416 DEBUG:
```

```
org.geogebra.common.kernel.prover.ProverBotanasMethod.prove[1691]: --ボ  
タナメソッド回答が不完全 cannot reliably tell if the statement is really  
false
```

これは、動作時刻とデバッグメッセージで、「--」記号以降が、表示する文字列としてあらかじめ準備していたものである。ここでは、Log.debug が埋め込まれた場所である、ProverBotanasMethod において、行われた分岐処理の処理状況、コード内のコメント文を転記する形で出力したものである。

5.4.3 UI ダイアログの更新

最終的にユーザーに画面を提示する際には、UI の画面表示には、「しかし一般には正しくない」のような混乱をもたらすような表現ではなく、「この図においては正しい」という表現に置き換える必要がある。現状においては、まだ本当に「この図においては正しい」という表現が、「しかし一般には正しくない」と同等であるかは確認ができていない状況ではあるものの目的を指向し表現の更新を行ったうえで実験を継続することとした。日本語の表現が含まれる 2 つのファイルについて、書き換えを行ってレスポンスを確認したところ、

```
menu_ja.properties
```

の箇所の書き直しを行うことで、UI に表示される日本語表現が更新されることが確認できた。そこで、「しかし一般には正しくない」という出力を「この図においては正しい」という表現に置き換えて、実験を続けた。

5.4.4 コメント挿入で見えた Prover の振る舞い

コメントをさしはさむことで動作状態を確認した結果、Prover は以下の図 5-5 のような動作を行っているものと推測した。

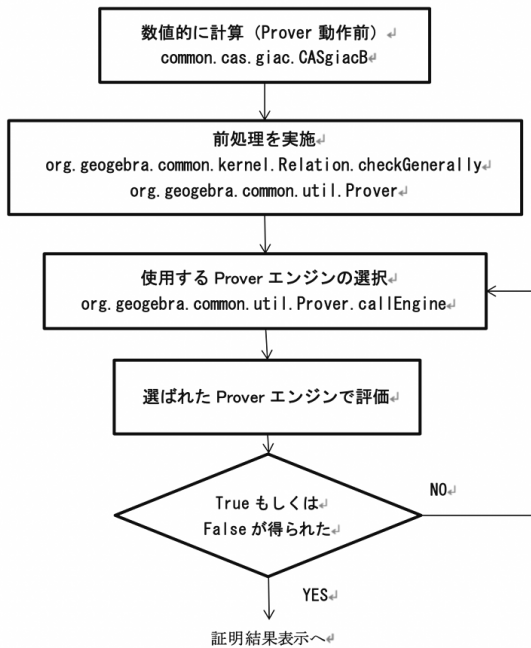


図 5-5. Prover の動作概略

Prover の核として動作するのは `Prover.java` であり、この Java コードを軸に、多数のクラスが配置されている。証明結果は、変数 `ProofResult` に格納されており、この変数名を検索することで、証明に関わるクラスを確認することができる。使用することができるエンジンは、`public enum ProverEngine` に列挙されており、起動順序を含めた設定は、`Prover.java` の `public Prover()` メソッドで行われている。標準の状態では、Prover のエンジンを自動選択するモードが動作するようになっており、`RECIOS_PROVER`、`BOTANAS_PROVER`、`OPENGEOPROVER_WU` の 3 つのエンジンが、この順番で用いられるように配列に格納される。

元の Java コードに記載されたコメント文等を参考に `Log.debug` に状況を示すためのメッセージを埋め込む拡張を済ませたプログラムにて、5.4.2 節で確認したものと同一構文を入力し、動作状況を試した。動作結果を表 5-6 に、そのとき

の画面上の更新事例を図 5-7 に示す。

表 5-6. Ver 5.0.366 における動作結果

構文	動作した Prover と順序	証明判定	状況
Relation[f, g]	Recio→終了 画面出力は f と g は平行である 「本図においては正しい」	False	Recio Prover にて 「STATEMENT IS FALSE」
Realaion[A, f]	Recio → Botana → 終了 画面出力は A は f 上にある 「常に正しい」	True	Recio Prover にて ステートメント取得に失敗 ↓ Botana Prover にて 「STATEMENT IS TRUE」
Relation[AB, CD]	Recio → Botana → OpenGeo(Wu)→終了 画面出力は Distance[A, B] と Distance[C, D] は等しい 「本図においては正しい」	?	Recio Prover にて ステートメント取得に失敗 ↓ Botana Prover にて 「statement is UNKNOWN」 ↓ OPENGEOPROVER_WU にて 「Failed in reading input geometry theorem」 ↓ 「STATEMENT IS FALSE」

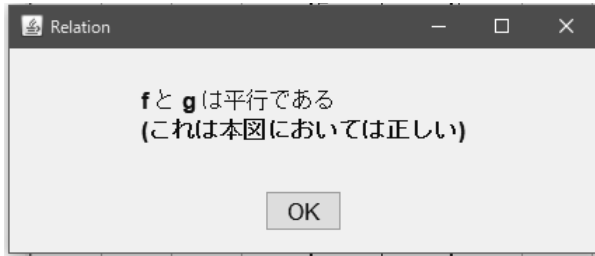


図 5-7. 画面表示の更新事例

今回実施した構文は、いずれも数値的チェックの段階では、「成り立つ」と判定されているものである。しかしながら、Relation[f, g]はRecio Prover が False 判定をした直後に処理を完了してしまい、「本図においては正しい」（当初の表記では「しかし一般には正しくない）」という出力となっている。Relation[AB, CD]については、Recio, Botana, OpenGeoProver Wu が動作したものの、Recio と OpenGeoProver は構文を処理することができず、読み込みの段階でエラーとなって処理が中断している。また Botana は動作したものの、判別不能となっており、最終的な判定結果は UNKNWN で推移し、最終的に False と判別されることになった。

以上の取り組みを通じて、Prover については、十分な内容分析ができているとは言い難い状況ではあるものの、全体としてみると、目下開発が行われている最中、という表現を用いるのが適切であると考えられる。具体的には、それぞれの Prover は、起動の呼び出し時等には独立したメソッドとして実装されているように見えるものの、実際の内部の動きを見てみると、Recio Prover から Botana Prover を呼び出すコードが存在する等、1つの研究チームが色々工夫を重ねて2つの方式を構築している、といった様子が伺える。Prover の起動順序はプログラマ的には変更可能であるものの、GeoGebra wiki では、「エンドユーザーは順番を変えるべきではない」と言及されており、今回確認できていない部分においても、Recio と Botana、両 Prover にはつながりがあることが予想される。また、理由は十分に検討することが出来なかったものの、出力を見る限り、動作の指定をしていなくても、証明の最後にもう一度 Botana Prover が動作した上で結果の出力がなされている。

このような Prover の状態を鑑みると、GeoGebra で証明問題を扱うためには、さらに Prover の動作状況を確認し、Prover を適切に動作させることができる点や線の定義方法について知見を深めた上で、どのような証明問題を解くことができるか、という部分に踏み込んでいく必要があるものと考えられる。他方、Prover が目下開発中であるという点からは、短期的には外部の証明機能を持つソフトウェアと連携する、あるいは、長期的には Prover の発展を待って取り組みを継続する、といったオプションも考えられる。

5.5 証明機能を利用する学習の枠組みの提供

5.5.1 2019年10月時点における Web アプリケーション版の証明機能

2019年10月の時点において、GeoGebra は Web アプリケーションとしての利用が主となっている。前節までで議論した証明学習について、GeoGebra が公式に提供している web アプリケーションによって我々の想定する学習環境が提供できるか、という点、現状では困難と言わざるを得ない。2019年10月現在、オフィシャルサイトで公開されている最新版 web アプリケーションにおいては、relation 関数や prove 関数を使うことができるようになっている。例えば、2つの平行で長さの異なる線分を作り、relation 関数の引数として、それぞれの線分を与えると、数値的にチェックした結果、平行である、といった出力を得ることができる。しかしながら、それ以上の説明を得ようとすると、標準の関数グラフでは、何も表示が行われぬ。GeoGebra Classic に切り替え、同様に2つの平行で長さの異なる線分を作り、relation 関数の引数として与えると、数値的にチェックした結果も、数値的に平行であるという表示が得られる。Classic においては、さらに詳細な表示を求めることができ、その場合は、以下の図 5-8 のように、「一般的に偽」という表示が得られる。

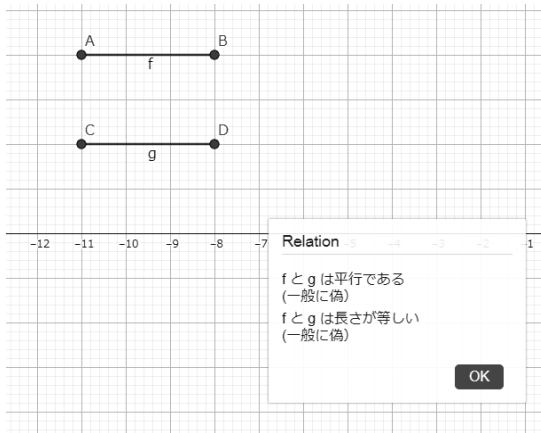


図 5-8. GeoGebra web アプリケーション (GeoGebra クラシック 6) における Relation 関数の適用結果

これらの振る舞いは、2017 年に調査を開始した時点からは変化がみられており、継続的な開発が行われていることが推測される。他方、2017 年の調査でソースコードをもとに動作状況を追跡・検討した段階においては、内部では様々な Prover が動作していることが確認できた。しかし、現状の web アプリケーションにおいては、どのような Prover がどのような処理結果を返したために「偽」という出力となったか、といった情報を得ることができない。よって、授業の実践という観点からは、最新の web アプリケーションが非常に高度な処理を行っていたとしても、それをユーザーにフィードバックするという動作が実現されていないため、証明学習で用いることが難しいという結論に至らざるを得ない。

5.5.2 証明学習を指向した学内向け GeoGebra の環境検討

5.5.1 節より、公式の web アプリケーションは利用が難しいことに変わりがないことから、証明学習を指向した学内向け GeoGebra の環境提供のためには、ソースコードに変更を加え、希望する機能を拡張した GeoGebra が必要である。この

ような拡張版 GeoGebra を大学学内で利用することを考えると、現実的な方策として、3つの方向性があるものと考えられる。

第一の方向性は、拡張版 GeoGebra を独自のサーバに組み込み、学内専用の web アプリケーションとして配信し、学習者がブラウザ経由で利用する、という形態である。このアプローチの最大のメリットは、多数の学生に対して円滑に学習環境を提供できる点である。GeoGebra のソースコードには、web 向けの機能が組み込まれていることは確認できる。よって、拡張を済ませたプログラム一式を war ファイル等の形式にまとめ、Java Servlet を処理できるよう整備した web サーバにデプロイすることができれば、動作が実現できる可能性がある。しかし、そもそも、GeoGebra をビルドして動作させる、というだけでも難易度が高く、とくに、web 上には、web アプリケーションとして GeoGebra を構成するための情報といったものはほとんど公開されておらず、手探りで実現するには、非常に難易度が高いものになる可能性が高い。現在試行的に動作を試みているものの、2019年9月の段階では、まだ動作の目途は立っていない。

第二の方向性は、学内のコンピュータにて、直接拡張版 GeoGebra を動作させる、という方策である。ただし、現状では拡張版の GeoGebra を単体の Java アプリケーションとして動作させることが困難であるため、開発環境である Eclipse とソースコードのセットとして配布し、学習者がそれぞれの PC 上で動作している Eclipse にて拡張版 GeoGebra を起動させる、というものである。甲南大学のパソコン教室のコンピュータには、Eclipse が導入されており、ソースコードを組み込んで、動作の準備を整えた Eclipse のワークスペースフォルダをまるごと配布することで、方策の実現ができる可能性がある。しかし、実際にワークスペースの配布を実施してみると、公開されているソースコードをそのまま利用することは難しい状態であった。公開されている GeoGebra のソースコードは、自動ビルドシステムである Gradle を利用しており、その中で、バージョン管理システム Subversion (SVN) を利用している。このため、福井ら [30] によって確認された方法でビルドを行った場合、これらの管理システムで設定された条件でビルドが行われることになる。しかし、この場合は、ワークスペースに必要なファイルがすべてそろっているのではなく、Gradle が管理しているキャッシュ領域にビルドに用いるソースコードが散在する状態である。このキャッシュ領域には、ソースコードとして配布されている Geogebra に大きく寄与するプログラムが複数展開されている。これらは Gradle を経由した SVN によってオンライ

ンのソースコードから同期されたものである。このような状態であるため、Eclipse のワークスペースを単体で配布するだけでは動作をさせることができず、動作させるコンピュータに対し、Eclipse とは別個に Gradle をインストールした上でソースコードを同期させておく、という手順が追加が必要となる。これらの手順は非常に煩雑であり、実験用のコンピュータを有する研究室であれば、学習者用に割り当てた PC に対して、相応の時間をかけて準備をすることで動作環境の構築が可能である。一方、甲南大学のパソコン教室にて動作をさせる環境を整えることを考えると、複数の追加ソフトのインストールが必要なうえ、Windows のシステムパス等のパラメータにも変更を加える必要があることから、試験的な導入のハードルは高いと言わざるを得ない。

第三の方向性は、拡張版 GeoGebra をビルドして動作させるための環境そのものを、学習者に提供する、という方策である。すなわち、VMWare や VirtualBox として知られる、コンピュータの仮想化ソフトウェアを活用し、Linux 上で整備した Eclipse による開発環境と拡張版 GeoGebra を、丸ごと学習者に提供する、というアプローチである。この方策については、第二の方向性で検討した際に問題となった、パソコン教室での利用について条件が緩和できることが期待される。甲南大学のパソコン教室には仮想化ソフトウェアとして VirtualBox が導入されているため、配布したオリジナルの Linux イメージを読み込むことができるような設定の変更が実現できれば、学習者が仮想化した Linux 上で試行が可能となる。

検討の結果、少人数の学習者による実証実験から研究を展開していくことを考えると、第二の方向性、すなわち Windows 版の動作環境構築を、準備した自由に設定変更ができるコンピュータ (数台) にセットアップする、という方法が最も現実的であると考えられる。次のステップとして複数の学生を対象とした授業形式での試行において取り組むべき方策と考えられるのが、仮想化ソフトウェアを利用したパソコン教室での動作整備となる。どのような方法でどの程度のコンピュータに実行環境を提供するかについては、研究プロジェクト全体の方向性を考慮したうえで、今後の実証に向けて継続的に取り組まなくてはならない事項である。

5.8 まとめ

本研究では、証明学習で目指す例題を提示し、証明学習に対応する拡張の検討、

および拡張を行った GeoGebra を学習者に提供する方法について検討した。様々な試行錯誤を通じて主体的に学ぶ、という観点からは、拡張済みの GeoGebra を学習者が納得ゆくまで使い倒すという体験を提供することはたいへん重要であり、コンピュータを利用した学習支援の新しい形として、新規性と有用性が訴求できるものと考えられる。一方、その実現に向けたプログラムの研究開発については、本研究の取り組みの本質的な部分として、研究開発が行われている最中のプログラムに対してアプローチをかけている部分もあり、十分にキャッチアップができていない箇所がみられる状態である。それでも、ソースコードのビルドが可能になったことで、Prover の発展中の状況を確認することができ、また、画面表示内容を本研究の主旨に合わせた学習体験に適する形に拡張することができた。しかし、その拡張版の GeoGebra を学習者に提供する段取りについてはいまだ多くの課題を残している。目下その学習体験のデザインを含めて取り組みを継続しており、少人数をターゲットにした小回りの利く授業の中で、試行を開始し、実証を重ねていきたいと考えている。他方、証明学習の発展という意味では、Prover 機能の状態にあわせてレスポンスを変えることについて、さらに拡張の余地が残されている。プログラムへの理解を深め、学習者の体験の質の向上につなげるというアプローチも重要であると考えられる。

参考文献

- [1] <https://ja.wikipedia.org/wiki/自動定理証明>.
- [2] Tadashi Takahashi: The Gap of Complete Proof and Incomplete Proof in Mathematics Education using Technology, Proc. of the 14th International Conference Challenges in Mathematics Education for Next Decade, pp. 328-330, 2017.
- [3] G. Polya: How to Solve It, A New Aspect of Mathematical Method, Princeton Science Library, 1954.
- [4] THEOREMA: On [https://www.risc.jku.at/research/theorema/software/documentation/tutorial /TheTheoremaSystem.html](https://www.risc.jku.at/research/theorema/software/documentation/tutorial/TheTheoremaSystem.html)
- [5] Tadashi Takahashi: Teaching and Learning by using GeoGebra, Proc. of the 8th International Conference on Innovations in Engineering, Technology, Computers and Applied Sciences (IETCAS-2017), pp. 129-

136, 2017.

- [6] Brunswik E. : Representative design and probabilistic theory in a functional psychology. *Psychological Review* 62, 1955.
- [7] Card S., Moran T. P. and Newell A: *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.
- [8] Clancey W. : The frame of reference problem in the design of intelligent machines. *Architectures for intelligence: The 22nd Carnegie symposium on cognition*. Lawrence Erlbaum Associates, 1991.
- [9] Duncker K. : *On problem solving*. Psychol Monograph, 1945.
- [10] Ichikawa S. : *Psychology of Learning and Education*, Iwanami Shoten, 91, 1995 (in Japanese).
- [11] Kahneman D. and Tversky D. : Prospect theory. An analysis of decision under risk. *Econometrica*, 47(2), 1979.
- [12] Kobayashi H, Suzuki H and Ono Y. : Formalization of Henzel' s Lemma, 18th International Conference, Ls, Oxford, UK, *Emerging Trends Proceedings Oxford Research Report*, 2005.
- [13] Kozuya T. (ed.) : *Memory and Knowledge (Cognitive Psychology Lecture 2)*, University of Tokyo Press, 17, 1978 (in Japanese).
- [14] Neisser U. : *Cognitive psychology*. Prentice-Hall, 1967.
- [15] Newell A. and Simon H. A. : *Human problem solving*. Prentice-Hall, 1972.
- [16] Newell A. : The knowledge level. *Artificial Intelligence*, 18, 1982.
- [17] Polson P. G. and Kieras D. E. : A Quantitative Model of the Learning and Performance of Text Editing Knowledge, *Proceedings of ACM CHI' 85 Conference on Human Factors in Computing Systems*, 1985.
- [18] Polya G. : *How to solve it*. Doubleday, 1957.
- [19] Rasmussen J. : *Information Processing and Human-Machine Interaction*, Elsevier Science Publishing, 1986.
- [20] Rasmussen J. : Recognition engineering of interface, *Keigakushuppan*, 1990 (in Japanese).
- [21] Skinner B.F. : *Science and human behaviour*. FreePress New York, 1953.
- [22] Tamura H. : *Human interface*. Ohm-sha, 1998 (in Japanese).

- [23] Tulving E.: Organization of memory, Quo vadis? The cognitive neuro-science, MIT Press, 1994.
- [24] Wallas G.: The art of thought. Harcourt, 1926.
- [25] Wiener N.: Cybernetics. John-Wiley & Sons, 1948.
- [26] <https://www.geogebra.org>.
- [27] 岩間 詞也, 高橋 正: GeoGebra のビルドについて, 統計数理研究所共同研究レポート 391, 動的幾何学ソフトウェア GeoGebra の整備と普及(2), pp. 62-64, 2017.
- [28] 篠田有史, 高橋 正: GeoGebra の Proving に求めるものとその機能について, 統計数理研究所共同研究レポート 396, 動的幾何学ソフトウェア GeoGebra の整備と普及(3), pp. 19-26, 2018.
- [29] 篠田有史, 高橋 正: 証明学習を志向した学内向け GeoGebra 環境の整備 統計数理研究所共同研究レポート 426, 動的幾何学ソフトウェア GeoGebra の整備と普及(4), pp. 6-9, 2019.
- [30] 岩間詞也, 福井昌則: GeoGebra のビルドと現状での証明機能の限界について, 甲南大学紀要 知能情報学編, Vol. 9, No. 2, pp.143-159, 2016.
- [31] 柿内賢信 訳 G. ポリア: “いかにして問題をとくか 第 11 版”, 丸善出版, 2011.
- [32] GeoGebra 数学アプリ: <https://www.geogebra.org>, 2019 年 2 月 23 日 閲覧

研究者紹介（執筆順）

高 橋 正 （甲南大学知能情報学部教授）

篠 田 有 史 （甲南大学共通教育センター准教授）

2020年(令和2年)2月28日 発行

甲南大学総合研究所

神戸市東灘区岡本8丁目9番1号（〒658-8501）

（非売品）