Understanding the Problem Structure Using Modified Problem-Posing in Mathematics Classrooms

Masanori Fukui
f-masanori@tokushima-u.ac.jp
Center for University Education
Tokushima University
1-1 Minamijosanjima, Tokushima
Japan

Yuji Sasaki
y.sasaki@keio.jp
Graduate School of Media and Governance
Keio University
5322 Endo, Fujisawa, Kanagawa
Japan

Abstract: *Understanding the structure of problems is essential in mathematics education. Especially in structured problems, such as those in mathematics, it is crucial to understand the problem structure before solving it to improve the solution method. It has been reported that problem-making, or problem-posing, is effective in promoting creativity and understanding problem structures. However, it is not always easy for students to create problems. Thus, we propose "modified problem-posing" as a method to help students solve problems. Easier than using conventional problem-posing, it involves transforming/modifying a problem, which is more likely to help students create problems and explain them in an understandable way. Expected to be developed in the future, this method also relates to computational thinking. This study proposes a methodology to enhance–the understanding of problem structure by focusing on modified problem-posing.*

1. Introduction

Understanding the structure of a problem is essential in mathematics education. Mathematics deals with structured problems; its structure and openness can be written in a concrete order. Polya [1] states that he gave a four-step method for solving mathematical problems: "1. First, you have to understand the problem," "2. After understanding, then make a plan," "3. Carry out the plan," and "4. Look back on your work. How could it be better?" Therefore, it is essential to consider ways to promote understanding the problem structure in mathematics education.

One way to promote such understanding is through problem-posing, an activity in which learners create their own questions. Problem-posing involves the generation of new problems about a situation, or the reformulation of a given problem [2,3]. Cai and Hwang [4] propose that problem-posing in mathematics education is related to several types of activities that entail or support teachers and students formulating or reformulating and expressing a problem or task based on a specific context, for example, as the problem context or problem situation. It has been pointed out that problem-posing involves thinking about the logical structure of a problem by doing the opposite of general learning (i.e., answering given problems) [5]. Einstein and Infeld [6] state that the formulation of a problem is often more essential than its solution and that creating problems is a very important activity, not only in mathematics education but also in other fields. Therefore, activities that help students understand the structure of the problem by learning to write questions are important in mathematics education and are believed to be essential for improving problem-solving skills.

Mestre [7] points out that the composition of first-time students lacks diversity, although it is important for them to conduct activities to promote understanding of the problem structure by learning to write problems. Therefore, it is necessary to develop a method to enable many students to learn by asking questions. Although it is expected that the question-making process promotes understanding of the problem structure, it may be difficult for some students to explain how they composed the questions, depending on the problem and their learning stage. Therefore, it is important to establish an activity that allows students to explain the purpose of creating a question to promote their understanding of the problem structure.

Therefore, we propose a “modified problem-posing” in which a given problem is transformed or modified to form a problem. The modified problem-posing is a learning method through which many students aim to create a problem by transforming a well-structured problem with a clear structure and goal. By extracting parts of a well-structured problem and deforming them, the problem can be clearly explained. In other words, it is easy to grasp that the deformable part “A” in the problem has been changed to “B,” and the same idea can be applied to other problems. In addition, students who transform the same problem in class can understand the content of another student’s problem. Therefore, we believe that “modified problem-posing” using good-structure problems is a promising method for promoting understanding of problem structure in mathematics education.

In this study, we describe the proposed method of modified problem-posing and show an example of its application.

2. Enhancing the Understanding of Problem Structure with Modified Problem-posing in Mathematics Classrooms

2.1 Modified Problem-posing and its Example

We focus on problem-posing as a learning method to promote an understanding of the structure of problems. Problem-posing learning is learning to create problems, that is meta-problems. Problem-posing improves learners’ problem-solving skills and is a measure of understanding, and its effectiveness is widely recognized [8]. A relationship exists between problem-posing and creativity [9] and problem-posing is related to understanding the subject matter [10]. In other words, it is expected that problem-posing will help students develop their understanding of existing mathematics learning and improve their problem-solving skills. However, Mestre [7] states that the composition questions of beginning students lack variety. In addition, it is often difficult for students who have never created a problem before setting up a task to solve it by themselves. Therefore, developing a method that allows students to learn problem-solving with various problems is crucial.

Polya [1] states that changing and creating new problems is vital for understanding and solving them. Wallas [11] also states that transforming problems fosters creativity; thus, focusing on modifying the problem may be effective.

One of the authors has developed a model for fostering creativity by transforming a well-structured problem with a clear structure and goal to enable each student to create a new problem [12]. This method focuses on the good structure problem used in existing education and is expected to promote understanding of the problem by changing the way it is viewed.

In addition, we are developing a method to evaluate the difference between the original problem and the transformed problem by rotating the original problem and the transformed problem using an “Activity Diagram” from the “Unified Modeling Language (UML).” Gogolla [13] states that UML is a graphical language for visualizing, specifying, constructing, and documenting software-related artifacts and provides a standard method for presenting system configurations, ranging from conceptual (such as business processes and system functions) to concrete (such as programming language statements). Besides, UML is not a programming language, but a visual one. OMG [14] defines thirteen types of diagrams, divided into three categories: six types of structure diagrams, three types of behavior diagrams, and four types of interaction diagrams. Activity diagrams, one of the Behavior diagrams, are used to think about how the various workflows in a system are constructed, how they are initiated, and what paths they follow from start to finish [14]. An activity diagram can clearly represent a well-structured problem, identify the problem transformations, and clearly show how the problem was transformed. It is also possible to describe how the problem transformation affects the given problem. If students understand the problem structure, they are expected to understand the problem in a way that is like an activity diagram. Also, the teacher can evaluate the student's problem transformation.

An example of modified problem-posing is provided using the "Collatz conjecture," a simple and well-structured problem in mathematics that can be described in an activity diagram. Understanding the problem structure does not require prior knowledge.

Example 2.1 (Collatz Conjecture, 3n+1 Problem)
For any natural number n > 1, we apply the following rules repeatedly.

(1) If n is even, divide it by two.
(2) If n is odd, triple it, and add one.
(3) Repeat the calculation, and when n becomes 1, the calculation is completed.

To make modified problem-posing easier to understand, we conveyed the problem as an activity diagram of a unified modeling language, as shown in **Figure 2.1.** If the learner can divide the problem and understand the game's flow, they can draw the structure of the problem, as shown in **Figure 2.1.** In addition, modification focusing on a part or balance of the whole is possible. Examples of the modifications are shown in **Table 2.1.** For example, in **Table 2.1**, focusing on the initial values, it is possible to restrict the range of n or change n to a real number r. Next, the branching part can be changed from even or odd to multiples of three or other. Furthermore, the computation part after the branch can be changed by dividing n by two or triple n and adding one to divide n by three or double n.

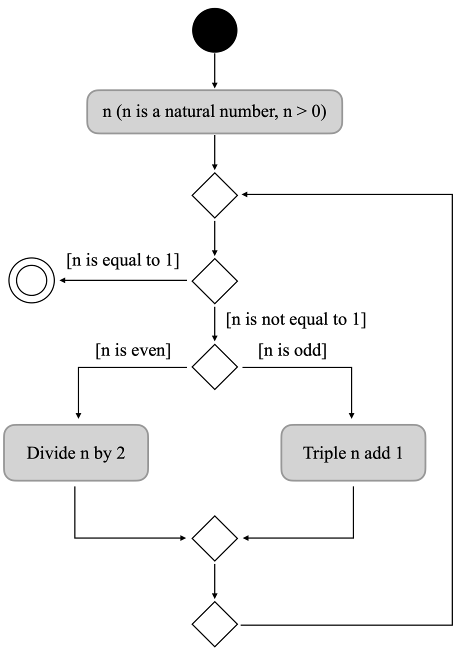

Figure 2.1 Activity Diagram of Collatz Conjecture's Structure [12]

Table 2.1 Examples of Modification

Focus point	Example of variant
Value n in natural number	To restrict the range of n Change n to a real number
Branching by even or odd numbers	Branching by multiples of three or other
Processing part after branching, divide n by two or triple n and add one	Processing part after branching, divide n by three or double n
Repeat until finished	Repeat three times

Thus, the problem can be understood as having a "skeleton" structure as shown in **Figure 2.2**. In addition to changes to the [ ] section and to the contents of the square, it is expected that additions and changes will be made to the dotted line. Once the structure is understood, it is easy to transform the problem. Students can modify different parts of the problem from the previous one, understand the problems that others have transformed, and modify multiple parts of the problem.

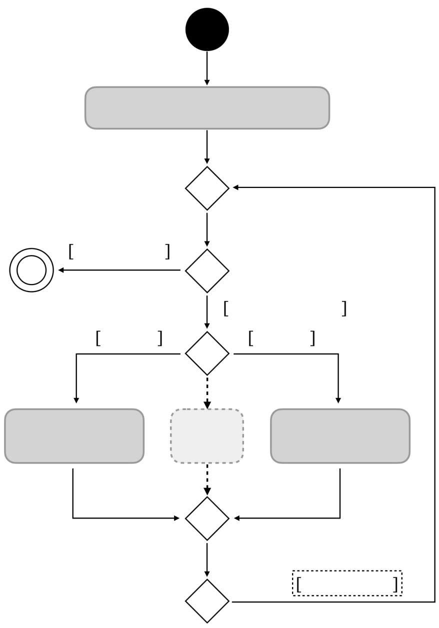

Figure 2.2 Activity Diagram of Collatz Conjecture's Skeleton Structure

To modify the problem while focusing on the parts, we must understand the structure of the problem. Students can explain the reason for the modification. In addition, it encourages students to think about the overall effect when a specific part is changed. In this way, modified problem-posing can deepen the understanding of the tasks used in existing education.

2.2 Relationship between Modified Problem-posing and Computational Thinking

In addition, one of the authors points out the connection between problem transformation and computational thinking.

Computational thinking is the process of organizing and expressing problems in a form that computers can solve [15] and is essential in understanding the structure of a problem. Google for

Education has developed a series of tools for the psychological process: abstraction, algorithm design, decomposition, pattern recognition, automation, and data representation as tangible outcomes. It defines abstraction, algorithm design, decomposition, and pattern recognition as psychological processes and automation, data representation, and pattern generalization as tangible outcomes, which are considered relevant to computer-aided problem-solving [16]. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have also provided operational definitions of computational thinking [17]. Although much research has been conducted on the topic, a need exists for the development of models that embed computational thinking into the actual content used in mathematics education, to promote an understanding of the structure of problems.

The four concepts of computational thinking, i.e., abstraction, algorithm design, decomposition, and pattern recognition, are considered to play an important role in understanding the problem structure. An explanation that demonstrates a good understanding of the problem structure provides an appropriate explanation of which part of the problem was transformed. In addition, it offers a situation in which students can understand the problem structure appropriately by acquiring general-purpose transformations that can be applied to other problems, rather than transformations that can be employed only in the presented problem. This suggests that, for example, the development of the ability to find deformable parts of a given problem, find common parts of problem deformations, learn methods of problem deformation that can be applied to other problems, and explain the problem structure in an orderly manner, play an important role in understanding the problem structure. These are, in turn, considered to correspond to computational thinking's decomposition, pattern recognition, abstraction, and algorithm design (algorithmic thinking). By having students modify problems based on computational thinking, teachers expect them to gain a deeper understanding of the structure of the problem, allowing them to make changes based on evidence and acquire skills that can be applied to other problems. In the modified problem-posing method, decomposition focuses on the parts of the problem; algorithmic thinking is associated with understanding the flow and relationships of the entire problem. Generalization is related to identifying patterns in the problem, and abstraction is associated with applying the problem to other problems. Therefore, modified problem-posing can be considered a better activity that focuses on decomposition, algorithmic thinking, generalization, and abstraction of computational thinking. The relationship between modified problem-posing and computational thinking is shown in **Table 2.2.**

Table 2.2 Relationship between Computational Thinking and Modified Problem-posing [18]

Concepts of computational thinking	Examples of activity
Decomposition	Find the changeable part of the problem
Generalization	Find a common point for changeable parts of the problem
Abstraction	Find a common point that can be used in other problems
Algorithmic thinking	Accurately represent the flow of the problem

2.3 Design of Activities in Mathematics Classrooms Using Modified Problem-posing

The relationship between modified problem-posing and computational thinking is discussed in Section 2.2. Modified problem-posing is adequate for understanding problem structure; however, its practice is not advanced. In this section, we develop a practice of modified problem-posing to enhance the ability to understand the problem structure. This flow is presented in **Table 2.3**.

Table 2.3 A Flow of Activity using Modified Problem-posing

Task	Contents
1	The students are asked to solve a given problem. In addition, an activity diagram corresponding to the original problem is presented from the teacher's side.
2	Have the students find as many changeable parts of the problem as possible.
3	Let the students choose some possible changes and change, add, or delete the selected parts. Then, ask students to consider the correspondence between the transformed problem and the UML.
4	Have the students write a UML of the transformed problem.
5	Have the students solve the transformed problem.
6	Have students consider how the transformed problem affects the main problem.

These activities include algorithmic thinking by considering the correspondence between the problem and the activity diagram, decomposition, and generalization or pattern recognition by examining commonalities and differences by decomposing the problem into its elements. The students can then compare the given problem and the transformed problem and follow the activities. By looking at commonalities and differences, we can expect to develop abstraction through decomposition and generalization or pattern recognition and by thinking about what is necessary to apply it to the next activity.

When focusing only on enhancing problem-solving skills, it is difficult to think deeply about how to capture the difference between the original problem and the transformed problem. However, considering that understanding the problem structure is an important first phase, the activities described above are useful to focus on helping the students understand the problem structure. The computer algebra system or programming language also makes it possible to compute structured problems such as sequences, gradual equations, and puzzle problems and examine whether the predicted results are correct.

Based on **Table 2.3,** these activities are designed to find a connection between the activity diagram and the problem, and to encourage the participants to think about the difference between the activity diagram and the transformed problem to identify the problem structure. Writing an activity diagram helps identify what needs to be changed. Or, understanding the problem structure facilitates creating an appropriate activity diagram.

Based on the above, it is expected that activities will focus on helping students understand the problem structure.

3. Summary and Future Work

In this paper, we proposed a modified problem-posing method to promote understanding of the problem structure. This method offers a way to reassess the subject matter used in mathematics education from the perspective of understanding the problem structure and enhancing computational thinking. Students can also easily create new problems, and teachers can guide and evaluate them.

This method's quantitative effectiveness must be evaluated and its future usefulness studied. Also, a model that can be applied to a broader range of fields must be developed, for example, by examining practical examples in other subjects.

Note. This study was based on the unpublished works of the Authors with many additions and corrections [12, 18].

Acknowledgments
This work was partially supported by 2022 Inamori Research Grants given by Inamori Foundation and JSPS KAKENHI (Grant Number 21K13644). We are grateful to Prof. Dr. Tsukasa Hirashima (Hiroshima University) for insightful discussions.

References

[1] Polya, G. (1990). *How to Solve It: A New Aspect of Mathematical Method*, Penguin, New York.
[2] English, L. D. (1997). The development of fifth-grade children's problem-posing abilities, *Educational Studies in Mathematics*, **34**, 183–217. https://doi.org/10.1023/A:1002963618035
[3] Silver, E. & Cai, J. (1996). An analysis of arithmetic problem posing by middle school students, *Journal for Research in Mathematics Education*, **27**, https://doi.org/10.2307/749846.
[4] Cai, J. & Hwang, S. (2002). Generalised and generative thinking in US and Chinese students' mathematical problem solving and problem posing, *The Journal of Mathematical Behavior*, **21**, no. 4, 401–421.
[5] Kojima, K., Miwa, K., & Matsui, T. (2010). An experimental study on support for leaning of problem posing as a production task, *Transactions of Japanese Society for Information and Systems in Education*, **27**, no. 4, 302–315. [in Japanese]
[6] Einstein, A. & Infeld, L. (1938). *The Evolution of Physics*, Cambridge University Press, Cambridge.
[7] Mestre, J. P. (2002). Probing adults' conceptual understanding and transfer of learning via problem posing, *Journal of Applied Developmental Psychology*, **23**, 9–50.
[8] Silver, E. A. (1994). On mathematical problem posing. *For the Learning of Mathematics*, **14**, no. 1, 19–28.
[9] Van Harpen, X. Y., & Sriraman, B. (2013). Creativity and mathematical problem posing: an analysis of high school students' mathematical problem posing in China and the USA. *Educational Studies in Mathematics*, **82**, no. 2, 201–221.
[10] Brown, S. I. & Walter, M. I. (1983). *The Art of Problem Posing*, Franklin Institute Press, Philadelphia.
[11] Wallas, G. (1926). *The Art of Thought*, Jonathan Cape, London.
[12] Fukui, M., Kashiwagi, M., Miyadera, R., Hagikura, J., Sasaki, Y. & Hirashima, T. (2019). Proposal of creativity evaluation method using well-structured problems, *International Conference on Computational Creativity 2019 (ICCC 2019)*. [unpublished]
[13] Gogolla, M. (2009). Unified modeling language, in Liu, L. & Özsu, M. T. (eds.), *Encyclopedia of Database Systems*, Springer, Boston. https://doi.org/10.1007/978-0-387-39940-9_440
[14] Object Management Group (2007). *Object Management Group Unified Modeling Language Specification*. https://www.uml.org/
[15] Wing, J. M. (2006). Computational thinking, *Communications of the ACM*, **49**, no. 3, 33–35.
[16] Google for Education (n.d.). *Exploring Computational Thinking: CT Overview*.

https://edu.google.com/resources/programs/exploring-ComputationalThinking/#!ct-overview
[17] International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) (2011). *Operational Definition of Computational Thinking for K-12 Education*. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf
[18] Fukui, M. (2021). A proposal of a method to promote creativity based on computational thinking in a classroom, *European Alliance for Innovation International Conference on Social Science (EAI-ICONESS 2021)*. [unpublished]