Teacher actions to promote mathematical competencies in programming activities

Runar L. Berge[1*], Ingeborg L. Berget[1], Andreas Brandsæter[1,2] and Beate K. Humberset[1]

[1]Volda University College, Faculty of Humanities and Education, Volda, Norway;
[2]NTNU, Faculty of Information Technology and Electrical Engineering, Ålesund, Norway;
[*]runar.lie.berge@hivolda.no

The use of mathematical competencies in education has gained considerable attention, especially over the last two decades. This paper outlines a study that explores the integration of programming used to enhance students' engagement with mathematical competencies. The study introduces a teaching session wherein students program physical robots, with a primary emphasis on generalizing the programs of the robots using mathematics. Our results show the importance of the teacher to ask questions that focus the discussion on mathematics, and we demonstrate how removing the robots prompts the students to communicate their ideas and reasoning. Our conclusion asserts that teaching sessions which emphasize mathematical competencies within the framework of programmable robots hold the potential to foster student engagement with respect to these mathematical competencies.

Keywords: Mathematical competencies, Mathematics education, programming, robots.

Introduction

Over the past decade, programming has been introduced into education in many countries (Balanskat & Engelhardt, 2014). In the Nordic countries, programming is mainly introduced as an integrated part of traditional school subjects (Bocconi et al., 2018). In the Norwegian curriculum, the main responsibility for teaching programming is placed in the subject of mathematics, starting in primary school (Kunnskapsdepartementet, 2020). It is specified in the mathematics curriculum that the students shall learn programming, but its relation to mathematics is not explicit, and the practical implementation is left to the teachers. Thus, knowledge about good teaching practices when teaching programming in mathematics is needed.

Many studies have established the connection between programming, computational thinking and algorithmic thinking skills (Weintrop et al., 2016), and studies investigate how mathematical topics can be learned by using programming (Misfeldt & Ejsing-Duun, 2015). The literature review by Forsström & Kaufmann (2018) identify three themes for the justification of programming in mathematics: motivation, student performance in mathematics, and the collaboration between students and changed role of the teacher. In this paper our focus is slightly different. Our main interest is on developing the mathematical competencies of the students, and programming is the setting in which we choose to do so.

There exist many theoretical frameworks for understanding the core of what mathematics is. Niss and Højgaard (2019) define eight mathematical competencies in the KOM-framework. The definitions of these competencies are made to capture the essence of mathematics, independent of the level of the person doing mathematics. These competencies were important in the creation of the framework for the PISA survey and Turner et.al. (2013) have created operational definitions of the competencies to analyse the difficulty of PISA items. We can also find these ideas in the Norwegian mathematics curriculum, described as the "core-competencies" of mathematics: exploration and problem solving, modelling and applications, reasoning and argumentation, representation and communication, abstraction and generalization, and mathematical fields of knowledge (Kunnskapsdepartementet, 2020) (The Norwegian curriculum uses the term "core-elements", but we have decided to use the term core-competencies).

In this paper we describe a teaching session developed in the research project "Programming for developing mathematical competencies". In this project, the teaching sessions are developed in an iterative design process in collaboration between researchers and teachers, and the teaching sessions are tested in classroom experiments. In previous work, five teacher actions that promote mathematical competencies are suggested based on this design process (Brandsæter & Berge, 2024), and in the current paper we focus on three of the suggested teacher actions: (1) Simplify needed syntax and remove non-mathematical elements, (2) disable the run-command, and (3) remove obvious representations and open up problem to new representations, and ask the research question: *How are students exposed to mathematical competencies when a teacher implements actions designed to promote students' work on mathematical competencies?*

Theoretical framework

In the analysis we use part of the theoretical framework developed by Brandsæter and Berge (2024), which is mainly based on the MEG item-difficulty coding framework (Turner et.al., 2013), the competencies in the KOM-framework (Niss & Højgaard, 2019), and the Norwegian curriculum, see Table 1. This study is limited to three of the core competencies.

Table 1: Framework for analysing work on mathematical competencies (Brandsæter & Berge, 2024)

Competence	Indication
Exploration and problem solving	a) Search for patterns and relationships (Norwegian Ministry of Education and Research, 2019, p. 2). b) "[S]electing or devising, as well as implementing, a mathematical strategy to solve problems arising from the task or context" (Turner et al., 2013). c) Use problem solving strategies (Pólya, 1990. d) Develop a method in an unknown situation (Eriksen & Vos, 2022).
Reasoning and argumentation	a) Present a "logically rooted thought processes that explore and link problem elements so as to make inferences from them" (Turner et al., 2013). b) Check a justification that is given (Turner et al., 2013). c) Provide a justification of statements (Turner et al., 2013). d) Make claims and conjectures.
Representation and communication	a) Interpret representations (Turner et al., 2013). b) Translate between representations (Turner et al., 2013). c) Make use of given representations (Turner et al., 2013). d) Select or devise representations to capture the situation or to present one's work (Turner et al., 2013). e) Evaluate the choice of representation (Niss & Højgaard, 2019).

Methods

We study the implementation of a teaching session where the student group was In-service Teacher Students (ITS) attending a course in programming and digital tools for mathematics teachers, grade 1-10, with little to no experience with programming before taking this course. The ITSs have a teacher education from higher education, but not necessary a specialization in mathematics. The session took 2x45 minutes over two days and the ITSs were randomly divided into groups of two or three ITSs.

The teaching session is built around the Sphero Indi robot, a small electric toy car that can be given instructions by placing coloured tiles on the ground. When the robot drives over a coloured tile it reads the colour and performs the corresponding instruction. Thus, the programming of the robot is done by placing tiles on the ground and the physical placement of the tiles defines the program. This allows for an intuitive programming of the robots, mostly eliminating syntax errors. We say mostly, because inaccuracy in the placement of the tiles or the initial orientation of the robot do sometimes cause the program to not run as intended (the robot misses one of the tiles causing the program to "crash"). We have chosen this robot due to its simplicity in use, making it easy to get started and the syntax (coloured tiles) is intuitive for any age group in schools. The teaching session is divided into three tasks, and the goal for each task is to elicit different mathematical competencies. The teaching session starts by an introduction of the robots that shows how to start the robot by placing it on a green tile, and that it stops when it drives over a red tile.

Task 1: In the first task, the ITSs are given a green, a red and a yellow tile, and are asked "what does the yellow tile do?". The task is meant as an introduction to the robots and how they are programmed, and it may at first glance seem like a trivial task. The yellow tile reduces the speed of the car, however, the speed difference between the initial speed and the speed set after driving over a yellow tile is quite small and not easy to see with bare eyes. This task targets the core competence *exploration and problem solving*, and it aims for the ITSs to use a systematic exploration of the yellow tile and to use mathematical relationships between, speed, distance, and time to design an experiment to test their hypothesis.

Task 2: The second task given to the ITSs is to "create a track that makes the robot drive on for as long as possible". One solution to this task involves creating a loop that makes the robot run infinitely long. Getting the idea of creating a loop is not the challenging part of this task, as we will see later. The main focus of this task is that the ITSs, while cooperating in the groups to design a track, get experience and skills relevant for developing *representation and communication*.

Task 3: In the final task, the ITSs study different loops that use different number of tiles and investigate how this relates to the sum of the robots' rotations. In collaboration with the teacher educator, the ITSs shall formulate and investigate the claim: "In a loop, the sum of the robot's rotation is always a multiple of 360 degrees". The task aims to give the ITSs experience with conjecturing, making statements and justifying about something that is always true, focusing on the core competencies *reasoning and argumentation*.

One of the authors of the paper was the teacher educator in the session, while the three other authors attended as observers. During the group work, each observer observed one group of ITSs and wrote a log of his/her observations. The logs include both descriptive and reflexive notes. The collected data from the three observers are combined and synchronized based on time stamps, allowing us to compare situations from the different groups. This is used to identify episodes where the teacher educator took an action to direct the ITSs work towards the mathematical competencies aimed for in the task.

The collected data is analysed qualitatively by identifying episodes where the ITSs are working on the core competence of the given task. We indicated (sub-)competencies in square brackets, in the results, based on the analytical framework in Table 1, e.g., [3a] indicate interpreting a representation.

Results and discussion

Simplify needed syntax and remove non-mathematical elements

All tasks involve elements where the ITSs can explore both the robots and mathematics. Nevertheless, the connection to mathematics was not always apparent to the ITSs, and in several instances the connections to mathematics were only made after direct questions from the teacher educator. This is exemplified through a sequence where the ITSs are working on the first task: “What does the yellow tile do?”. In this task the teacher educator *simplified the syntax* by removing all tiles but the yellow (slow down), green (start) and red tile (stop) (the full set consisted of 8 different colours). During the group work, the three ITS groups that were observed took slightly different approaches to the given problem. The first group explored the yellow tile in an unsystematic way, the second group tried systematically the different colours and combinations of colours [1c]. Both groups concluded that all the yellow tile did was to make a sound and flash yellow, even if they at some point in their discussion mentioned that perhaps the robot also decreased its speed. The third group first made a hypothesis that they proceeded to test by using the robot. By doing multiple runs with and without the yellow tile, they concluded that the robot slowed down, based on observations with their naked eyes.

When the teacher educator gathered all groups for a joint discussion, he first asked the ITSs if they knew what the yellow tile does. The first ITS to respond said that it makes a sound, but when the second ITS said that he thought that it slows down, the first ITS agreed but was not sure. The discussion continued as follows:

Teacher educator:	Are you sure that the tile makes the car slow down? [To ITS 2]
ITS 2:	Mmmm, I think so.
Teacher educator:	What can we do to figure that out? How can we be sure?
ITS 3:	With a stopwatch?
Teacher educator:	With a stopwatch! Others?
ITS 4:	Test with the tiles?

In the last suggestion the ITS proposed to compare one robot that runs over a yellow tile with one robot that does not run over a yellow tile [1d]. This involves a direct comparison of the speed of the robots to test if the yellow slowed down the robot. The first suggestion of using a stopwatch implies an indirect comparison between the speed of the robots [1d]. Here, a natural experimental setup is to measure the time used over the same distance. This may be a way to give an intuitive relation between time, speed, and distance.

Note that none of the observed groups took initiative themselves to use mathematics to reason why their solution was correct, they were satisfied with “seeing” that the robot “probably” slowed down.

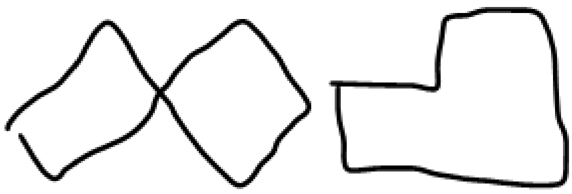

Figure 1: Illustration of the sketches made by the ITS to represent two different tracks. Here, lines are used to indicate the indented path of the robot, and sharp corners represent blue or pink tiles.

Thus, simplifying the syntax to only investigate the yellow tile was not sufficient. In addition, the teacher educator had an important role to make the ITSs reason and justify their claims. To make the ITSs continue their exploration in a systematic way, the teacher educator asked what the ITSs could do to be sure what the yellow tile does. This was sufficient to prompt the ITSs to propose several methods for how they could show that the yellow tile slowed down the robots [1d].

Disable the run-command

In the next part of the session, two new tiles were introduced; a pink and a blue tile that makes the robot rotate 90 degrees to left and right respectively. With a set of tiles at their disposal (1 green (start), 1 red (stop), 1 yellow (slow down), 4 pink (left turn) and 4 blue (right turn)), the ITSs were asked to "create a track that makes the robot drive on for as long as possible". The goal of this task is to facilitate development of reasoning and argumentation competence. We know that students` use of unsystematic trial-and-error might have a negative impact on their formulation of mathematical arguments when engaging in programming (Kaufmann & Stenseth, 2021). Therefore, to provide mathematical reasoning and communication, the teacher educator after a while reduced the opportunities for the trial-and-error method in this activity by removing the robots (*disabling the run-command)*.

When the robots were removed from the ITSs they started to use a variety of different strategies to evaluate their tracks. Pointing with fingers (or just with the eyes) and saying the instructions out loud were used by all observed groups [3d]. Objects such as unused tiles were also used to represent the robot and the ITSs would "drive" these objects through the tracks by moving them manually. One group additionally made use of pen and paper to represent the tracks, as illustrated in Figure 1, and used this representation when discussing feasible loops [3b,3d].

When the groups had access to the robots, we often observed that one group member laid out the tiles and then immediately ran the track with the robot without discussing it with the other group members. In addition, we observed that the ITSs spent a lot of time adjusting the tiles' position, sighting the initial angle of the robot, and handling other technical issues with the robots. When the robots were removed, we observed that the ITSs more actively explained how they thought the robot would drive in the proposed track [2c].

This didactical choice is of course not without pitfalls. Removing the robots do not completely remove trial-and-error approaches as it is easy to visualize the track and adjust the tiles "on the fly". Trial-and-error methods where the ITSs did not communicate or explicitly reason were still observed in all groups. Another issue is that when the robots are removed, this also removes the feedback from

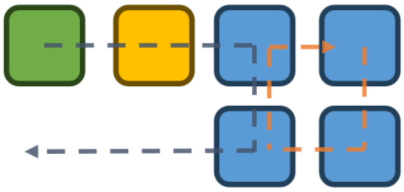

Figure 2: An example of a track created by the ITSs. The dashed blue line shows the path the robot will take, while the dashed orange line indicates the intended path

Figure 3: Example of a loop with 720 degrees turns before returning to the start

the robots, exemplified by one group's first attempt of making a loop. This attempt was not successful, as the robot would not follow the track they expected (see track in Figure 2). After they had placed the tiles, the following sequence took place:

ITS 1:	Right, right, right, right [points at the blue tiles].
ITS 2:	Theoretically, it will drive for infinity.

The first ITS pointed at the four blue tiles when arguing for the loop [2a,2d]. Neither of the ITSs realized the error of the track, and they proceeded to try out more complicated tracks. Here, the teacher educator has an important role. A technique used by the teacher educator to make the ITSs catch such mistakes as presented above, was to ask the ITSs to pretend to be the robot and walk the track themselves. This action keeps the run-command disabled, however, it provides the ITSs a method to self-evaluate the tracks and provide feedback to themselves.

Remove obvious representations and open up problem to new representations

When the third task was presented, the ITSs had become familiar with the behaviour of the robots. This task started by asking the ITSs if it is possible to make loops where the sum of rotations are different than 0 and 360 degrees. To promote the use of other representations, the teacher educator took away both the robots and the tiles from the ITSs, which *removed the tiles as a representation* for the track. While working on task 3, the ITS used both degrees and number of tiles as a unit for rotation [3c] and translated between them [3b]. They also represented a left and right rotation as negative and positive rotation [3c]. Removing the tiles caused all four groups to create representations of the tracks in the form of drawings [3d], see Figure 1 for examples. Up to this point only one group had explicitly used drawings to represent the tracks, and all groups had extensively been manipulating the physical tiles in their discussion. However, while removing the robots and the tiles did make the students use drawings to represent the tracks, they struggled to make progress on the task. The teacher educator intervened the work to help the ITSs. The teacher educator "walked the path" shown in Figure 3 on the floor and asked:

Teacher educator:	How many degrees did I turn?
ITS:	720 degrees.
Teacher educator:	Can I make a loop with another number of degrees?
ITS:	Yes, 1080, 1440 and so on, by turning multiple times before returning to the start.

Here, the teacher educator by the act of walking around introduced the ITSs to the idea that rotation can represent a dynamic measurement (not only a static) [3c], and one ITS made the initial claim that it is possible to make any multiple of 360 [2d]. In the continued work, the ITS struggled to argue for why the sum of the rotations in a loop must be multiple of 360 degrees. Two of the groups formulated algebraic expressions that did not help them in justifying the claim [1b,3d].

Concluding remarks

In this paper, we discuss the observations of a teaching session with a group of ITSs. We investigate how the ITSs work on mathematical competencies when the teacher educator uses three of the teacher actions suggested by Brandsæter and Berge (2024).

Trial-and-error approaches may have a negative impact on the mathematical reasoning of students (Kaufmann & Stenseth, 2021), and removing the robots from the ITSs in this study aimed at reducing the trial-and-error approaches. However, simply removing the robots was not sufficient to promote reasoning and argumentation among the ITS, it was necessary for the teacher educator to ask questions such as “why do you think … happened?”, “how can you be sure?”, “how do you know?”, to make the students justify and reason about their claims. Removing the possibility of immediately running a program is also a key component of the PRIMM (predict, run, investigate, modify make)-methodology (Sentance et al., 2019) where the students shall predict what a computer code does before running it. It is important to keep in mind that disabling the run-command, is not something we think a teacher should always do. This choice should be evaluated by several factors, and maybe most importantly the goal of the teaching session.

Misfeldt and Ejsing-Dunn (2015) points to the importance of the teacher to focus on mathematical concepts and on the didactical principles used by the teacher when using programming to learn mathematics. The three teacher actions investigated in this paper did initially prompt students to work on mathematical competencies, but the actions were not sufficient by themselves. The teacher educator had an important role direct the attention of the students towards mathematical competencies during the ITSs work.

In this work we have observed ITSs experiences concerning the mathematical competencies exploration and problem solving, reasoning and argumentation, and representation and communication through the indicators presented in Table 1. While we found indications for work on all these competencies, we do not investigate to what extent or level each competence is exercised. Thus, in further works one should investigate in more depth each mathematical competence, e.g., by using the four levels of the MEG framework (Turner et al., 2013). In addition, expansion of the study to other mathematical competencies (Niss & Højgaard, 2019) should be done.

Acknowledgments

This research is partly funded by the Norwegian Research Council, project number 344313.

References

Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding –Priorities, school curricula and initiatives across Europe.* European Schoolnet. http://www.eun.org/documents/411753/817341/Coding+initiative+report+Oct2014/2f9b35e7-c1f0-46e2-bf72-6315ccbaa754

Bocconi, S., Chioccariello, A., & Earp, J. (2018). *The Nordic Approach To Introducing Computational Thinking and Programming in Compulsory Education.* Report prepared for the Nordic@BETT2018 Steering Group. https://doi.org/10.17471/54007

Brandsæter, A and Berge, R.L. (2024). Promoting mathematical competence development through programming activities, *Submitted*

Eriksen, S. and Vos, P. (2022). Kjerneelementer og eksempeloppgaver. *Tangenten – tidsskrift for matematikkundervisning*, 33(3), 45–51

Forsström, S. E., & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, *17*(12), 18–32. https://doi.org/10.26803/ijlter.17.12.2

Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, *52*(7), 1029–1048. https://doi.org/10.1080/0020739X.2020.1736349

Kunnskapsdepartementet. (2020). *Læreplan i matematikk 1.–10.trinn (MAT01-05)*. Fastsett som forskrift. Læreplanverket for Kunnskapsløftet 2020.

Misfeldt, M., & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. In K. Krainer & N. Vondrová (Eds.), *CERME 9—Ninth Congress of the European Society for Research in Mathematics Education* (pp. 2524–2530). Charles University in Prague, Faculty of Education, ERME. https://hal.archives-ouvertes.fr/hal-01289367

Niss, M., & Højgaard, T. (2019). Mathematical competencies revisited. *Educational Studies in Mathematics*, *102*(1), 9–28. https://doi.org/10.1007/s10649-019-09903-9

Pólya, G. (1990). *How to solve it: a new aspect of mathematical method.* Penguin Books Ltd

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2–3), 136–176. https://doi.org/10.1080/08993408.2019.1608781

Turner, R., Dossey, J., Blum, W., & Niss, M. (2013). Using Mathematical Competencies to Predict Item Difficulty in PISA: A MEG Study. In M. Prenzel, M. Kobarg, K. Schöps, & S. Rönnebeck (Eds.), *Research on PISA: Research Outcomes of the PISA Research Conference 2009* (pp. 23–37). Springer Netherlands. https://doi.org/10.1007/978-94-007-4458-5_2

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5