The intersection of programming and mathematics in an educational context relevant to young students

Anna Sjödahl

Örebro University, Faculty of Science and Technology, Örebro, Sweden; anna.sjodahl@oru.se

There is a lack of knowledge in research about the intersection between programming and mathematics in an educational context relevant to young students. This paper aims to contribute to the research field with a theoretical framework that captures this intersection. The intersection is explored by interpreting open programming tasks in a visual programming environment through the lens of Bishop's six mathematical activities, resulting in a framework that shows how mathematical activities intersect with either a context-independent programming process or context-dependent programming processes.

Keywords: Programming, mathematical activity, young students.

Introduction

The idea of an intersection between programming and mathematics is nothing new (Feurzeig, 1969). In the educational field, this has mainly been investigated amongst students with experience of formal mathematics education (e.g., Benton et al., 2017; Kaufmann & Stenseth, 2021), and more rarely in the context of preschool or the first years of school. It is reasonable to think that the intersection also exists and can be capitalized on in early childhood education. Therefore, this paper sets out to investigate how the intersection of programming and mathematics can be understood in a context that is relevant for young students. This paper focuses on the first year of school, but much of the previous literature referred to was conducted in a preschool setting. Therefore, the terms *children* and *students* are both used, depending on whether they describe the present study or previous literature.

Early childhood mathematics can be framed in different ways. Kozlowski (2022) pointed to three mathematical content areas: number concepts, spatial concepts, and measuring concepts. However, a more fruitful way of understanding early childhood mathematics is through the lens of Bishop's (1988a) six mathematical activities: designing, playing, explaining, counting, locating, and measuring. This changes the focus from mathematical content to what children do when they engage in mathematics. The activity perspective has proved to be useful when talking about young children's mathematics, in that it describes the mathematical activity that occurs in preschool, rather than preparing children for the mathematical content that they will encounter at school (Johansson, 2015).

Programming for young children can be taught through various programming environments. Common among these are robotics (e.g., Flannery & Bers, 2013) and visual programming environments (VPE) (e.g., Portelance et al., 2016). These kinds of environments offer young students the opportunity to explore powerful programming ideas.

The three mathematical areas pointed out by Kozlowski (2022) have been noted, to some extent, in research on the intersection of programming and mathematics for young children. Palmér (2017) reported results from a preschool intervention using educational robots, which show how children can explore spatial thinking, counting, and symbols. Likewise, Fessakis et al. (2013) noted the potential to learn one-to-one correspondence, counting, number comparison, orientation skills, and angle turn concepts amongst children aged 5–6 years engaged in tasks in a LOGO-like programming environment. However, there are few studies like these. As far as can be ascertained, no research has gone beyond exploring mathematics concerned with counting and spatial thinking in a programming context.

Depending on the perspective, mathematics for young children can take different forms. Changing from a content-oriented view (Kozlowski, 2022) to an activity-oriented perspective on mathematics (Bishop, 1988a) could lead to new opportunities to investigate the intersection between programming and mathematics and its different aspects of the intersection, given that such a perspective is concerned with the process of doing mathematics and programming rather than the intersection between concepts in these two areas. Hence, the purpose of this paper is to explore and develop a theoretical framework that provides a possibility to investigate the intersection of programming and mathematics. This is done by zooming in on the mathematical activities first-year students demonstrate when engaging in programming. To this end, the perspective on mathematics is shaped through the lens of Bishop's (1988a) six mathematical activities.

The following research question has guided the work: How can an intersection of programming and mathematics be framed if mathematics is treated as a set of activities? To illustrate the intersection, the actions of first-year students engaging in a programming task are presented.

Six mathematical activities – a perspective on mathematics

The theoretical framework that is explored and developed in this paper consists of two parts: a programming context relevant to young students and an activity-oriented perspective of mathematics. First, the mathematical perspective is expressed using the six mathematical activities suggested by Bishop (1988a). *Playing* is the root of hypothetical thinking and is represented through different types of games, such as imaginative or imitative games (Bishop, 1988a), that are constituted by more or less formal rules that can be explored (Bishop, 1988b). *Explaining* is the activity of answering the question "why?" and exposes the explanatory relationships between different phenomena (Bishop, 1988a), which can be expressed in various ways, such as using language, symbols, or as figural explanations (Bishop, 1988b). *Designing* is concerned with the idea of shaping nature into a particular structure by abstracting the unnecessary parts and creating a new object with a given purpose (Bishop, 1988a). *Locating* is the activity of exploring and symbolizing a spatial environment (Bishop, 1988b), which is important for understanding geometrical notions and ideas about direction, order, and finiteness (Bishop, 1988a). *Counting* is the use of counting systems and numbers (Bishop, 1988a) to compare and order discrete phenomena (Bishop, 1988b). *Measuring* relates to activities concerned with comparing, ordering, and quantifying relevant qualities (Bishop, 1988a), which also includes the development of measuring systems and the use of measuring instruments (Bishop, 1988b).

Programming contexts relevant for young students

The second part of the suggested framework concerns programming that is relevant for young students. This could be robotic programming or visual programming. Here, the intersection of programming and mathematics was investigated through a task in visual programming. An open-task design was used, similar to the design used by Zhong et al. (2016), because it offers students the opportunity to take part in the whole programming process: formulate a problem in a story or narrative form, followed by a plan that captures the idea in a format that can be translated into a new representation, such as a coded project in the VPE ScratchJr. This VPE allows young students to code their own stories in a creative and playful environment with a low floor and appropriate high ceiling, designed for children aged 5–7 (Flannery et al., 2013). The entities that can be coded in ScratchJr are called sprites and the code is written by adding blocks in a row. To guide the young students in their coding, an important feature of the task design is the making of a plan that helps them to find direction and motivate the design of their code. Thus, the programming environment and the task design allow young students to explore programming concepts and engage in programming activities.

Method

To develop a theoretical framework that supports an investigation into the intersection of programming activities and mathematical activities, the programming task described above has been analysed through an activity-oriented mathematical lens using a theoretically driven approach in which the concepts have been sensitized, using theoretical concepts as a starting point and interpreting them in a new context (Charmaz, 2006). For illustration, an episode with two young students was deliberately chosen as an example in which the most can be learned (Merriam & Tisdell, 2015) about the intersection. The episode shows students, aged 6–7 in the first term of their first year in school and therefore with limited prior experience in formal mathematical education and programming, engaging in the programming task. The episode displayed in the paper shows how the intersection of programming and mathematics can be revealed. The students' engagement in the programming task is captured as screen recordings and provides the analysis with data showing what the students do on the screen and how they communicate with each other. The names in the excerpt are pseudonyms.

The analysis began with an investigation of the task design to reveal what opportunities it might offer in terms of mathematical activity. Every mathematical activity suggested by Bishop (1988a) was sensitized by asking a number of questions to understand how the activity could be interpreted in the programming context; for example, what opportunities do the students have to explore the idea of direction (locating)? By interpreting mathematical activities in the task design, a first set of codes was developed to determine what the intersection of programming and mathematics might consist of. The screen recordings capturing the first-grade students' actions in the programming task were then analysed through further sensitizing of the mathematical activities. The initial set of codes was used to identify and interpret relevant situations. For example, the task offered the students opportunities to explore direction when they used code blocks with arrows to move the sprite from one spot to another. Identifying these kinds of situations made it possible to view the intersection of programming and mathematics by looking at how the students demonstrated their mathematical activity in an age-appropriate programming task and how the mathematical activities related to each other. Thus, the theoretical framework outlined in this paper emerged by systematically working through all six mathematical activities in this way.

Results

At the end of this section, Table 1 displays the outcome of this exploration in the form of a theoretical framework. The analysis distinguished two types of programming processes intersecting with mathematical activities. Playing, explaining, and designing are demonstrated as elements of a *context-independent programming process*, which is not dependent on a specific programming environment. In contrast, counting, locating, and measuring are demonstrated as elements of a *context-dependent programming process*, possible because of specific features in ScratchJr.

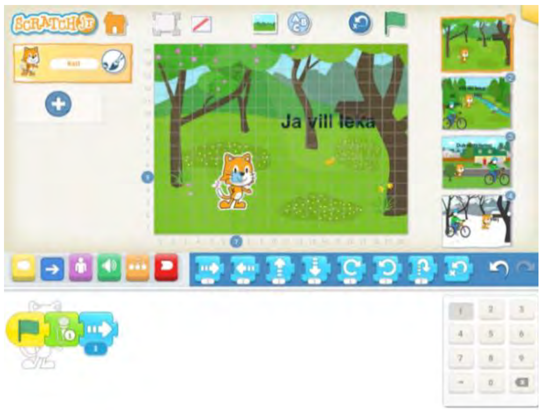

Figure 1: Sara's project in ScratchJr with the output view covered with a grid and the code blocks at the bottom

The episode chosen to illustrate the intersection shows how Maria joined Sara and her friend to help them with the problem of how to make a sprite move from one spot to another (see Figure 1). This episode is chosen due to the presence of all mathematical activities and the results are presented through the two distinct types of programming processes.

Sara:	He should actually go there. We didn't know how to do it.
Maria:	Yes, I know.
Sara:	How?
Maria:	Like this. And then you take another one. Did you have this one move forward? Then you click there. As much as you want. [Maria shows how to change an arrow block from one to any other number.]
Sara:	But we did that.
Maria:	Yes, but. Was he standing about there?
Sara:	And then he moves forward.
Maria:	Perhaps he should move 10 steps forward if he stands there.
Sara:	No, because we've already decided that he should stand there.
Maria:	Look. Here's a grid. As many squares as you should go. So he could perhaps move to the bushes or something. [Shows the grid in the output view]
Sara:	4,5,6,7,8,9,10,11,12,13,14. Okay. 14.

It later turned out that Sara could have been referring to a different problem than the one Maria wanted help with. Nevertheless, their conversation becomes interesting since they demonstrate mathematical activities belonging to both context-independent and context-dependent programming processes.

A context-independent programming process

Programming is about exploring rules since every programming environment has its own unambiguous language to use and be creative with. In the excerpt above, the girls accept and explore the rules of ScratchJr, notable as Maria shows how to set an arrow block to any number of steps. In that sense, they are involved in the mathematical activity of *playing*. Another aspect of playing is demonstrated already in the first line when Sara expresses an idea: "He should actually go there". She and her friend have an idea about where the cat should move to and have imagined what should happen, a sort of imaginative game. It can be seen as an element in a context-independent programming process since it is necessary to imagine an idea before starting to think about how that idea could be represented in code, regardless of which programming environment would be in use.

Maria interprets the other girls' intentions and thinks she has a solution to their problem. She starts to explain by telling and showing in the VPE. Sara tries to tell Maria that this is not the problem. Maria continues by showing them what to do: "Perhaps he should move 10 steps forward if he stands there." Sara expresses their intentions and what they have already decided to do: "No, because we've already decided that he should stand there." Even though the communication between the two girls is not working, they are both involved in the activity of *explaining*, answering the question of why, using both language and figural explanations. Sara refers to an idea as an argument for why the cat should be positioned in a specific spot. The idea itself is not revealed in this excerpt but the way Sara expresses it indicates that she uses their plan as an argument to explain why the cat should have a specific positioning. At the same time, Maria is busy explaining her conceptual understanding of what the arrow blocks do, how they can be used, and what function the grid has. She does this partly through language, although her conceptualization is mainly communicated through figural explanation, as she shows how the blocks can be set to any number depending on how far the sprite should move. Thus, Maria is explaining her conceptual understanding through theory-in-action. In that sense, mathematical explaining can be demonstrated in any programming environment, by coding according to a theory about what the code is expected to do.

Once Maria has introduced the grid, that measuring instrument seems to capture the attention of Sara, who starts to use it to solve how many steps the cat should take, ending up at 14. Their communication leads to an action of coding. Coding becomes a way of giving shape to a particular structure by abstracting unnecessary parts and creating a new object with a given purpose. In this case, Sara only focuses on what is interesting at the moment, how the cat should move, and codes that as a part of the programming project she and her friend intended to create. The shaping of their intention can be understood as the mathematical activity of *designing*. In that sense, coding in any programming environment would be a demonstration of designing given that the coding is done with a purpose.

Context-dependent programming processes

In contrast to playing, explaining, and designing, counting, locating, and measuring are connected to context-dependent programming processes, which students can be engaged in depending on the specific programming environment. As shown above, Maria and Sara are interested in different aspects of the design process: Sara focuses on the idea of what to shape and Maria focuses on how to shape it. However, both are interested in where the cat should be positioned and move in the two-dimensional space in the output. They explore a spatial environment. Maria asks where the cat was standing before and suggests: "Perhaps he should move 10 steps forward if he stands there". Sara expresses her idea about where the cat should be located and then starts to work out how many steps are needed to get to a different location. These acts demonstrate the mathematical activity of *locating*. In this case, it is the visual feature in ScratchJr that makes locating possible to engage in.

Sara is also involved in *counting*, which is made more accessible thanks to Maria's introduction of a grid. Sara uses the grid and demonstrates her understanding of one-to-one correspondence and cardinality as she compares the number of squares in the grid with discrete numbers in the counting system to work out how many steps are needed. Counting is probably present in any programming environment since computers are good at processing numbers. However, for those girls, counting is dependent on the VPE. They count what they see. In that sense, counting becomes a context-dependent programming process for them.

Using the grid as a measuring instrument is one type of *measuring* act. Also, the students get involved in measuring in ways that are not displayed in the excerpt above. They compare relevant quantities as they change the size of the sprites with the intention of making them coherent with each other. Using the grid and changing the size of the sprites are measuring activities that are dependent on specific visual features in the programming environment.

Table 1: A framework for intersecting programming and mathematical activities (Bishop 1988a, 1988b), highlighting a context-independent programming process and context-dependent programming processes

Mathematical activities in a context-independent programming process	**What students do when engaging in programming**
Playing	Imagining an idea that could be represented in a programming environment; exploring and relating to rules in a programming environment
Explaining	Expressing arguments verbally or as theory-in-action for why an idea or a representation in a programming environment needs a specific shape
Designing	Creating a representation of an idea in a programming environment
Mathematical activities in context-dependent programming processes	**What students do when engaging in programming**
Locating	Exploring a spatial environment in the visual output
Counting	Using the counting system to compare visual output with discrete numbers
Measuring	Comparing relevant quantities and using measuring instruments supported by the VPE

Discussion

The framework proposed here shows an intersection of programming and mathematics in an educational context that is relevant to young students. The results highlight two different types of processes: a context-independent programming process, which are independent of specific features in the programming environment, and context-dependent programming processes, which are dependent on what features the programming environment offers. Those two types of processes offer different opportunities to engage in mathematical activities.

Previous research regarding the intersection of programming and mathematics among young children focused on mathematics that can be made accessible in context-dependent programming processes.

For example, Palmér (2017) showed how robots can be used to explore spatial thinking, which could be interpreted as the activity of locating. Fessakis et al. (2013) identified the potential of developing the counting activity through one-to-one correspondence in a VPE. Those results request specific programming environments for certain mathematical content or activities that provide a limited picture of the potential that programming possesses to support teaching and learning in the mathematics classroom.

The present study has shed light on what mathematical activities a context-independent programming process enables. When young students engage in an open-task design they are able to imagine an idea and how to solve it (playing), producing arguments for how that solution could be formulated (explaining) and shape a new object with a specific use or intention (designing). The programming process involving playing, explaining, and designing is independent of which programming environment is used. Instead, what becomes important is the task design. Open tasks allow young students to engage in those three mathematical activities, letting them engage in the creative process of programming, starting with formulating ideas about what they intend to create.

The results discussed here have implications for teacher practice when using programming to engage young students in mathematics. First, presence of mathematical activities motivates the use of programming in the mathematics classroom since it provides opportunities to engage in a range of different mathematical activities. Further, the choice of programming environment, as well as the choice of task design, will affect what mathematical activities will be enabled. The type of task design used in this study presents opportunities to engage in a context-independent programming process. If the purpose of using programming in the mathematics classroom is to target mathematical activities enabled by context-dependent programming processes, the task design probably needs to be directed more towards specific features in a programming environment. Further research will be needed to understand how programming tasks can be designed to create opportunities for young students to engage in mathematical activities.

References

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115–138. https://doi.org/10.1007/s40751-017-0028-x

Bishop, A. (1988a). *Mathematical Enculturation: A Cultural Perspective on Mathematics Education*. Springer Netherlands.

Bishop, A. (1988b). Mathematics education in its cultural context. *Educational Studies in Mathematics*, *19*(2), 179–191. https://doi.org/10.1007/BF00751231

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Sage Publications.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers Education Information Technologies*, *63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Feurzeig, W. (1969). *Programming-Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the First Fifteen Months of the LOGO Project*. (Report No. 1889). Bolt, Beranek & Newman Inc.

Flannery, L. P., & Bers, M. U. (2013). Let's Dance the "Robot Hokey-Pokey!": Children's Programming Approaches and Achievement throughout Early Cognitive Development. *Journal of Research on Technology in Education*, *46*(1), 81–101. http://doi.org/10.1080/15391523.2013.10782614

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for early childhood learning through computer programming. In J. P. Hourcade, E. A. Miller, & A. Egeland (Eds.), *Proceedings of the 12th international conference on interaction design and children* (pp. 1–10). Association for Computing Machinery. https://doi.org/10.1145/2485760.2485785

Johansson, M. (2015). *Perceptions of mathematics in preschool: "-Now we have a way of talking about the mathematics that we can work with"*. [Doctoral dissertation, Luleå tekniska universitet]. Digitala Vetenskapliga Arkivet. https://www.diva-portal.org/smash/record.jsf?dswid=8235&pid=diva2%3A990408

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

Kozlowski, J. S. (2022). *Children's Mathematical Engagement based on Their Awareness of Different Coding Toys' Design Features*. [Doctoral dissertation, Utah State University]. ProQuest. https://www.proquest.com/openview/abda62bc295a8e80c0648ad17325bc2e/1?pq-origsite=gscholar&cbl=18750&diss=y

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.

Palmér, H. (2017). Programming in Preschool – With a Focus on Learning Mathematics. *International Research in Early Childhood Education*, *8*(1), 75–87.

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr Programming Language in the Early Childhood Classroom. *International Journal of Technology and Design Education*, *26*(4), 489–504. http://dx.doi.org/10.1007/s10798-015-9325-0

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, *53*(4), 562–590. https://doi.org/10.1177/0735633115608444