Examining How Students Leverage Code as a Representational Tool in Combinatorics

Rebekah Kuss
Oregon State University

Elise Lockwood
Oregon State University

Keywords: combinatorics, discrete mathematics, computing, programming.

In order to have students use computational tools in exploring mathematical ideas (e.g., Papert, 1980) and to explore the role of computing in students' mathematical reasoning (e.g., Geraniou et al., 2024; Lockwood & Morken, 2020), we wanted to explore how students engage with coding modules while solving combinatorics problems. While analyzing student work, we noticed that students pointed out different aspects of their code in their explanation of how they solved the problems. In this poster, we seek to answer the following research question: *What aspects of the code did students attend to as they used code as a representation in solving combinatorial problems, and how did they leverage these aspects of code*?

Literature and Theoretical Perspectives

This poster will focus on connecting counting process and sets of outcomes from Lockwood's (2013) model, whose components are counting processes (the procedures used when solving counting problems, formulas/expressions (the number, variables, and operations used to get an answer to a counting problem), and sets of outcomes (the sets of objects that are being counted). We build upon prior work that has explored students' use of programming while solving counting problems (e.g., De Chenne & Lockwood, 2022; Lockwood & De Chenne, 2020). Here, we will expand on such prior work on students' use of code and aspects of the code as a representation. In this poster, we will also focus on *machine-based computing*: "the practice of developing and precisely articulating algorithms that may be run on a machine" (Lockwood & Morken, 2021, p. 405), which in this poster is programming (both for writing and editing code) (in alignment with Weintrop et al., 2016). The code itself is the counting process, and the output of the code is the set of outcomes.

Methods

A set of modules were designed that focused on students engaging with coding in Python and counting problems, and they were implemented in an upper-level mathematics classroom for pre-service teachers. The whole class worked through the modules, and we video- and audio-recorded pairs of students during six class sessions while they worked; the instructor and another researcher checked in and asked questions about their thinking and work. For data analysis, we reviewed the paired sessions and identified instances in which students referred to particular aspects of the code (including initialized sets and lists, lines of code, and output), and we examined, compared, and contrasted episodes where students focused on different parts of code.

Results

In this poster, we will show different ways that the students used various aspects of the code to represent different aspects of their combinatorics process. We found that some of the ways that the code was used to represent their listing process, to represent their counting process, and using the output of the code as a representation. In all, the code aligned with counting processes, sets of outcomes, and the connection between the two from Lockwood's (2013) model.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1650943.

References

De Chenne, A. & Lockwood, E. (2022). A task to connect counting processes to lists of outcomes in combinatorics. *Journal of Mathematical Behavior*, *65*. https://doi.org/10.1016/j.jmathb.2021.100932

De Chenne, A. & Lockwood, E. (2023). A Case of a Computational Setting Facilitating Empirical Re-Conceptualization. In Cook, S., Katz, B. & Moore-Russo D. (Eds.). *Proceedings of the 25th Annual Conference on Research in Undergraduate Mathematics Education*. Omaha, NE. 494-501.

Geraniou, E., Faggiano, E., Medová, J. *et al.* Guest Editorial for Special Issue "Digital Experiences in University Mathematics Education: Advances and Expectations". *Int. J. Res. Undergrad. Math. Ed.* 10, 683–689 (2024). https://doi.org/10.1007/s40753-024-00260-4

Lockwood, E. (2013). A model of students' combinatorial thinking. *Journal of Mathematical Behavior, 32,* 251-265. doi:10.1016/j.jmathb.2013.02.00.

Lockwood, E. & De Chenne, A. (2020). Using conditional statements in Python to reason about sets of outcomes in combinatorial problems. *International Journal of Research in Undergraduate Mathematics Education, 6,* 303-346. doi:10.1007/s40753-019-00108-2

Lockwood, E. & Mørken, K. (2021). A call for research that explores relationships between computing and mathematical thinking and activity in RUME. *International Journal of Research in Undergraduate Mathematics Education, 7*(3), 404-416. doi: 10.1007/s40753-020-00129-2.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas.* New York: Basic books.

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 and Education Technology, 25*, 127-147. Doi: 10.1007/s10956-015-0581-5.