

Exploring Students Solutions to Concurrent and Parallel Programming Exercises – Impact of Generative AI

Maxim Mozgovoy

Department of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Japan, mozgovoy@u-aizu.ac.jp

Calkin Suero Montero

Department of Information Technology, Uppsala University, Uppsala, Sweden, calkin.suero.montero@it.uu.se

ABSTRACT

Background. Concurrent and parallel programming is difficult to teach and learn as the understanding of complex and abstract concepts such as nondeterminism, semaphore, and rare conditions, among others, is required [1, 2, 9], having as core issue the synchronisation of processes to achieve a common goal [4]. It is well-acknowledged that concurrent and parallel programming skills are fundamental as nowadays computing is increasingly handled in a parallel manner [7].

Problem and Motivation. Therefore, identifying the pitfalls and successes of students when solving practical concurrent and parallel programming exercises could shed light on best approaches and strategies that they use [3]. In addition, the advent of large language models, and generative AI applications such as ChatGPT, has prompted intensive research on their use in several areas including programming teaching and learning [8]. Yet, the studies in the literature have focused on issues related to learning to program by novice students in introductory courses (e.g., CS1, CS2) [6]. Less work, however, has been presented on the impact of generative AI tools in advanced programming practices such as concurrent and parallel programming.

Methodology. To investigate whether generative AI has had an impact on the submitted concurrent and parallel programming exercises solutions at the University of Aizu, Japan, we performed a comparison analysis of the students' submissions over 2020–2023. The analysis included five different exercises covering the basis of concurrency through various tasks and scenarios where the implementation of parallel processes is needed as solution. For instance, exercises 2.3 and 2.4 required to create parallel processes and perform independent computations; exercises 3.2 and 3.3, required synchronisation of the parallel processes; and in exercise 3.5 a code template was given for modification. We analysed the submissions of 72 undergraduate 3rd year students (avg. 18 students/year) and labelled the solutions using the following nomenclature: OK, indicating a good solution; OKFeat, a good solution but with unusual features; AdvLib, use of unnecessary advanced library or functionality; BadTool, use of an inappropriate tool when the task definition explicitly required a different tool; CodeErr, general coding error; SyncErr, concurrent programming specific error; N/A, solution not submitted or incomplete.

Results and Analysis. Results show a substantial increase in the incidence of use of advance libraries (AdvLib) and the wrong tools (BadTool) among students in 2023 for three out of the five analysed exercises. At the same time the

concurrency programming-specific errors (SyncErr) also see a reduction in all the exercises. (Figure 1). This coincides with the availability of generative AI tools such as ChatGPT [5], which warrants further investigations to understand how students, teachers and instructors could harness the affordances of large language models in their concurrent programming learning, teaching, and practice.

Contribution and Impact. This paper presents an initial step towards investigating the impact of generative AI on advanced programming topics. This research will continue to uncover strategies for the lecturers and instructors to identify the affordances and use of generative AI and to design exercises that harness these affordances to support students learning of difficult programming concepts.

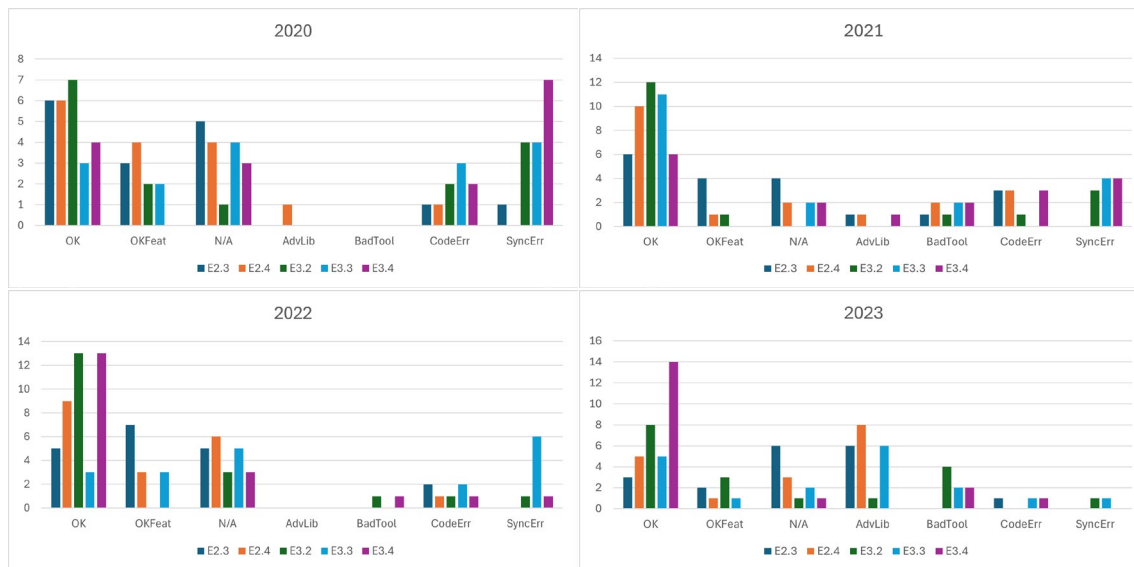


Figure 1: Four-year comparison of concurrent and parallel programming exercises solutions at the University of Aizu. An increase in the use of advance libraries and a decrease in concurrent programming errors (SyncErr) is of notice coinciding with the availability of generative AI technologies in 2023.

CCS CONCEPTS • Computing methodologies ~ Parallel computing methodologies • Computing methodologies ~ Concurrent computing methodologies ~ Concurrent algorithms

Additional Keywords and Phrases: Large language models in advanced programming, Evaluation of students' exercises.

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock,

NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

REFERENCES

- [1] Michal Armoni, and Mordechai Ben-Ari. 2009. The concept of nondeterminism: its development and implications for teaching. *ACM SIGCSE Bulletin* 41(2) (2009), pp. 141–160.
- [2] Yifat Ben-David Kolikant. 2004. Learning concurrency as an entry point to the community of computer science practitioners. *Journal of Computers in Mathematics and Science Teaching* 23(1), pp. 21–46.
- [3] Jan Lönnberg. 2009. Understanding students' errors in concurrent programming. Licentiate's thesis, Helsinki University of Technology.
- [4] Santiago Ontañón, Jichen Zhu, Brian K. Smith, Bruce Char, Evan Freed, Anushay Furqan, Michael Howard, Anna Nguyen, Justin Patterson, and Josep Valls-Vargas. 2017. Designing visual metaphors for an educational game for parallel programming. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2818–2824.
- [5] OpenAI. 2022. Introducing ChatGPT. Retrieved May 12, 2024 from <https://openai.com/index/chatgpt/>
- [6] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reily, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The robots are here: Navigating the generative ai revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, pp. 108–159. <https://doi.org/10.1145/3623762.3633499>
- [7] Herb Sutter. 2005. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobbs' journal* 30(3), pp. 202–210.
- [8] Ramazan Yilmaz, and Fatma Gizem Karaoglan Yilmaz. 2023. The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence* 4 (2023): 100147.
- [9] Jichen Zhu, Katelyn Alderfer, Anushay Furqan, Jessica Nebolsky, Bruce Char, Brian Smith, Jennifer Villareale, and Santiago Ontañón. 2019. Programming in game space: how to represent parallel programming concepts in an educational game. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pp. 1–10. 2019.