International Society of the Learning Sciences

Uncovering Middle School CS Students' Understanding of Variables and Control Structures: A Cognitive Think-aloud Approach

Hui Yang, Satabdi Basu, Daisy Rutstein, Arif Rachmatullah, Carol Tate, Christopher Ortiz and Eliese Rulifson
{hui.yang, satabdi.basu, daisy.rutstein, arif.rachmatullah, carol.tate, christopher.ortiz, eliese.rulifson} @sri.com
SRI International

Abstract: This poster presents findings on middle school students' understanding of core Computer Science (CS) concepts, such as variables and control structures, using cognitive think-aloud interviews with eight students. Each student worked on 16-22 formative assessment tasks designed to assess understanding on the 'Algorithms and Programming' middle school CS standards. Our study describes students' interpretations of the CS concepts and discusses potential factors influencing student interpretations. Significance and next steps are described.

Introduction and Research Questions

Education researchers and policy makers have emphasized introducing CS as early as possible to prepare K-12 students for future careers and life in a computationally intensive society. Many CS curricula, particularly for middle school levels, have been developed to help students understand key introductory CS concepts such as variables and control structures. These curricula predominantly employ a block-based programming representation which helps students learn core CS concepts without having to deal with complex syntax. However, some studies have explored and found that young students still encounter challenges in understanding the concepts of variables and control structures, even when using block-based code. For example, Mladenović and colleagues (2018) compared elementary students' misconceptions of loops in block-based and text-based programming languages and found that young students had misconceptions about simple and nested loops, even in block-based languages. Although some such studies have started to explore students' challenges with key CS concepts, there is still a need for research on middle school students' understanding of key CS concepts, such as variables and control structures, using a variety of programming representations, both block-based and text-based. Here, we report on a pilot study leveraging a cognitive think-aloud approach to uncover middle school students' challenges with the concepts of variables and control structures and identify the contributing factors. Our research questions are as follows: (1) What do middle school students' responses to various short assessment tasks reveal about their understanding of the concepts of variables and control structures?; and (2) What are the potential factors influencing middle school students' responses to CS assessment tasks?

Methods

Context

We employed an evidence-centered design approach (Mislevy & Riconscente, 2006) to design 75 formative assessment tasks aligned with fine-grained learning targets associated with middle school 'Algorithms and Programming' standards. Tasks used a JavaScript programming representation and a combination of block- and text-based formats for these tasks to align with the representation used in the CS Discoveries (CSD) curriculum that was familiar to all participating students (Basu et al., 2022). We present data collected from eight hour-long interviews with seven 8th-grade students and one 7th-grade student who worked on 16-22 tasks each. About half the students reported having taken CS courses in their previous grade levels. All students were instructed to independently think-aloud through each task during the one-hour interviews. All the interviews were audio recorded and transcribed.

Data Analysis

Based on the cognitive interview recordings and notes, we developed a memo for each student by listing their final answers for each assessment task, how they described their problem-solving processes, challenges they faced, time taken to finish the task, and feedback on the tasks. The analytical approach comprised reviewing students' responses and their thinking processes and grouping their responses into five categories: 1) provided correct answers and correct reasoning; 2) provided correct answers but partially correct reasoning; 3) provided incorrect answers but a partially correct reasoning; 4) provided incorrect answers and inaccurate/irrelevant explanations; and 5) decided to skip due to lack of understanding of the task and/or the underlying concept.

Preliminary Findings

Students' understanding of variables

Students demonstrated an understanding of how to *name variables* appropriately on half of the tasks (2 students each responded to 2 tasks on this aspect). There was only 1 task aligned with *variable initialization*, and the only student who responded to this task was able to provide the correct answer. In contrast, students struggled with *manipulating values of variables*, whether it be numeric variables, string variables, or variables representing compound datatypes such as lists. For instance, some students did not understand what a string variable is, so they had challenges parsing string variables. Some students did not understand when two numeric variables are linked by assignment such as "count = start + 1." Moreover, students struggled with using variables alongside loops and conditionals.

Students' understanding of control structures

Students answered most tasks on *nested loops* and *nested conditionals* partially or fully correctly (6 students each worked on 1-2 tasks on each of these two aspects). In comparison, students demonstrated difficulty understanding code that includes a compound conditional statement using Boolean operators as part of the condition. Students could neither accurately identify needed Boolean expressions for representing a condition nor predict the correct output of a program with a conditional including Boolean operators. For *repeated conditionals*, only 2 students worked on aligned tasks, and only one student answered correctly, albeit with partially correct reasoning. *Procedures inside control structures* was also difficult for students; 4 students each worked on an aligned task and only two students answered correctly with partially correct explanations.

Potential factors influencing students' reasoning

We found that some factors may influence students' reasoning, such as their prior programming experience and familiarity with JavaScript block-based and text-based representations. We noticed that most students reasoning process leveraged their prior programming experience. Students familiar with other block-based programming languages such as Scratch often tried to tie their reasoning back to that representation, even when not appropriate. Regarding code representations, we found that students generally preferred a visual block-based representation, especially some students who had never seen text-based code, even though the CSD curriculum programming environment allows them to toggle between block-based and text-based representations.

Discussion and future directions

This work is part of a pilot study for a larger project that aims to deepen middle school teachers' understanding of five Algorithms and Programming standards by implementing standard-aligned formative assessments. Our findings suggest that programming representation plays a significant role in formative assessment tasks and the types of student understanding and challenges that tasks can elicit. To forge deep conceptual learning, students should be exposed to different programming representations so that their learning is not limited to the affordances and constraints of one programming representation. Regarding the next steps, we are revising some of our formative assessment tasks to better capture student challenges and understanding of the underlying concepts.

References

Mladenović, M., Boljat, I., & Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, *23*(4), 1483-1500.

Mislevy, R. J., & Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational measurement: issues and practice*, *25*(4), 6-20.

Basu, S., Rutstein, D., Tate, C., Rachmatullah, A., & Yang, H. (2022, February). Standards-Aligned Instructional Supports to Promote Computer Science Teachers' Pedagogical Content Knowledge. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (pp. 404-410).

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. DRL-2010591. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.