Evaluating the Effectiveness of Explicit Instruction in Reducing Program Reasoning Fallacies in Elementary Level Students

Ashish Aggarwal
University of Florida
Gainesville, FL 32611, U.S.A
ashishjuit@ufl.edu

Christina Gardner-McCune
University of Florida
Gainesville, FL 32611, U.S.A
gmccune@ufl.edu

David S. Touretzky
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A
dst@cs.cmu.edu

ABSTRACT

Previous research in K-5 CS education has focused on improving students' engagement in programming using visual block-based environments like Scratch. However, little is known about how elementary school students' reason about programs. We define computational reasoning as the ability to read, write, trace and debug programs and predict program behavior. Recently, computing education researchers have become interested in exploring how elementary school students build their computational reasoning abilities. This poster presents results from a study which analyzed the role of explicit instruction in the form of 'laws of computation' in cultivating elementary school (4th and 5th graders) students' ability to reason about programs using Microsoft Kodu Game Lab. We used pre-tests to record students' default models of reasoning about programs and then used post-tests to measure the effectiveness of intervention by noting students' reasoning responses on a similar program. Our findings indicate that by default students reason sequentially about program execution which can be incorrect in situations like parallel rule execution. We also found that the use of explicit instruction in the form of 'laws' is helpful for students to refine their understanding of program execution and to improve their reasoning ability.

CCS CONCEPTS

• Social and professional topics~K-12 education

KEYWORDS

Computational Reasoning, Explicit Instruction, Kodu, Elementary School Students, Analogical Reasoning, Program Comprehension.

ACM Reference format:

Ashish Aggarwal, Christina Gardner-McCune, and David S. Touretzky. 2019. Evaluating the Effectiveness of Explicit Instruction in Reducing Program Reasoning Fallacies in Elementary Level Students. In *Proceedings of ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'19). ACM, New York, NY, USA, 1 page.*
https://doi.org/10.1145/3304221.3325588

1 EXTENDED ABSTRACT

As the demand to include computing activities into K-12 curriculum increases [4], it is important to focus on evaluating and assessing how students develop correct mental models [1]. In the past, there has been extensive work on the use of analogies in helping students learn about computers. While the use of analogies may be helpful, incorrect use of analogy may inhibit students from developing correct mental models of reasoning [2].

The study presented in this poster consisted of 18 elementary school students (4th and 5th grade) who were asked to predict the program behavior of similar programs before and after an intervention. In this intervention, the study staff explicitly taught the students to reason about programs using the 'laws of Kodu' [3]. The pre-test was used to observe students' default reasoning on program execution and the post-test was used to understand the effectiveness of intervention in guiding students' reasoning about program execution. We found that a majority of students were sequentially reasoning about the execution of programs which was incorrect, and their reasoning significantly improved after the intervention.

The research presented in this poster highlights two contributions: (1) students actively apply analogical reasoning models accumulated through their day-to-day experience in reasoning about programs; and (2) while students cannot be prevented from using analogical reasoning, the use of explicit instruction can help in molding a student's reasoning wherever analogies do not hold true. We can thus help students to develop correct mental models of how program statements are evaluated. These models will improve students' ability to read and understand programs and predict program behavior, which is essential for their success as computational thinkers.

REFERENCES

[1] Ashish Aggarwal, David S. Touretzky, and Christina Gardner-McCune. 2018. Demonstrating the Ability of Elementary School Students to Reason About Programs. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). ACM, New York, NY, USA, 735-740. DOI: https://doi.org/10.1145/3159450.3159488

[2] Michael Clancy. 2004. Misconceptions and attitudes that interfere with learning to program. In Computer Science Education Research. London: Taylor & Francis Group, 85−100

[3] David S. Touretzky, Christina Gardner-McCune, and Ashish Aggarwal. 2017. Semantic Reasoning in Young Programmers. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 585-590. DOI: https://doi.org/10.1145/3017680.3017787

[4] Allen Tucker. 2003. A Model Curriculum for K--12 Computer Science: Final Report of the ACM K--12 Task Force Curriculum Committee. Technical Report. ACM, New York, NY, USA. ACM Order No.: 104043.

ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland Uk

ACM ISBN 978-1-4503-6301-3/19/07.
https://doi.org/10.1145/3304221.3325588