Misconception of Abstraction: When to Use an Example and When to Use a Variable?

Hanxiang Du
h.du@ufl.edu
University of Florida
Gainesville, FL, USA

Wanli Xing
wanli.xing@coe.ufl.edu
University of Florida
Gainesville, FL, USA

Yuanlin Zhang
y.zhang@ttu.edu
Texas Tech University
Lubbock, TX, USA

ABSTRACT

Abstraction, which is considered the most important computational thinking skill, can be learned from programming or computational thinking learning activities. We implemented a 8-week long course to teach high school students statistics and programming. This work reports some interesting observations we made on students' misconception of abstraction while examining students' responses to test questions.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; **Computing education**.

ACM Reference Format:

Hanxiang Du, Wanli Xing, and Yuanlin Zhang. 2022. Misconception of Abstraction: When to Use an Example and When to Use a Variable?. In *ICER'22, August 07–11, 2022, Lugano, Switzerland*. ACM, New York, NY, USA, 1 page. https://doi.org/xxxxxxxxxx

1 INTRODUCTION AND BACKGROUND

Abstraction is considered the most important computational thinking skill [2]. This study reports some preliminary interesting findings on students' misconception of abstraction. We define misconception as "a unit of cognitive function or structure that can be inferred from a mistake on a programming task" [1].

We developed a course integrating statistics and programming for high school students. Students first learned how to abstract a problem using mathematical expressions (e.g., set theory), then transferred the expression in R programming language—whose data structure is directly related to set theory—to solve the problem. A total number of 53 students registered for the elective face-to-face course in a public high school in Texas, Spring 2021. The 15 100-minutes-long lessons cover topics like sets and their representation, functions and their use, primitive types, the basics of statistics, and advanced topics like linear function and regression. A pre- and post-test was designed to measure students' understandings of computing and statistics. To grade the test questions, one professor of computer science and one senior doctoral student of education collaboratively developed rubrics. In addition, we conducted clinical interviews with a few volunteered students to ask them to solve a few problems that are similar to exercise questions. Researchers followed a think-aloud protocol to understand how students approach problem-solving. In our study, abstraction is measured in different ways: write mathematical expressions for problems, compose functions to present solutions, and manipulate sets and R vectors. While grading responses to post-test and translating interview records, we made some interesting observations.

ICER'22, August 07–11, 2022, Lugano, Switzerland

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/xxxxxxxxxx

2 RESULTS

Misconception 1, students used an example (i.e., a member of a collective) when they should have used a collective (e.g., variables, sets). Many students were able to deal with examples, but not abstract the problem. When students were asked to write the signature of the function + or write a R expression whose value equals the residues between actual observations and predicted values, some used specific examples (e.g., 1 + 2 = 3) instead of variables. For a few other questions where students were asked to write an expression for vector operations or indexing, some provided direct answer instead of an expression.

Misconception 2, the output of a function always changes as input changes. For a question whose output is a constant, students gave different answers for different input. Students considered the output as a variable that changes as the input changes—which is a misconception.

Our speculations: first, students cannot develop the concept of collective entity; second, students cannot manipulate multiple collective entities simultaneously; third, students do not understand either the differences between the value of a variable and the expression of a function, or the question itself.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation (NSF) of the United States under grant number 1901704. Any opinions, findings, and conclusions or recommendations expressed in this paper, however, are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] Luke Gusukuma, Austin Cory Bart, Dennis Kafura, and Jeremy Ernst. 2018. Misconception-driven feedback: Results from an experimental study. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. 160–168.

[2] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.