Augmenting Code Pattern Detection with Software Verification and Examining How Teaching Assistants Interact with Student Code Structure

Matthew Hooper
University of Utah

UUCS-21-014

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

10 May 2021

Abstract

Expert programmers expect certain commonly-used patterns in code. However, novice coding patterns often deviate from these expectations, even if they're still functionally correct. University instructors recognize the need to educate their students to use more advanced patterns, but giving personalized feedback is time-consuming, and teaching good coding style and structure doesn't have an obvious home in a 4-year curriculum. It may be possible to employ software that can automatically identify novice coding patterns, or even to instruct students about their style without using the limited time and resources of course instructors. Since teaching assistants are viewing and giving feedback about student code more often than instructors, it may be possible to use their experiences to design effective feedback for an automated pattern detector. How teaching assistants interact with students specifically about their code structure appears not to be studied previously. Through several interviews with teaching assistants I've found that teaching assistants don't always agree with expert coding patterns, and their reasons for whether or not they agree may be vague or poorly articulated - demonstrating a lack of mastery over certain programming topics. In addition, by applying techniques from software verification, I've been able to expand the kinds of novice patterns that can be detected by existing pattern detection tools.