More reliable software thanks to mathematical demonstrations

January 14 2015

Credit: Thinkstock

It is possible to test computer programs using tools borrowed from mathematicians and their famous demonstrations. A team from the EPFL is exploring new territory in this area.

"The status quo on the software market needs to change, explains Viktor Kuncak, Associate Professor at the School of Computer and Communication Sciences' Laboratory for Automated Reasoning and Analysis. All too often, developers adopt a software patch strategy for products that are already being sold on the market!" And yet there are tools out there that can improve the reliability of software before products reach consumers. Moreover, these tools are already widely used for hardware development. "Once a microchip has been manufactured,

it's too late to fix an error." In fact, the later an error is detected, the more expensive it becomes to fix it." Greater software reliability can also bring benefits to many different areas such as aviation or medical equipment, making them more reliable.

Viktor Kuncak and post-doctoral researcher Andrew Reynolds are currently working to develop automatic software verification tools. Just like a mathematical demonstration seeks to prove that an affirmation may be true for a given set of values, software verification is aimed at ensuring that a function will not cause a crash or produce an outlier for the full range of possible user entries or actions.

If there is one tool that mathematicians regularly use, it is mathematical induction. This method consists of proving that a theorem holds true for a given natural number (n) and that what is true for n is also true for the natural number that follows (n+1). Once this has been done, one must also prove the validity of the theorem for the smallest natural number: 0. This form of demonstration was previously not possible with an SMT solver (satisfiability modulo theories), which is an essential component of automatic software verification. The two researchers have now managed to overcome this obstacle, developing an approach that allows one to automatically use mathematical induction in a constraint resolution process. The code input language to be verified is Scala, "we chose it because of its similarity with mathematical formulations and reasoning," explains Viktor Kuncak.

Apart from the computerization of mathematical induction, this approach is also remarkable because it breaks down tasks. "Whenever a verification cannot be executed directly, our tool breaks the problem down into sub-problems that it may be able to resolve. When there is a known path, this path will be used. Otherwise, other paths will be explored and a new breakdown may be performed, thereby allowing a large number of potential problem-solving pathways to be explored,

explains Viktor Kuncak.

Although the EPFL researchers are more specifically interested in potential applications to software verification, advances in the area of automated mathematical proofs can also benefit other fields. "The primary focus of SMT solvers has been for problems faced by industry (including software verification). Only more recently have SMT solvers been used for solving more difficult problems, which would include solving open mathematical problems. So far, this has been somewhat limited, but is starting to become more common," states Andrew Reynolds. "One project along these lines is the TPTP library of problems pursued by the University of Miami. The use of SMT solvers such as the one I work on (CVC4) has led to solving several previously unsolved problems in this library."

The approaches that the EPFL researchers are focused on are interesting both in terms of success rates and speed of execution. What are the limits of this tool? "It's a bit like having a brilliant student," explains Viktor Kuncak, "We cannot ascertain the theoretical limits but empirically, we have observed that the tool has more difficulty handling certain formulas than others," The research conducted with Andrew Reynolds at the EPFL began in 2014. "We have obtained good results in only a few months, notes Viktor Kuncak. The EPFL has a certain lead in this field and we hope to consolidate this position."

More information: "Induction for SMT Solvers Lecture Notes in Computer Science," Volume 8931, 2015, pp 80-98. link.springer.com/chapter/10.1007%2F978-3-662-46081-8_5

Provided by Ecole Polytechnique Federale de Lausanne

Citation: More reliable software thanks to mathematical demonstrations (2015, January 14) retrieved 28 April 2024 from https://phys.org/news/2015-01-reliable-software-mathematical.html

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.