Semi-automated Test Case Generation for ACAS X Implementation Validation

Submitted by Katie Dey on Mon, 05/06/2019 - 3:24pm. Contributors: Daniel GeninMark ThoberM. Scott Doerrie

We present FastPACE (Provable Assertion Checking Engine), a weakest precondition analysis tool, developed for semi-automated generation of tests cases for the Test Suite of the Airborne Collision Avoidance System Xa/Xo (ACAS X). The ACAS X Test Suite will be used as part of an assurance argument for the system, showing that an implementation matches the system specification. ACAS X is the next generation replacement for the Traffic-Alert and Collision Avoidance System (TCAS) currently in use. Its goal is to prevent midair collisions between aircraft by issuing advisories to pilots when there is imminent danger of collision. It is, therefore, critical that the software components of the system operate robustly and correctly. System verification is complicated by the fact that specification and implementation of ACAS X are performed by separate organizations. The ACAS X specification, which is published as ACAS X MOPS DO-385 (Minimum Operational Performance Standards) by RTCA, was developed under the auspices of the FAA TCAS Program Office by MIT Lincoln Laboratory and JHU Applied Physics Laboratory (JHU/APL) in collaboration with other aviation regulatory organizations and avionics manufacturers. Implementation and integration of ACAS X into a complete avionics system is performed by avionics vendors. These implementations of ACAS X are proprietary and closed-source. In order to facilitate verification of vendor implementation adherence to the published specification, JHU/APL was tasked with developing the ACAS X Test Suite, providing, among other requirements, complete code coverage of the specification. The ACAS X specification is written in the Julia programming language, which makes code analysis of the specification more tractable than for a traditional specifications written in a mix of natural language and pseudo-code. We leveraged our knowledge of the system and a custom built weakest pre-condition analysis tool FastPACE to fulfill the complete code coverage requirement, which would have otherwise been an extensive manual effort. This talk describes FastPACE, and our experiences applying it during the development of the Test Suite for ACAS X. We discuss optimizations we implemented to make weakest pre-condition analysis scalable and the shortcomings of this approach due to existing SMT solver limitations. We also discuss our initial efforts to extend FastPACE to analyze MATLAB code, which is frequently used for algorithm design and specification.

Dr. Mark Thober is a Senior Computer Scientist and Principal Staff at JHU/APL. He received his B.S. in computer science and mathematics from Nebraska Wesleyan University and his Ph.D. in computer science from the Johns Hopkins University. His interests include automated code analysis, programming language-based security, type theory, and malware analysis.

Dr. M. Scott Doerrie is a Senior Computer Scientist at JHU/APL. He earned his bachelor's degree in computer science from Gustavus Adolphus College in 2002 and his Ph.D. in computer science from the Johns Hopkins University in 2015. His research interests include software assurance, machine-assisted verification, advanced type systems, safe language run-times, access control mechanisms, robust operating systems and robust hardware.

Dr. Daniel Genin is a Senior Computer Scientist at JHU/APL. He received his bachelor's and doctoral degrees in mathematics from the Pennsylvania State University. After joining APL Daniel contributed to a range of different projects from kernel integrity measurement to formal software verification. His current interests are in the area of formal software analysis and verification, as well as reverse engineering and embedded systems.