4yp

Thomas Aston

April 14, 2024

1 Introduction

In a world reliant on computer systems, the correctness of those systems are vital. Indeed, simple programming errors can lead to major, and sometimes fatal, incidents; examples of these include an automated trader losing $460 million[] and the Ariane flight V88 breaking up after launch due to an overflow error[1].

There are two main approaches to developing correct software - testing and verification. Though thorough unit testing can be effective in minimising software bugs[2], it is significantly less effective with concurrent code. This is predominantly due to the sheer number of different interactions between threads and the requirement to be able to both consider all interactions and model them. Linearizability testing is an effective alternative approach, although again this relies on the random testing of edge cases; clearly this is not exhaustive either[3].

By contrast, formal verification can be used to show that systems satisf[]

References

[1] Independent Inquiry Board. *ARIANE 5: Flight 501 Failure*. Report. 1996. URL: http://sunnyday.mit.edu/nasa-class/Ariane5-report.html.

[2] Gunnar Kudrjavets Laurie Williams and Nachiappan Nagappan. *On the Effectiveness of Unit Test Automation at Microsoft*. URL: https://collaboration.csc.ncsu.edu/laurie/Papers/Unit_testing_cameraReady.pdf.

[3] Gavin Lowe. "Testing for linearizability". In: *Concurrency and Computation: Practice and Experience* 29.4 (2017), e3928. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3928.