
Miscomputation in software development: Learning to live with errors

Tomas Petricek*¹

¹University of Cambridge – United Kingdom

Abstract

Computer programs do not always work as expected. There is a complex taxonomy of possible miscomputations or errors. Programs may not work as a result of syntax errors, incorrect implementations of algorithms or hardware failures. But what follows after a miscomputation? In this talk, I will discuss different practical strategies that software developers employ when dealing with errors or miscomputations. We look at four different approaches. In the first one, miscomputation must be avoided at all costs, e.g. using formal proofs. In the second, miscomputation is an integral part of the development process, but is avoided in the final software. In the third approach, miscomputation even becomes a normal part of running the software and is mitigated by the runtime. Finally, in the fourth approach, miscomputation is a part of human computer interaction during the software execution. In other words, the talk follows the history of cohabiting with miscomputation from the early days when programming errors were not acknowledged as a significant issue to the modern development methodologies that are finding new ways of living with ubiquitous errors.

*Speaker