Getting Software Verifiers Ready for Industrial Use

Dirk Beyer
LMU Munich, Germany
https://www.sosy-lab.org/~dbeyer/

Automatic software verification (model checking) became quite effective and efficient in the past two decades, and the technology is successfully applied by software giants such as Microsoft and Facebook, and for verifying Linux.

However, "software verification sucks" is the feedback from 'normal' developers. --- Why is that?

First, we need to blame the high false-alarm rate of early approaches that were based on imprecise data-flow analysis. It is no fun digging through large error traces just to find out the tool was wrong. This scared people away.

Second, we need to understand that we developed a large number of verifiers that are not as exchangeable and interoperable as they could be. It is bad to experience the technology lock-in effect: integrating a new tool into the workflow requires investment, ... and then a better tool is available.

Third, a large amount of resources is lost because there are no high-value, first-class objects that are produced and can be stored for later reuse. It is frustrating that after a verifier consumed expensive computing resources, the answer is a mere true or false, without anything more useful.

This presentation reports on some recent developments that are able to constructively change the picture.