Forthcoming Requirements on Software Verification

Patrick COUSOT
École Normale Supérieure
45 rue d'Ulm
75230 Paris cedex 05, France
Patrick.Cousot@ens.fr www.di.ens.fr/~cousot

Invited Panel on « The Future of Software Verification »,
Third International Workshop on Automated Verification of Infinite-State Systems (AVIS'04)
Barcelona, Spain, 3rd-4th April 2004

Who Cares?

- No one is legally responsible for bugs:

  This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

- So, no one cares about software verification
- And even more, one can even make money out of bugs (customers buy the next version to get around bugs in software)

Who Really Cares?

- The victims (for lost data, money and even lives)
- Victims get no repair
- So no one really cares
- The general public might lose confidence in software-based technology (this is one of the explanations for the success of open software)

Why No One Cares?

- Software designers don't care because there is no risk in writing bugged software
- The law/judges can never enforce more than what is offered by the state of the art
- Automated software verification by formal methods is undecidable whence thought to be impossible
- Whence the state of the art is that no one will ever be able to eliminate all bugs at a reasonable price
- And so no one ever bear any responsability

Current Research Results

- Research is presently changing the state of the art (e.g. ASTRÉE)
- We can check for the absence of large categories of bugs (may be not all of them but a significant portion of them)
- The verification can be made automatically by mechanical tools
- Some bugs can be found completely automatically, without any human intervention

The Next Step (5 years)

- If these tools are successful, their use can be enforced by quality norms
- Professional have to conform to such norms (otherwise they are not credible)
- Because of complete tool automaticity, no one can be discharged from the duty of applying such state of the art tools
- Third parties of confidence can check software a posteriori to trace back bugs and prove responsabilities

A Foreseeable Future (10 years)

- The real take-off of software verification must be enforced
- Development costs arguments have shown to be ineffective
- Norms/laws might be much more convincing
- This requires effectiveness and complete automation (to avoid acquittal based on human capacity limitations arguments)

Conclusion

- The state of the art will change toward complete automation, at least for common categories of bugs
- So responsabilities can be established (at least for automatically detectable bugs)
- Whence the law will change (by adjusting to the new state of the art)
- To ensure at least partial software verification
- For the benefit of all of us