Colloquium d'Informatique de l'UPMC Sorbonne Universités

Abstract interpretation

Patrick Cousot

New York University

Amphi 15

4, place Jussieu
75005 Paris
Metro Jussieu

29 Septembre 2016 à 18h00

The complexity of large programs grows faster than the intellectual ability of programmers in charge of their development and maintenance. The direct consequence is a lot of errors and bugs in programs mostly debugged by their end-users. Programmers are not responsible for these bugs. They are not required to produce provably safe and secure programs. This is because professionals are only required to apply state of the art techniques, that is testing on finitely many cases. This state of the art is changing rapidly and so will irresponsibility, as in other manufacturing disciples.

Scalable and cost-effective tools have appeared recently that can avoid bugs with possible dramatic consequences for example in transportation, banks, privacy of social networks, etc. Entirely automatic, they are able to capture all bugs involving the violation of software healthiness rules such as the use of operations with arguments for which they are undefined.

These tools are formally founded on abstract interpretation. They are based on a definition of the semantics of programming languages specifying all possible executions of the programs of a language. Program properties of interest are abstractions of these semantics abstracting away all aspects of the semantics not relevant to a particular reasoning on programs. This yields proof methods.

Full automation is more difficult because of undecidability: programs cannot always prove programs correct in finite time and memory. Further abstractions are therefore necessary for automation, which introduce imprecision. Bugs may be signalled that are impossible in any execution (but still none is forgotten). This has an economic cost, much less than testing. Moreover, the best static analysis tools are able to reduce these false alarms to almost zero. A time-consuming and error-prone task which is too difficult, if not impossible for programmers, without tools.

Patrick Cousot received the Doctor Engineer degree in Computer Science and the Doctor ès Sciences degree in Mathematics from the University Joseph Fourier of Grenoble, France. He was a Research Scientist at the French National Center for Scientific Research at the University Joseph Fourier of Grenoble, France, then professor at the University of Metz, the École Polytechnique, the École Normale Supérieure, Paris, France. He is Silver Professor of Computer Science at the Courant Institute of Mathematical Sciences, New York University, USA. Patrick Cousot is the inventor, with Radhia Cousot, of Abstract Interpretation.

contact : colloquium@lip6.fr

http://www.lip6.fr/colloquium/

Vidéo disponible sur le site

cnrs

LIP6

UPMC SORBONNE UNIVERSITÉS