« Abstract Interpretation–based Formal Verification of Complex Computer Systems »

Patrick Cousot

Jerome C. Hunsaker Visiting Professor
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

cousot@mit.edu www.mit.edu/~cousot

École normale supérieure, Paris, France
Patrick.Cousot@ens.fr www.di.ens.fr/~cousot

Minta Martin Lecture, May 13th, 2005

Abstract

The computing power of computers, which has doubled every eighteen months since 1975, is now so huge that it is possible to embed very large and extremely sophisticated software in ever more complex systems, from small devices to large-scale, interconnected, distributed, real-time systems. This includes the most highly mission-critical and safety-critical computer-based infrastructures, as produced by the aerospace, automotive, customer electronics, defense, energy, industrial automation, medical device, rail transportation and telecommunication industries.

The exponential expansion of software in all application domains leads to the unfortunate situation where software engineers can build increasingly large software, but are less and less confident in the quality of the software they produce. Defaults in such complex software are not so uncommon, as can be experienced everyday by computer end-users. Such bugs can have catastrophic consequences as the most famous, and certainly most costly one, to date, the overflow at the origin of the failure of the Ariane 5.01 flight on 4 June 1996.

Because present-day software engineering, which is almost exclusively manual, with very few useful automated tools does not scale up, a grand challenge is therefore to develop knowledge, methods, technologies and tools to master software complexity.

Mathematical results show that the automatic software verification problem is indeed extremely hard.

Recent progress in the rigorous analysis of software and embedded systems has been possible thanks to abstract interpretation, formalizing the idea of sound approximation of complex mathematical structures, in particular those involved in the semantic models of computer systems. Abstract interpretation can be applied to the systematic construction of methods and effective algorithms to approximate undecidable or very complex problems in computer science such that the semantics, the proof, the static analysis, the verification, the safety and security of software and hardware computer systems.

Abstract interpretation-based static analysis, which automatically infers dynamic properties of computer systems, has been very successfully applied in recent years to automatically verify complex properties of real-time, safety critical, embedded systems, such as the verification of absence of runtime errors in the primary flight control software of commercial airplanes.

The slides will be available after the lecture.