Procedure-Modular Verification of Control Flow Safty Properties

Siavash Soleimanifard
Royal Institute of Technology, KTH
Stockholm, Sweden
siavashs@csc.kth.se

Many software applications are downloaded from remote systems through a network and then executed on a local platform. Such mobile code often originates from untrusted sources and presents a potential threat to the security of the platform. One of the most important challenges that computing research faces today is the development of sound and efficient security mechanisms for ensuring the safe deployment and execution of untrusted mobile code. We have developed a *compositional verification* method based on *maximal model* construction to cope with this challenge.

Our method is for verification of *open systems*, that is, systems where some components are available through their implementation, called internal (e.g., platform), while some other components are given through their *specifications*, called external (e.g., downloaded applications). A specification of a component is a formal description of its essential properties.

The technique we employ is based on the notion of *maximal models* for each given specification. A maximal model is a model that represents all models satisfying an specification. In our approach, maximal models of external components and internal components are constructed, composed and model checked against a control flow global safety property. When a component implementation becomes available, to ensure that the provided specification satisfies its implementation, we model check the specification against a model generated from its implementation.

In my talk, I will introduce ProMoVer, a tool for fully automated procedure–modular verification of Java programs equipped with method–local and global assertions that specify safety properties of sequences of method invocations. The tool is essentially a wrapper around a developed tool set for compositional verification of control flow safety properties, where program data is abstracted away completely. In our tool set, as specification language we have used fragments of LTL and μ–calculus. I will talk about the underlying framework, logics and tools for extracting models from implementation, constructing maximal models, composing models and model checking them. At the end of my talk I will present a demo of a realistic case study verified by ProMoVer.