Storing Dynamically Sized States

Dependable Systems - Dirk Nowotka

Project Description

Model Checking is a well-established verification technique within the formal methods community. The technique has, despite numerous large-scale case studies, yet to obtain mainstream adoption in the industry. This can partially be attributed the fact, that in order to apply Model Checking, one must first convert the system/program that are being verified to a formal model. A conversion that requires expert knowledge of the modelling language and expert knowledge about how to express the properties to be verified. Pushing Model Checking, and formal methods in general, to industry would be easier if the techniques could be applied to their existing language (source code), or an intermediate language which their language can be automatically translated to.

As an answer to the above, the Dependable Systems group have started developing the tool Lodin that, among other things, performs Explicit-State Model Checking of LLVM bitcode. Being a Model Checker, Lodin naturally encounters the same problem as every other Explicit-State Model Checking tool e.g. state space explision problem, but also faces problem very specific to the analysis of source code. One of these is how to represent the varying size states of LLVM-code in a way that is both efficient to work with, but also compact while stored in memory. Furthermore, it should be easy to compare a state we are currently working on with one stored already stored in memory. This project is exactly about implement a state storage model for dynamically sized states. Although this project gets its problem from Lodin, it is not tied to Lodin as such.

A succesull student of this project will, obtain knowledge about Model Checking technology, obtain a deeper understanding of the difficulties related to efficiently representing states. The student will also develop/implement a storage mechanism and evaluate its efficiency.

Applicable For

Bachelorstudents	☑
Masterstudents	☑

Skillset

Programming	●●●●○

Keywords

State Space Compression
Model Checking

Contact

Danny Bøgsted Poulsen
@ dbp@informatik.uni-kiel.de