Universität Bamberg

Fakultät Wirtschaftsinformatik und Angewandte Informatik

Kolloquiumsvortrag

Donnerstag, 07. Dezember 2017, 16:00 Uhr, Raum WE5/02.020

Polymorphic Behaviour Inference for Deadlock Checking

Prof. Dr. habil. Martin Steffen, University of Oslo

Deadlocks are a common problem for concurrent programs with shared resources. The inherent non-determinism make deadlocks, as other errors in the presence of concurrency, hard to detect and to reproduce. We present a static analysis using behavioral effects to detect deadlocks in a higher-order concurrent calculus. Deadlock freedom, an important safety property for concurrent programs, is a thread-global property, i.e., the blame for a deadlock in a defective program cannot be put on a single thread, it is two or more processes that share responsibility; the somewhat atypical situation, where a process forms a deadlock with itself, cannot occur in our setting, as we assume re-entrant locks. The approach presented in this paper works in two stages: in a first stage, an effect-type system uses a static behavioral abstraction of the codes' behavior, concentrating on the lock interactions. To analyze the consequences on the global level, in particular for detecting deadlocks, the combined individual abstract thread behaviors are explored in the second stage.

Two challenges need to be tackled to make such a framework applicable in practice. For the first stage on the thread local level, the static analysis must be able to derive the abstract behavior, not just check compliance of the code with a user-provided description. This is the problem of type and effect /inference/. As usual, the abstract behavior needs to over-approximate the concrete one which means, concrete and abstract description are connected by some /simulation/ relation. For the second stage, exploring the (abstract) state space on the global level, obtaining /finite/ abstractions is crucial. In our setting, there are four principal sources of infinity: the calculus, 1) allowing recursion, supports 2) dynamic lock creation, 2) with re-entrant locks, the lock counters are unbounded, and 4) dynamic thread creation. Our approach offers sound abstractions for the mentioned sources of unboundedness, except the last point.

Kontakt: Prof. M. Mendler: michael.mendler@uni-bamberg.de