

Pun, Ka I; Steffen, Martin; Stolz, Volker

Deadlock checking by data race detection. (English) [Zbl 1434.68112](#)

Arbab, Farhad (ed.) et al., Fundamentals of software engineering. 5th international conference, FSEN 2013 Tehran, Iran, April 24–26, 2013. Revised selected papers. Berlin: Springer. Lect. Notes Comput. Sci. 8161, 34-50 (2013).

Summary: Deadlocks are a common problem in programs with lock-based concurrency and are hard to avoid or even to detect. One way for deadlock prevention is to statically analyze the program code to spot sources of potential deadlocks.

We reduce the problem of deadlock checking to race checking, another prominent concurrency-related error for which good (static) checking tools exist. The transformation uses a type and effect-based static analysis, which analyzes the data flow in connection with lock handling to find out control-points that are potentially part of a deadlock. These control-points are instrumented appropriately with additional shared variables, i.e., race variables injected for the purpose of the race analysis. To avoid overly many false positives for deadlock cycles of length longer than two, the instrumentation is refined by adding “gate locks”. The type and effect system and the transformation are formally given. We prove our analysis sound using a simple, concurrent calculus with re-entrant locks.

For the entire collection see [[Zbl 1320.68009](#)].

MSC:

- [68N30](#) Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.) Cited in 1 Document
- [68Q85](#) Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.)

Full Text: [DOI](#)