Vercors – Verification of Concurrent Data Structures

Modern software applications are making more and more demands on the speed of the underlying hardware. In the last decades, hardware speed has increased significantly, but the limits seem to have been reached: because of physical restrictions, it is impossible to increase processor speed further.

An obvious solution is to use *multi-core* processors, where several units compute simultaneously. To make optimal use of such a multi-core processor, the software running on it should describe which computations can be done in parallel. However, a major problem with such software is that it is highly error-prone: *all* possible interactions between the computations have to be considered to be sure that an application functions correctly. For a programmer it is difficult not to overlook any interaction, and thus to ensure that errors can never happen.

Within the field of formal methods, for many years methods have been developed to reason about the correctness of programs. So-called program logics are used to show that a program behaves as it should under all circumstances. For sequential programs, powerful tools are available to support this reasoning, but for multithreaded programs, development of such methods is still an open problem. Nevertheless, in the last 5 years significant progress has been made, in particular by the use of a special program logic (called *separation logic*), that allows to specify in a natural way when program fragments may and may not influence each other.

Marieke Huisman has used this logic to reason about multithreaded programs written in the programming language Java. Within the VerCors project, this technique will be further extended and generalised, in order to make it applicable to other programming languages as well. Since other programming languages use different technique to ensure that parallel computations correctly synchronise with each other, this mean that the verification method also has to be extended with different synchronisation techniques.

A specific application domain for the project is the correctness of data structures that can be used in a multithreaded context. A data structure is a general structure, used to store data. Since data structures can be reused in many different applications, it is of paramount importance to implement them correctly. Within the VerCors project the multithreaded behaviour of such data structures will be specified and verified. The specifications will be stated in such a way that they are independent of the particular synchronisation mechanism used, thus ensuring that an implementation can easily be exchanged with another, possibly more efficient, implementation, provided it has the required behaviour.

All results of the project will be incorporated in a tool that will make it possible to automatically verify correctness properties of multithreaded programs.