Induction and Co-Induction in Sparkle
Thesis number 550

Leonard Lensink
Colloquiumkamer HG00.308

August 25th 2006, 9:30 am

Abstract

Developing correct software remains one of the most important subjects in computer science. Bugs can be costly and annoying. One of the most secure ways to eliminate errors in computer programs is by proving them to be correct in a mathematical context. Sparkle is a proof assistant that helps programmers with constructing mathematical proofs about algorithms written in Clean or any other functional language. Proofs on programs that use one of the most important programming techniques, called recursion, usually need proof principles called inductive and co-inductive reasoning.

Support for these mathematical proof steps were limited within Sparkle. In order to support for reasoning about a larger class of programs we have extended the proof techniques in Sparkle.

A method that supports mutually reasoning on mutually recursive types has been added. A new method has been devised that allows for the derivation of an induction scheme from function definitions. For co-inductive reasoning, a tactic was added based on bisimulation relationships. Another speculative tactic was devised that derives a co-inductive proof principle from a function definition.

In this thesis these four methods and their implementation are presented.