Reasoning about Programs Using a Scientific Method

Peter W. O'Hearn

Queen Mary University of London

Abstract. Reasoning about programs has traditionally been done using deductive reasoning, where mathematical logic is used to make proofs that connect programs with specifications. In this talk I describe an approach where an automated reasoning tool approaches program code as a scientist would in the natural world. Instead of just deductive logic, versions of abductive reasoning (generation of new hypotheses) and inductive generalization are used in an iterative fashion to discover specifications that partly describe what programs do, starting from bare code. The resulting specifications are partial or conservative, but the inference/discovery aspect makes it much easier to approach large code bases, quickly, than with the traditional deductive-only approach.

The underlying program logic in this work is separation logic, a logic for reasoning about the way that programs use computer memory, and the inference method attempts to discover a logical assertion describing the program's footprint: the collection of cells that it touches. Aiming for the footprint provides a strategy to select compact specifications, amongst the enormity of all potential specifications (which would be too many to consider). After describing the inference techniques, I report on experience using a software tool that automates the method, which has been applied to large code bases.

This talk is based on joint work with Cristiano Calcagno, Dino Distefano and Hongseok Yang.

S. Qin and Z. Qiu (Eds.): ICFEM 2011, LNCS 6991, p. 14, 2011.