A Method of Program Refinement

Jim Grundy

Technical Report: 318

ABSTRACT

A method of specifying the desired behaviour of a computer program, and of refining such specifications into imperative programs is proposed. The refinement method has been designed with the intention of being amenable to tool support, and of being applicable to real-world refinement problems.

Part of the refinement method proposed involves the use of a style of transformational reasoning called 'window inference'. Window inference is particularly powerful because it allows the information inherent in the context of a subexpression to be used in its transformation. If the notion of transformational reasoning is generalised to include transformations that preserve relationships weaker than equality, then program refinement can be regarded as a special case of transformational reasoning. A generalisation of window inference is described that allows non–equivalence preserving transformations. Window inference was originally proposed independently from, and as an alternative to, traditional styles of reasoning. A correspondence between the generalised version of window inference and natural deduction is described. This correspondence forms the basis of a window inference tool that has been built on top of the HOL theorem proving system.

This dissertation adopts a uniform treatment of specifications and programs as predicates. A survey of the existing approaches to the treatment of programs as predicates is presented. A new approach is then developed based on using predicates of a three-valued logic. This new approach can distinguish more easily between specifications of terminating and nonterminating behaviour than can the existing approaches.

A method of program refinement is then described by combining the unified treatment of specifications and programs as three-valued predicates with the window inference style of transformational reasoning. The result is a simple method of refinement that is well suited to the provision of tool support.

The method of refinement includes a technique for developing recursive programs. The proof of such developments is usually complicated because little can be assumed about the form and termination properties of a partially developed program. These difficulties are side-stepped by using a simplified meaning for recursion that compels the development of terminating programs. Once the development of a program is complete, the simplified meaning for recursion is refined into the true meaning.

The dissertation concludes with a case study which presents the specification and development of a simple line-editor. The case study demonstrates the applicability of the refinement method to real-world problems. The line editor is a nontrivial example that contains features characteristic of large developments, including complex data structures and the use of data abstraction. Examination of the case study shows that window inference offers a convenient way of structuring large developments.