Proving Program Refinements and Transformations

Martin Ward
St. Annes College Oxford

D.Phil Thesis
June 1989

Abstract

In this thesis we develop a theory of program refinement and equivalence which can be used to develop practical tools for program development, analysis and modification. The theory is based on the use of general specifications and an imperative kernel language. We use weakest preconditions, expressed as formulae in infinitary logic to prove refinement and equivalence between programs.

The kernel language is extended by means of "definitional transformations" which define new concepts in terms of those already present. The extensions include practical programming constructs, including recursive procedures, local variables, functions and expressions with side-effects. This expands the language into a "Wide Spectrum Language" which covers the whole range of operations from general specifications to assignments, jumps and labels. We develop theorems for proving the termination of recursive and iterative programs, transforming specifications into recursive programs and transforming recursive procedures into iterative equivalents. We develop a rigorous framework for reasoning about programs with **exit** statements that terminate nested loops from within; and this forms the basis for many efficiency-improving and restructuring transformations. These are used as a tool for program analysis and to derive algorithms by transforming their specifications. We show that the methods of top-down design and program verification using assertions can be viewed as the application of a small subset of transformations.

Acknowledgements

I would like to thank Dr.A.W.Roscoe for agreeing to be my supervisor.
Special thanks go to my wife Katherine for her support and encouragement
throughout this period.

To Katherine.