Abstractions Before Proofs

Cliff Jones
Newcastle University
Newcastle upon Tyne, UK

Abstract

Proving that programs satisfy their specifications can benefit enormously from tool support but theorem proving tools can also constrain a user's thinking. This talk argues that, for large or complex programs, it is layers of abstraction that make or break the comprehensibility of developments.

However powerful a theorem proving tool is, it will make little long-term contribution to the understanding of programs if the user is forced to bend their steps of development to fit the tool. Abstraction is essential to achieve separation of issues and to help in the understanding of complex systems. The formalism chosen governs the difficulty of completing detailed proofs that can be verified with mechanically checkable rules.

This talk will emphasize abstractions and techniques for reasoning about the development of concurrent programs. In conclusion, the argument will turn to positive recommendations for tool developers.