

Abstract

We give a semantically based axiomatic treatment of partial correctness and deadlock for an imperative shared variable parallel programming language. The Owicki-Gries proof methodology for this language proves conventional Hoare-style partial correctness assertions and involves the notion of interference-freedom of proofs (to guarantee soundness), auxiliary variables (to guarantee relative completeness), and global invariants (to permit reasoning about deadlock-freedom). Our axiomatic proof system is based more explicitly on the underlying operational semantics, using assertions whose syntactic structure directly reflects the operational behavior of parallel programs at an appropriate level of abstraction. We build a proof system that requires neither interference-freedom nor auxiliary variables. Novel features include the use of a syntactic form of parallel composition of assertions, and the use of conjunction and implication as connectives on assertions. It is possible simultaneously to reason about partial correctness and deadlock-freedom using our proof system, without recourse to global invariants. We discuss some non-trivial examples, and compare our proof methodology with some other proof methods from the literature.