

Abstract

We consider a parallel Algol-like language, combining procedures with shared-variable parallelism. Procedures permit encapsulation of common parallel programming idioms. Local variables provide a way to restrict interference between parallel commands. We provide a denotational semantics for this language, simultaneously adapting “possible worlds” [Rey81, Ole82] to the parallel setting and generalizing “transition traces” [Bro93] to the procedural setting. This semantics supports reasoning about safety and liveness properties of parallel programs, and validates a number of natural laws of program equivalence based on non-interference properties of local variables. The semantics also validates familiar laws of functional programming. We also provide a relationally parametric semantics, generalizing [Bro93] to permit reasoning about relation-preserving properties of programs, and adapting work of O’Hearn and Tennent [OT95] to the parallel setting. This semantics supports standard methods of reasoning about representational independence, adapted to shared-variable programs. The clean design of the programming language and its semantics supports the orthogonality of procedures and shared-variable parallelism.