

ON THE AXIOMATIC TREATMENT OF CONCURRENCY

Stephen D. Brookes
Carnegie-Mellon University
Department of Computer Science
Schenley Park
Pittsburgh

0. Abstract.

This paper describes a semantically-based axiomatic treatment of a simple parallel programming language. We consider an imperative language with shared variable concurrency and a critical region construct. After giving a structural operational semantics for the language we use the semantic structure to suggest a class of assertions for expressing semantic properties of commands. The structure of the assertions reflects the structure of the semantic representation of a command. We then define syntactic operations on assertions which correspond precisely to the corresponding syntactic constructs of the programming language; in particular, we define sequential and parallel composition of assertions. This enables us to design a truly compositional proof system for program properties. Our proof system is sound and relatively complete. We examine the relationship between our proof system and the Owicki-Gries proof system for the same language, and we see how Owicki's parallel proof rule can be reformulated in our setting. Our assertions are more expressive than Owicki's, and her *proof outlines* correspond roughly to a special subset of our assertion language. Owicki's parallel rule can be thought of as being based on a slightly different form of parallel composition of assertions; our form does not require *interference-freedom*, and our proof system is relatively complete without the need for auxiliary variables. Connections with the "Generalized Hoare Logic" of Lamport and Schneider, and with the Transition Logic of Gerth, are discussed briefly, and we indicate how to extend our ideas to include some more programming constructs, including conditional commands, conditional critical regions, and loops.