KEEPING A FOOT ON THE GROUND
(A Position Paper)

Brent Hailpern
IBM T. J. Watson Research Center
Yorktown Heights, New York 10598

Much of the progress in the field of programming logics lies in creating new logic systems. How should new systems be judged? If the work presented at this conference is any indication, then logics should be judged by their power (completeness). For example, logic A is better than logic B if logic A can prove everything that logic B can and more. Comparing logics in this manner yields results that may provide insight into the fundamental aspects of programming---just as the equivalence of Turing machines, Post machines, recursively enumerable sets, etc. is fundamental to the study of mechanical computability. I propose, however, that an additional criteria be used to judge the quality of a new programming logic: how easy is it to prove the correctness of programs with this logic?

In this proposal I echo Pnueli and Ben-Ari's (1) proposal to develop "a corpus of formal proofs that can then serve as a body of experimental data upon which further theorizing can be done". They hope that the comparison of two logics would not be that "our logic is more expressive than your logic", but rather that "our proof of algorithm X is more elegant than yours". In other words, a feature that makes a logic more powerful but that confuses the user (programmer, system designer, logician, etc.) is not desirable.

I am not calling for stopping research into more powerful logics---far from it. I do urge, however, that research projects that have devoted all of their effort to developing new logics, should consider trying to verify some "non-trivial" programs with their logics.

There are many domains in the realm of computer science that need the insight that verification can give: network protocols, resource allocation, hardware, and security are examples. The scientists in these fields are highly intelligent individuals, but we cannot expect them to take their time to learn all of our theories in order to decide which is appropriate to their field. Instead, some of us can look for domains---simple areas at first---that are amenable to our techniques. Not only would this benefit those in the field of application, but it might point out some strengths and weaknesses of our techniques.

REFERENCE

(1) Mordechai Ben-Ari and Amir Pnueli. Temporal logic proofs of concurrent programs. Submitted for publication, November, 1980.