EDITORIAL BOARD

Dear Colleague,

The *IEEE Standard for the Scheme Programming Language* was approved last December and is now available from IEEE as Std 1178-1990. Apparently this standard has also been approved by ANSI as an American national standard. I wish to congratulate and thank the working group and especially its chair, Chris Haynes, and the editors, David Bartley, Chris Hanson, and James Miller, for the harmonious and relatively swift progress of this first (but not the last!) formal standard for a dialect of Lisp.

In this issue of *Lisp Pointers* we are publishing the *Revised[4] Report on the Algorithmic Language Scheme*, which is the document on which the IEEE standard was based. There are quite a few differences between the IEEE standard and the Revised[4] Report, so this issue of *Lisp Pointers* cannot substitute for a copy of the standard. Anyone who is implementing Scheme or is writing portable Scheme code should certainly obtain and read IEEE Std 1178-1990. For example, the Revised[4] Report does not explicitly say what will happen if a program uses a top-level definition to bind a variable such as `car` that is already bound in the standard initial environment. The IEEE standard explicitly says that such definitions are permitted, and that they do not affect the behavior of any standard procedure.

The IEEE standard is less wishy-washy than the Revised[4] Report. The IEEE standard allows programs to use internal definitions; the Revised[4] Report says only that some implementations support them. The IEEE standard requires that the empty list be distinct from `#f` (the boolean value that represents false), and requires that the empty list count as true rather than false in boolean contexts; the Revised[4] Report does not specify whether the empty list counts as true or false, and does not even require that the empty list be distinct from `#f`. In both of these examples the Revised[4] Report is imprecise out of respect for the opinions of minorities. Some of the report's authors regard internal definitions as an undesirable irregularity in the language, and some of the report's authors feel that implementations should be free to represent the empty list by any object that is not a pair. The Revised[4] Report is a consensual document that reflects the sometimes quirky philosophies of its authors. The IEEE standard is a democratic document that reflects the practical need for standardization.

Both documents are conservative, but I hope that the revised reports will become less conservative now that the IEEE standard has been adopted, so that the revised report no longer has to act as an informal standard. As the IEEE standard itself explains:

> Future revisions to this standard may be anticipated by a series of informal language specifications collectively known as the "Revised Reports on the Algorithmic Language Scheme" ... These reports extend the stan-

dard language to include features that are considered mature enough to merit widespread implementation and experimentation. Addition of such a feature to the report suggests that Scheme implementors should consider the addition and carefully consider it as a candidate for formal standardization.

The appendix to the Revised[4] Report is an example of such an extension. The hygienic macro technology described there merits widespread use, but is not yet ready to be included in a formal standard. The binding constructs and pattern language described in that appendix seem right, and I do not expect them to change very much in revised reports of the future. I do expect the low-level macro facility that is described in that appendix to evolve considerably, or to be replaced by something altogether different. Two alternative low-level macro facilities will be reported in the next issue of *Lisp Pointers*.

Sincerely,

William Clinger

William Clinger

Guest Editor