The Key of Lisp

When asked to do this department, I didn't quite catch the fact that I would be required to produce two columns before I could expect some 'letters to the editor'. I do hope that there are some out there because there isn't much here that I haven't talked about before.

Why can't LISP read files?

Of course LISP can read files, the heading was just to get your attention. There is a problem here, however; different implementations of LISP read files in different manners. This is further confused by the existence of LISPs that allow for READ-MACROS with different implementations. All of this causes problems when programs are converted from one system to another.

There are problems with programs that read files that appear to contain text because of the way special characters are handled and the interpretation that is made for special forms like floating point numbers. Imagine the surprise of the programmer who has been running a program for some time that reads a control code from a file and suddenly discovers that the code '23E7' is changed to 230,000,000. This could not occur, naturally, if the file had been written by a LISP program, but as the use of LISP advances into the real world, programmers tend to read files that were created by programs written in other languages.

Other problems arise with reading files that contain numbers that are represented in the internal form of some computer (even the one that the LISP program is running on). Recently I got caught with the need to process information from a hexadecimal dump of a file. That was interesting because 'ADBC' is a symbol, '1230' is an integer, '12E5' is a floating point number, and '12F0' is invalid!

This problem is only going to get worse until a standard is reached for the language.

...becoming a standard

How often have you seen this phrase in the advertising of your favorite LISP vendor? Windowing systems, graphics systems, and object-oriented extensions are all touted as becoming to be accepted as a standard in the industry. I think that there is another article in this issue about what is required for something to become a standard, but I would urge anyone who reads something about a standard to ask the vendor if he/she is represented on the X3J13 Committee for standardizing LISP because at the moment **there is no standard.** Not all of the vendors who are claiming to have the straight path into the standard are even represented on the committee.

The committee is open to anyone, but all of the anyones have to apply and be active on the committee. If you can't find out elsewhere how to get on the committee, please give me a call and I will send you the information.

Languages vs. Environments

Despite all of the talk about LISP, there are few actual implementations of the language by itself. Some people perceive the real value to be the environment that surrounds LISP rather than the language itself (ignoring the possible connection between the architecture of the language and the ability to create environments). Except on PC's and a few older implementations, no one ***writes programs in LISP***, they write programs using a complete environment that is built on and for the LISP language.

Is there any value for a standard for the language by itself without a standard for the programming environment that goes along with it? Is the community lying to itself by trying to set a standard for the language? On the other hand, is it possible to describe the environment in a machine independent manner? This was certainly a problem with SMALLTALK; some of the definition of the language was specific to the machines that were used for the first implementation. The experience of the people who were finally able to port the language to other machines might be of value in answering the question about the possibility of defining an environment for LISP, if it should be decided to do that.

Another problem is to try to list what would have to be in a complete environment. Certainly editors are needed and that already causes problems because different groups prefer different editors. As more complex parts of the environment are considered, I suspect that there would be more controversy about the proper definition to use. This is to consider the problem strictly from the vantage point of the LISP programmer. There is a lot of code in existence that is written in other languages and the LISP programmer is going to have to deal with that as well. I suspect that the more sophisticated the LISP environment becomes, the more difficult it will be to deal with existing code in other languages.

Teaching LISP

If there a problem with the way that LISP is taught in the Universities? It has been suggested that LISP is only used in the AI courses with a very few exceptions. Why isn't it more heavily used for the first course in programming, for data structures, or any other courses not directly related to AI? Perhaps there are a large number of Universities that use LISP for teaching these types of courses. Please let me know and I will compile a list and publish it. If you do use LISP in teaching, do the students have access to a LISP interpreter/compiler or an environment? If you are the instructor for such a class, would you be willing to share some of your teaching materials with others? I will also include that information according to the responders wishes.

The examples change, but it appears to me that most of the problems with the commercial use of the language are due to the lack of a standard. Another problem may be with the definition of and teaching of the concept of a full programming environment. I would like to get comments from the readers of this column.

Despite the fond expectations of all of my supporters, I am not omniscient. I accept that there might be other objections to the use of LISP in the real world. If there are, please send me your cards and letters and I will see if I can address the problems. My address is: Susan Ennis, Amoco Production Company, Box 3588, Tulsa, OK 74120.