LISP

LISP is an acronym for List Processing Language, a high-level programming language developed by John McCarthy at MIT in the late fifties and early sixties. LISP is still very popular for Artificial Intelligence applications and is, together with FORTRAN and COBOL, one of the oldest programming languages.

LISP is a functional language; this means that computations are performed by defining functions and applying them to data and also to other functions. Each function application requires a set of parenthesis and therefore extensive programs tend to end in long strings of closing parenthesis. The main data structures in LISP are lists, which can contain atomic elements or other lists. Nested lists provide a representation for trees. LISP programs themselves are also lists. Therefore, it is fairly easy to write LISP programs that process other LISP programs. This is the main difference to languages like FORTRAN or Pascal, which are not geared towards this kind of “reflective” computing.

At the Darmouth Summer Research Project on Artificial Intelligence, a seminal conference for the history of AI held in 1955, McCarthy formulated the line of research he intended to follow and his thoughts about a new programming language: “It therefore seems to be desirable to attempt to construct an artificial language which a computer can be programmed to use on problems and self-reference. ... I hope to try to formulate a language having these properties ... with the hope that using this language it will be possible to program a machine to learn to play games well and do other tasks.” McCarthy’s first approach was extending FORTRAN with list processing functions, but very soon he was developing a new language. Looking for a way to describe a “universal function” capable of evaluating any other function, he finally hit on the idea of using list processing, the functional notation, and list processing instructions. Oddly enough, some of them like the CAR (take the first element of a list) and CDR (take the rest of a list by deleting the first element) commands took their names from similar assembler instructions in IBM machines. The canonical version of LISP was defined in 1962 by McCarthy in his book about LISP 1.5.

Mac-Lisp, a version of LISP written at MIT for the PDP-10 computer became very popular and contributed to the disemination of the language. However, LISP was still much slower than other more conventional languages and so the idea arose of building LISP machines, in order to lower the “semantic gap” between the software and hardware. In the early 1980s, some companies were formed that tried to build those LISP machines. The more successful of them was Symbolics Inc., which stayed in business almost ten years. However, advances in compiler technology and faster workstations eventually made LISP machines obsolete. It was interesting though, that all system code for those machines was written also in LISP. In the framework of the Japanese Fifth Generation Project (1981-1991) Prolog machines were built, whose design philosophy owed much to the LISP machines projects.

The very success of LISP led to several incompatible versions for many different computers. Therefore at an ARPA-sponsored meeting in 1981, several of the main LISP developers met and formed a committee to explore the possibility of agreeing on a common language. The definition of Common LISP was published in 1984 in a book titled “Common Lisp: the

Language”. Several software companies immediately implemented the new version of the language.

LISP continues to be, until this day, the favorite programming language of AI researchers.

Raúl Rojas

References

McCarthy, John, *LISP 1.5 Programmer's Manual*, Massachusetts Institute of Technology, Computation Center and Research Laboratory of Electronics, 1962.
Steele, Guy L., *Common Lisp: the Language*, Digital Press, Burlington, Mass.,1984.