McCarthy, LISP, & λ-Calculus

LISP History according to McCarthy's memory in 1978. Presented at the ACM SIGPLAN History of Programming Languages Conference, June 1-3, 1978. It was pubished in **History of Programming Languages**, edited by Richard Wexelblat, Academic Press 1981. ***Two quotations:***

I spent the summer of 1958 at the IBM Information Research Department at the invitation of Nathaniel Rochester and chose differentiating algebraic expressions as a sample problem. It led to the following innovations beyond the FORTRAN List Processing Language:

• • • •

(c) To use functions as arguments, one needs a notation for functions, and it seemed natural to use the λ-notation of Church (1941). I didn't understand the rest of his book, so I wasn't tempted to try to implement his more general mechanism for defining functions. Church used higher-order functionals instead of using conditional expressions. Conditional expressions are much more readily implemented on computers.

• • • •

Logical completeness required that the notation used to express functions used as functional arguments be extended to provide for recursive functions, and the LABEL notation was invented by Nathaniel Rochester for that purpose. D. M. R. Park pointed out that LABEL was logically unnecessary since the result could be achieved using only λ — by a construction analogous to Church's Y-operator, albeit in a more complicated way.

Other key McCarthy publications:

Recursive Functions of Symbolic Expressions and their Computation by Machine (Part I). The original paper on LISP from **CACM**, April 1960. Part II, which never appeared, was to have had some Lisp programs for algebraic computation.

A Basis for a Mathematical Theory of Computation, first given in 1961, was published by North-Holland in 1963 in **Computer Programming and Formal Systems**, edited by P. Braffort and D. Hirschberg.

Towards a Mathematical Science of Computation, IFIPS 1962 extends the results of the previous paper. Perhaps the first mention and use of ***abstract syntax***.

Correctness of a Compiler for Arithmetic Expressions with James Painter. May have been the first proof of ***correctness of a compiler***. Abstract syntax and Lisp-style recursive definitions kept the paper short.

An HTML site concerning Lisp history can be found at:

http://www8.informatik.uni-erlangen.de/html/lisp-enter.html