IVERSON81
Kenneth E. Iverson & Adin D. Falkoff, 1981: The Evolution of APL, in History of Programming Languages, Academic Press, 1981, pp. 661-691 What has made APL a programming language? Why and how was it designed? What factors have influenced its design? The best person to answer these questions is Ken Iverson (APL's designer) and the best answer he can give is found in this paper.

ABSTRACT This paper discusses the evolution of APL by dividing its progress into four phases according to the major use or preoccupation of the period. These four phases are: academic use (1957-1960), machine description (1961-1963), implementation (1964-1968), and system (1968-now). The main influence in the first phase of its development came from the designer's mathematical background. This mathematical influence shows itself in a variety of ways, notably in the use of functions with explicit arguments and results, use of logical functions and variables, use of concepts and terminology from tensor analysis (inner product, outer product, rank etc.), the emphasis on generality (generalized summation [F/], inner product [F.G] etc) and the emphasis on identities. The idea of an OPERATOR (something that applies to a function or functions and produces a new function) was from the idea of derivative operator and convolution operator in mathematics. The next phase was the machine description phase. This phase was marked by the complete or partial description of a number of computer systems, especially APL/360. Many aspects of the language were consolidated and regularized. For example, the embracing symbols (such as |x| for absolute value) were replaced by the leading symbol only, thus unifying the syntax for monadic functions. The requirement for parenthesis surrounding an expression involving a relation was eliminated. The right-to-left order of execution and reduction were enforced. In 1964, the implementation of the language was seriously considered. The first step was to define a character set for APL. The choice of an 88-character set exercised a significant degree of influence on the development of the language. It led to the linearization of the language (with no subscripts or superscripts) as well as the important notion of composite characters formed by striking one of the basic character over another. It also led to a more systematic exploitation of the notion of ambiguous valence (the representation of both a monadic and cyadic function by the same symbol). Other questions were considered, including the formal definition of functions, the localization and scope of names, and the use of tolerance in comparisons and in printing output. In the system phase, the use of the language was extended to provide communication with the outside world. The major change to the language was the addition of shared variables, system variables and system functions. In reviewing the history of APL, the authors have clearly identified two most important influences: its mathematical and data processing orientation. APL was designed as a thinking language with the goal of consistency, generality, and conciseness in all its constructs and expressions.

Conclusion As a conclusion, this paper gives one an excellent insight into the history and design goal of APL. One's understanding of APL can be very much deepened by reading this paper.