IVERSON80 Iverson, K.E.; Notation as a Tool of Thought; CACM 23, 8 (August 1980) pp. 444-465.

This paper is concerned with the characteristics necessary for a good notation. Kenneth Iverson derives many of these characteristics, though not all, from mathematical notation and exemplifies them by the use for the programming language APL. His theses, in this paper, is that advantages of universality and executability (which he claims are not found in mathematical notation) can be combined in one coherent language with the advantages offered by mathematical notation. The tone of the paper is indicated by him with the following quote:

> By relieving the brain of all unnecessary work,
> a good notation sets it free to concentrate on
> more advanced problems, and in effect increases
> the mental power of the race.
> A.N. Whitehead

Iverson stresses the following mathematical characteristics for any notation.

Ease of expressing constructs arising in problems: A notation must be flexible enough to allow convenient expression of notions arising directly from a problem as well as those arising from the analysis, generalization, and specialization of the problem.

Suggestivity: A suggestive notation is such that the forms of expressions in one set of problems suggests related expressions which can be applied in other problems. Iverson then cites examples of this in APL. For example, exponentiation written as N*M is equivalent to x/M N as multiplication written as NxM is equivalent to +/M N.

Subordination of detail: Brevity facilitates reasoning and brevity is achieved by subordination of detail. Iverson mentions three ways of doing this: the use of arrays, the assignment of names to functions and variables, and the use of operators.

Economy: Economy of notation requires that a large number of ideas be expressible in terms of a relatively small vocabulary and simple grammatical rules. One example in APL, is the sum of the first N natural numbers which is given as +/ N. This expression is made up of the function iota ( ) that produces a vector of integers from one to N, and the function +/ which sums the elements of a vector. Note that even this function, +/, is constructed from the more general notion of a reduction operator, /, applied to the addition operator, +.

Amenability of formal proofs: The importance of formal proofs is obvious. Certain types of proofs are greatly facilitated by notation such as APL. Proof by exhaustion, as in the case of truth tables, is simply a matter of using the outer product operator of APL. Proof by a sequence of identities and proof by induction are also facilitated in this notation by the use of identities which can be established by applying mathematical laws such as commutativity, associativity, etc, or by proofs by exhaustion.

In conclusion, this paper puts forth some of the important aspects for a good notation and uses APL to illustrate these aspects. The paper itself is probably not nearly as important in the evolution and development of programming languages as the language APL itself and the concepts and ideas that it has brought to light for programming language design. The paper, however, by categorizing some of the characteristics which the author considers good notation, does offer an insight into the view taken in the design of APL and perhaps into the view that should be taken in the design of other languages.