

RC 7091 (#30399) 4/26/78
18 pages

Research Report

OPERATORS AND FUNCTIONS

Kenneth E. Iverson
APL Design Group
Research Division
Yorktown Heights



Research Division
San Jose Yorktown Zurich

RC 7091 (#30399) 4/26/78
18 pages

OPERATORS AND FUNCTIONS

Kenneth E. Iverson
APL Design Group
Research Division
Yorktown Heights

Abstract

This paper treats the syntax of operators in APL, presents definitions of a number of new operators and functions, and provides examples of their use.

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication elsewhere and has been issued as a Research Report for early dissemination of its contents. As a courtesy to the intended publisher, it should not be widely distributed until after the date of outside publication.

Copies may be requested from:
IBM Thomas J. Watson Research Center
Post Office Box 218
Yorktown Heights, New York 10598

OPERATORS AND FUNCTIONS

Kenneth E. Iverson
APL Design Group
Research Division
Yorktown Heights

Abstract

This paper treats the syntax of operators in APL, presents definitions of a number of new operators and functions, and provides examples of their use.

This paper presents some of the operators and functions which I have found useful in treatments of topics in mathematics and in data processing. Readers are assumed to be familiar with APL as defined in Reference [1]. I am indebted to a number of my colleagues for useful discussions and suggestions, particularly to A.D. Falkoff and D.L. Orth. I am also indebted to Michael Halpern for discussions leading to the formulation of the axis operator.

To allow functions to be defined as conveniently as variables we will permit the use of the assignment arrow $\bar{\leftarrow}$ as follows: a name followed by $\bar{\leftarrow}$ followed by an expression which produces a function, assigns the name to the function. Thus $F\bar{\leftarrow}.x$ and $G\bar{\leftarrow}$, and $H\bar{\leftarrow}G$ are valid expressions.

Operators may take numeric, character, and even operator, arguments as well as functions. For example, the slash (/) applied to a dyadic function (as in +/) produces the monadic reduction function, and applied to a Boolean vector (as in U/) produces the monadic selection function called compression. Numeric operator arguments introduce the possibility of two juxtaposed vector constants in a valid expression. The ambiguity inherent in such juxtaposition must be resolved by the use of parentheses.

Zero-origin indexing will be used throughout this paper.

1. SYNTAX AND ORDER OF EXECUTION

A function produced by an operator is called a derived function. Defined and derived functions behave like primitive functions in that they may serve as arguments to operators and may be divalent. Although derived functions may be divalent, the operators themselves are not. For

example, reduction (/) and scan (\) are monadic and the dot (in inner and outer product) is dyadic. The jot (°) as used in the outer product (°.×) produces, in effect, a special "monadic case" of a dyadic operator. It will be used extensively in this manner in the definition of new operators.

Since a monadic function applies to the argument to its right, and a monadic operator applies to the argument to its left, operators and functions behave in a syntactic sense as mirror images. This behavior will be carried through in the rules laid down for the order of execution.

The order of execution of functions in an unparenthesized expression is governed by a single rule: the right argument of any function (monadic or dyadic) is the result of the entire expression to the right of it. An operator (monadic or dyadic) applies in a similar manner to the left except that its left argument is the result of the longest possible operator sequence on its left. An operator sequence is a portion of an expression consisting of a non-operator followed by a sequence consisting of monadic operators or of dyadic operators each followed by one non-operator. The non-operators may themselves result from parenthesized expressions. Finally, operators enjoy precedence over functions in the sequence of execution, and obey parentheses in the usual manner.

Composite symbols formed from the dieresis (¨) and the overbar (¯) will be reserved exclusively for operators, but operator symbols will not be limited to this class. The symbols $_$ and $\bar{_}$ denote infinity and negative infinity, respectively.

2. POWER AND IDENTITY OPERATORS

The power operator, denoted by $\ddot{*}$, applies to a monadic function left argument F and an integer right argument K to produce the K th power of F in the following sense: $F\ddot{*}K \leftrightarrow F\ddot{*}K-1$, and $F\ddot{*}1 \leftrightarrow F$. In particular, $F\ddot{*}0$ is the identity function and $F\ddot{*}-1$ is the inverse of F . Moreover, $F\ddot{*}_$ denotes the limit of F , that is, the limiting function $F\ddot{*}N$ for N large. Similarly, $F\ddot{*}\bar{_}$ denotes the limit of the inverse of F .

The expression $F\ddot{*}\circ$ yields the identity function of a dyadic function F , defined to be the function G with the following properties: $A\ G\ 1 \leftrightarrow A$, and $A\ G\ 0$ yields the identity element of F . Thus \times is the identity function of $+$ and $*$ is the identity function of \times , properties which account for the utility of expressions of the form $B+. \times L$ and $B \times . * L$ in applying the functions $+$ and \times over subsets of the elements of a vector B specified by the Boolean array L .

For example, if $L \leftarrow (K\rho 2) \tau_1 2 * K \leftarrow \rho B$, then $B + . * L$ and $B \times . * L$ give sums and products over all subsets of B . Moreover, $B \downarrow . (\downarrow * \circ) L$ and $B \uparrow . (\uparrow * \circ) L$ give minima and maxima over all subsets of B . Finally, $X \vee . (\vee * \circ) L$ and $X \wedge . (\wedge * \circ) L$ clearly apply the functions \vee and \wedge over all subsets of X , whereas the equivalent expressions $X \vee . \wedge L$ and $X \wedge . \vee L$ are not so obviously applications of \vee and \wedge over subsets.

3. COMMUTATOR

The monadic operator $\tilde{\cdot}$ commutes the sense of the arguments of a dyadic function F , that is, $X F \tilde{\cdot} Y \leftrightarrow Y F X$. For example, the secant slope $(- / F X + S, 0) \div S$ may be written as $S \div \tilde{\cdot} - / F X + S, 0$, and $(+ / V * 2) * .5$ may be written as $.5 * \tilde{\cdot} + / V * 2$. Moreover, the familiar transposition identity $A + . * B \leftrightarrow \circ (\circ B) + . * \circ A$ may be written (using the dual operator of Section 8) without arguments (i.e. as a relation between certain derived functions) as: $+ . * \leftrightarrow + . * \tilde{\cdot} \circ$, or equivalently, $+ . * \tilde{\cdot} \leftrightarrow + . * \tilde{\cdot} \circ$.

4. SCALAR OPERATORS AND THE IDENTITY FUNCTION

For each of the scalar dyadic primitive functions, we define an associated dyadic operator denoted by the symbol for the function overstruck with the overbar. The definition will be illustrated by the definition for $+$:

$$\text{(monadic)} \quad F \bar{+} G \ Y \leftrightarrow (F \ Y) + (G \ Y)$$

$$\text{(dyadic)} \quad X \ F \bar{+} G \ Y \leftrightarrow (X \ F \ Y) + (X \ G \ Y)$$

The sum over the values produced by the function F applied to its entire domain will be denoted by $F \bar{+} \circ$. A similar definition will apply to each of the scalar operators. For example, if P is a proposition, then $P \bar{+} \circ$ is 1 if and only if there exists an element of its domain for which P is true, and the number of elements in the set defined by P is given by $P \bar{+} \circ$.

If F is divalent, the expression $F \bar{+} \circ$ applies to the monadic function denoted by F ; the dyadic form must be specified explicitly in the manner presented in the following section.

One of the arguments of a scalar operator may be numeric, in which case the following definitions (shown for the specific scalar function $\bar{2}$ and the specific numeric quantity 2) apply:

$$2 \bar{2} F \ Y \leftrightarrow 2 * F \ Y$$

$$F \bar{2} \ Y \leftrightarrow (F \ Y) * 2$$

$$X \ 2\bar{*}F \ Y \leftrightarrow 2 * X \ F \ Y$$

$$X \ F\bar{*}2 \ Y \leftrightarrow (X \ F \ Y) * 2$$

Certain of the scalar operators are very useful in the treatment of sets. If we consider the proposition (i.e., a function whose result is 0 or 1) P which defines a set (i.e., PX is 1 if X belongs to the set and 0 if it does not), then $P\bar{\wedge}Q$ defines the intersection of the sets defined by P and by Q . Similarly, $P\bar{\vee}Q$ gives the union, $P\bar{>}Q$ the ordinary difference, and $P\bar{z}Q$ the symmetric difference.

Use of the form $P\bar{\vee}\circ$ will be illustrated by a proof of the fact that the number of elements in the union of sets P and Q (that is, $P\bar{\vee}Q\bar{\vee}\circ$) is equal to the sum of the numbers in P and Q , less the number in the intersection $P\bar{\wedge}Q$. For boolean arguments A and B , the expression $A\bar{\vee}B$ is equivalent to $(A+B) - A\bar{\wedge}B$. Consequently, $P\bar{\vee}Q \leftrightarrow (P\bar{\vee}Q)\bar{\vee}\circ - (P\bar{\wedge}Q)$, and:

$$\begin{aligned} &P\bar{\vee}Q\bar{\vee}\circ \\ &(P\bar{\vee}Q)\bar{\vee}\circ - (P\bar{\wedge}Q)\bar{\vee}\circ \\ &(P\bar{\vee}Q\bar{\vee}\circ) - (P\bar{\wedge}Q\bar{\vee}\circ) \\ &(P\bar{\vee}\circ) + (Q\bar{\vee}\circ) - (P\bar{\wedge}Q\bar{\vee}\circ) \end{aligned}$$

Further examples of the use of scalar operators will be found in the treatment of the derivative operator in Section 11.

5. DOMAIN AND VALENCE OPERATORS

If F is a function, then $F\bar{\Delta}\circ$ is a proposition of the same valence, and with an unrestricted domain, which defines the domain of F in the following sense:

(monadic) $F\bar{\Delta}\circ X$ is 1 if X is in the domain of F .

(dyadic) $X \ F\bar{\Delta}\circ \ Y$ is 1 if the pair X, Y is in the domain of F .

If F is a function and P is a proposition of the same valence, then $F\bar{\Delta}P$ is a function equivalent to F but further restricted to the domain defined by P . Formally:

$$F\bar{\Delta}P\bar{\Delta}\circ \leftrightarrow F\bar{\Delta}\circ \bar{\wedge}P$$

The valence of a function F may be delimited by one of the expressions $F\bar{\Delta}1$ or $F\bar{\Delta}2$, the value of the right argument determining the valence of the resulting derived function. Thus, as noted in Section 4, the sum over the dyadic domain of the bivalent function F would be denoted by $F\bar{\Delta}2\bar{\vee}\circ$, whereas the sum over the monadic domain could be denoted by either $F\bar{\Delta}1\bar{\vee}\circ$ or $F\bar{\vee}\circ$.

6. UNIFORM FUNCTIONS

A number of familiar monadic functions can be characterized by the rank of the argument or arguments for which they are defined, and by the ranks of the results they produce. For example:

	<u>Expression</u>	<u>Rank</u>
Monadic:	\int^{-3}	0 0
	$+/1\ 2\ 3\ 4\ 5$	0 1
	$\sqrt{5}$	1 0
	$\phi\ 1\ 2\ 3\ 4\ 5$	1 1
	Determinant	0 2
	Matrix inverse (\boxtimes)	2 2
Dyadic:	$3+4$	0 0 0
	$2\ 2\ 2\ \tau\ 6$	1 1 0
	$3\ \phi\ 1\ 2\ 3\ 4\ 5$	1 0 1
	$2\ 1\ 4\ \boxtimes\ M$	1 1 2

In general, functions are defined to have an explicit rank, although degenerate cases of lower rank are sometimes permitted, as in reduction or reversal of a scalar, or as in $\boxtimes\ 1\ 2\ 3$. We will therefore speak of the rank of a function as a vector whose first element specifies the result rank and whose remaining element or elements specify the argument rank or ranks. There remain some open questions concerning the specification of the ranks of certain functions (such as ravel) which appear to be of unrestricted rank.

Arguments and results of fixed ranks may still vary in shape, and we now define a uniform function as one whose result shape depends only upon the argument shape. For example, reversal, reduction, and scan are uniform functions, but \sqrt{N} is not. The importance of uniform functions is that their definition can be extended to arguments of higher rank in a systematic way, the argument being treated as an array of "subarrays" (along final axes which define the units of the function of appropriate rank), and the result is treated as the same array of subarrays of individual results. Non-uniform functions can be extended in a similar manner by employing the scalar representation function discussed in Section 10.

A uniform monadic function of rank RR, AR (result rank and argument rank) applies to an array X of rank AR or greater, to produce an overall result of rank $((-AR)+\rho X), RR$. The result can be perceived as follows: F is applied to each of the rank AR "units" of an "array of rank $(-AR)+\rho X$ " whose elements are these units, the RR coordinates of the individual results being placed last to produce the overall result of rank $((-AR)+\rho X), RR$.

7. NUCLEAR AXIS OPERATORS

The nuax operator (denoted by $\ddot{\circ}$) applies to a function left argument and a variable right argument to specify the axes which define the nuclei to which the function is to apply. For example, if ρA is 3 4 3 5 5 and F is the determinant function (of rank 0 2), then $F A$ yields a result of shape 3 4 3, and $F \ddot{\circ} 0 2 A$ yields a result of shape 4 5 5. The coax operator $\ddot{\circ}$ is also provided; its argument specifies the axes complementary to the nuclear axes. For example, $F \ddot{\circ} 1 3 4 A$ is equivalent to $F \ddot{\circ} 0 2 A$, whereas $F \ddot{\circ} 4 3 1 A$ has shape 5 5 4.

It must be emphasized that if a function F has a result rank greater than zero, the corresponding axes occur as the final axes in the overall result, and that the axial operators do not specify the allocation of these "result" axes in the overall result. In this they differ from the use of brackets to specify axes, as in $+[3]A$ and $\phi[3]A$, since the 3 in these cases specifies not only that the units be along axis 3, but also that the vector results are to lie along the same axis. For example, if ρA is 2 3 4 5, then $\rho+[2]A$ is also 2 3 4 5, but the result of plus scan over axis 2 as specified by the nuax operator has shape 2 3 5 4.

The fact that the result axes occur last in the overall result is often convenient. For example, $+/(\ddot{\circ}I)A$ sums over the axes I of A .

For a dyadic left argument F , the right argument of the axis operator $\ddot{\circ}$ has the form $I, _, J$, where I and J are vectors (of any length, including zero) specifying the units of the left and right arguments respectively (and where $_$ represents infinity). For example, $\boxplus \ddot{\circ} 1 _ 1 3$ applies \boxplus to vector left arguments along axis 1 and to matrix right arguments along axes 1 and 3. Similarly, $K \phi \ddot{\circ} _ 3 A$ rotates the vectors along axis 3 of A by amounts specified by the scalars of K .

If the right argument of $\ddot{\circ}$ requires the general form $I, _, J$, then any right argument K which does not include an infinity applies to specify axes of both the left and the right argument of the resulting derived function and is therefore equivalent to $K, _, K$. The argument $_, K$ is, of course, equivalent to $'', _, K$ and specifies scalar units in the left argument, and $K, _$ specifies scalar units on the right.

8. COMPOSITION AND DUALITY

The dual operator, denoted by $\ddot{\vee}$, is a slight extension of the notion of dual functions implicit in deMorgan's law ($\vee \ddot{\vee} \sim \leftrightarrow \wedge$ and $\neq \ddot{\vee} \sim \leftrightarrow =$), the extension being to include a monadic left argument, as in $[\ddot{\vee} X \leftrightarrow] X$. Composition, denoted

by \circ , is the familiar notion of the composition of two monadic functions ($F \circ G \ X \leftrightarrow F \ G \ X$) extended to the two cases of one dyadic argument, as well as to the case of character arguments which define a function in terms of the expression represented by one of the arguments.

Duality and simple composition. Composition and the dual operator applied to a divalent left argument and a monadic (or divalent) right argument yield parallel definitions of divalent derived functions as follows:

$$\begin{aligned} \text{Composition:} \quad & F \circ G \ Y \leftrightarrow F \ G \ Y \\ & X \ F \circ G \ Y \leftrightarrow (G \ X) \ F \ (G \ Y) \end{aligned}$$

$$\begin{aligned} \text{Dual:} \quad & F \check{\vee} G \ Y \leftrightarrow (G \check{\ast}^{-1}) \ F \ (G \ Y) \\ & X \ F \check{\vee} G \ Y \leftrightarrow (G \check{\ast}^{-1}) \ (G \ X) \ F \ (G \ Y) \end{aligned}$$

It should be noted that the extension of the dual to include the monadic definition makes the identities $[\check{\vee} \leftrightarrow]$ and $[\check{\vee} \leftrightarrow]$ hold for both the monadic case (floor and ceiling) and for the dyadic case (minimum and maximum). Moreover, for the dyadic case the exponential function yields the identities $\times \check{\vee} \ast \leftrightarrow +$ and $+ \check{\vee} \ast \leftrightarrow \times$, the latter of which provides the basis for the use of natural logarithms in multiplication, just as the identity $+ \check{\vee} (10 \ast) \leftrightarrow \times$ forms the basis for the use of base ten logarithms.

In the dyadic case of composition above, the first function F is dyadic and the second function G is monadic. This is the case that prevails when the functions have these valencies or are divalent. If either F is specifically monadic or G is specifically dyadic, we obtain another important case defined as follows:

$$X \ F \circ G \ Y \leftrightarrow F \ X \ G \ Y$$

For divalent functions this form can be obtained by fixing the valence of one or both of the functions with the valence operator defined in Section 5. Frequently the valence becomes fixed by composition with a variable (as defined in the following subsection), as in the expression $2 \ast$.

The corresponding case for the dual operator ($F \check{\vee} DG$) would lead to the definition $F \check{\ast}^{-1} \ F \ X \ DG \ Y$. It would therefore be vacuous (being equivalent to DG) and is consequently excluded.

The definition of the composition $F \circ G$ given above must be understood as referring to a single argument of rank appropriate to the function G . For example, if F is \mathbb{Q} and G is \mathbb{Q} , and ρA is $\begin{smallmatrix} 3 & 4 & 4 \end{smallmatrix}$, then the units of G are $\begin{smallmatrix} 4 & 4 \\ 4 & 4 \end{smallmatrix}$ matrices, each of which is inverted and transposed to produce an overall result of shape $\begin{smallmatrix} 3 & 4 & 4 \end{smallmatrix}$. Thus (assuming that \mathbb{Q} is of rank $\begin{smallmatrix} 2 & 2 \end{smallmatrix}$, and extends to higher rank arrays in the normal manner) $\mathbb{Q} \circ \mathbb{Q} A$ differs from the expression $\mathbb{Q} \ \mathbb{Q} \ A$,

since the latter would invert each of the three 4 by 4 matrices of A to produce a result of shape 3 4 4 which would then be transposed to produce a result of shape 4 4 3.

Similar remarks apply to the dual operator, and in general to all operators. In other words, a derived function applies as an indivisible whole to each of the nuclei of its argument. The dual operator is defined formally in terms of composition as follows:
 $F\check{V}G \leftrightarrow G\check{*}1''(F\check{*}G)$.

Composition with one variable argument. The composition operator applied to a function F and a variable V produces a result of valence one less than the valence of F . Thus if F is dyadic, the result is monadic; if F is monadic, the result is a constant. For example:

- *".5 $X \leftrightarrow X * .5$ (*".5 is monadic)
- *".5 $\leftrightarrow * .5$ (A constant value)
- 2"* $X \leftrightarrow 2 * X$ (2"* is monadic)
- 2"*".5 $\leftrightarrow 2 * .5$ (A constant value)

This form of composition is particularly useful in providing the monadic cases (both left and right) of dyadic functions, as illustrated above for the square root (*".5) and the powers of two (2"*).

The valence of the derived function is determined by context, and this in turn determines the valence of the argument function as one greater. For example, in the expression *".5 (4 9), the function *".5 is monadic, * is dyadic, and the result is the constant 2 3; in the expression *".5, the function *".5 is niladic (a constant), * is monadic, and the result is the exponential *.5.

Composition is useful in a wide variety of ways. For example, with one variable argument it provides the monadic cases (both left and right) of dyadic functions, as illustrated above for the powers of two (2"*) and the square root (*".5). An example for the case of two functions is treated in Section 11.

The dual operator is also widely useful. For example, $\langle \backslash$ applied to a logical vector yields the same vector with all the 1's following the first replaced by zeros, whereas $\langle \backslash \check{\vee} \phi$ yields the vector with all the 1's preceding the last replaced by zeros. Similarly, if F is any function which applies to columns of a matrix argument, then $F\check{V} \phi$ applies similarly to rows. A further important use of the dual is discussed in Section 10.

Composition with two variables (characters). This use of composition is modeled on the direct or $\alpha \omega$ definition form defined in Elementary Analysis [2]. It incorporates the same abilities to define functions with any number of global (non-explicit) arguments, to prevent side-effects (by automatically localizing all variables specified within the defining expression), and to make conditional and recursive definitions. The definition is generalized to allow the specification of the names of explicit arguments in a second argument (with a jot (\circ) for second argument signifying the use of α and ω), and to allow the specification of divalent functions. Thus, $F \overleftarrow{B} X' \overleftarrow{B}$ and $F \overleftarrow{\omega} X' \circ$ are equivalent, and if $X \leftarrow 5$, then $F \overleftarrow{20}$ is 4, and $10 \overleftarrow{B} X' \overleftarrow{X B} \overleftarrow{20}$ is 2.

Recursive definition requires the use of the function name in its definition; this name is identified by placing it first in the right argument, separated from the other name or names by a colon. For example, if $C \overleftarrow{X \times F} X-1: X=0:1$, then $C \overleftarrow{F: X}$ defines the factorial function on non-negative integers. The name F is local and is not associated in any way with the derived function. For the $\alpha \omega$ case $C \overleftarrow{\circ}$, the name of the function is denoted by $\underline{\alpha}$, as in $C \overleftarrow{\omega \times \underline{\alpha}} \omega-1: \omega=0:1$.

The symbols α and ω may be used like any other names except that they must not remain as global names in the resulting function. For example, one may use $C \overleftarrow{\alpha \omega}$ in lieu of $C \overleftarrow{\circ}$, and $\overleftarrow{X \alpha}$ or $\overleftarrow{\omega \alpha}$ for the equivalent of $\overleftarrow{\alpha + \omega} \circ$.

This form of composition is very convenient for defining and applying functions in computations and for use in theoretical work. For example:

$$\begin{aligned}
 & POL \overleftarrow{(\omega \circ . * 1 \rho \alpha) + . \times \alpha} \circ \quad [8.1] \\
 & \quad 1 \ 3 \ 3 \ 1 \ POL \ 2 \ 4 \\
 27 \ 125 & \\
 & \quad 1 \ 3 \ 3 \ 1 \ \overleftarrow{(\omega \circ . * 1 \rho \alpha) + . \times \alpha} \circ \ 2 \ 4 \\
 27 \ 125 & \\
 & SPOL \overleftarrow{C} POL \ \omega \circ \\
 & \quad C \overleftarrow{1 \ 4 \ 6 \ 4 \ 1} \\
 & \quad SPOL \ 2 \ 4 \\
 16 \ 625 & \\
 & \overleftarrow{\alpha + \omega} \circ / 1 \ 2 \ 1 \ 2 \text{ (Continued fraction)} \\
 1.375 & \\
 & \overleftarrow{\alpha + \omega} \circ \setminus 1 \ 2 \ 1 \ 2 \text{ (Convergents)} \\
 1 \ 1.5 \ 1.333 \ 1.375 & \\
 & F \overleftrightarrow{\omega} \overleftarrow{F \alpha} \circ \\
 & \quad \overleftarrow{(\alpha \ POL \ X) \times \omega \ POL \ X} \circ \leftrightarrow \\
 & \quad \overleftarrow{(+ f(-1 \rho \alpha) \phi \alpha \circ . \times \omega, 1 + 0 \times \alpha) \ POL \ X} \circ
 \end{aligned}$$

Divalent functions are defined by catenating the arguments for the monadic and dyadic cases, inserting a pair

of colons to separate the two parts. More precisely, if $CA''NA$ and $CB''NB$ define (not necessarily respectively) monadic and dyadic functions, then $(CA, '::', CB)''NA, '::', NB$ defines the corresponding divalent function. Analogously, $(CA, '::', CB)''\circ$ defines the divalent function corresponding to $CA''\circ$ and $CB''\circ$. For example:

$$F\bar{+}G\leftrightarrow '(F\omega)+G\omega::(\alpha F\omega)+\alpha G\omega''\circ$$

The identity element (or elements) of a function may be specified by appending to the name (in the right argument of $\bar{}$) the symbol $\bar{}$ followed by the appropriate value. For example, $'X+\bar{\div}Y''\bar{}'X\ Y\ \bar{}'$, and $'X+Y''\bar{}'X\ \bar{}\ 0\ Y\ \bar{}\ 0'$.

9. REDUCTION AND SCAN

The reduction operator is defined as follows: F/A is an application of the dyadic function F to the set of arguments obtained by indexing A on its last axis. For example, if ρA is $4\ 4\ 5$, then $+.x/A$ applies the matrix product over the five four-by-four matrices of A to produce a four-by-four result. Similarly, if ρA is $4\ 4\ 4\ 5$, then $+.x/A$ yields a result of shape $7\rho 4$, and $+.x\circ 1\ 2/A$ yields a result of shape $3\rho 4$, and $\circ.(+.x\circ 1\ 2)/A$ yields a result of shape $7\rho 4$.

The reduction $F\bar{+}A$ is defined analogously (the indexing applying to the leading axis), as are the scans $\bar{}$ and $\bar{}$.

The axis operators $\bar{}$ and $\bar{}$ apply to the reduction operator ($/$ or $\bar{+}$) to produce an operator which applies its argument function to the arguments obtained by indexing the specified axis. For example, $+.x(/ \bar{} 0)A\leftrightarrow+.x\bar{+}A$.

It should be noted that $F(\bar{}\circ K)$ and $F\bar{}[K]$ are not, in general, equivalent, since the result vectors of the former lie along the last axis of the result rather than along axis K . Similarly, $F(/ \bar{}\circ K)$ and $F/[K]$ are not equivalent in general, although they are for a scalar function F .

10. SCALAR REPRESENTATION

The enclose function (denoted by \langle) produces a scalar representation of its argument in the sense that the result is of rank zero, and that there exists an inverse function (called disclose, and denoted by \rangle) such that $A\leftrightarrow\rangle\langle A$ for all A . Any result producible by an expression which does not employ the enclose function is called a simple array, or is said to be simple.

Selection and reshaping functions apply without change to non-simple arrays. However, non-simple arrays are outside the domain of all other functions except for

enclose, disclose, and equality (together with those functions such as \neq and \in which are defined in terms of equality).

The equality function is extended to non-simple scalar arguments as follows:

1. $\langle A \rangle \neq A$ for all A
2. If A equals B (in rank, shape, and all elements), then $\langle A \rangle = \langle B \rangle$ yields 1

The enclose function applies to all axes of its argument (i.e., to the entire argument) to produce a single scalar result, but it can also be used in conjunction with the axial operators to apply to units determined by the specified axes. For example, if $A \leftarrow 2 \ 3 \ 4 \ 5 \ \rho_1!5$, then $M \leftarrow \langle \langle 1 \ 3 \ 4 \rangle \rangle$ is a 2 by 4 matrix such that for scalar indices I and J , the element $M[I;J]$ is a non-simple scalar whose disclose $\langle M[I;J] \rangle$ is a 3 by 5 simple matrix.

The disclose function is scalar in the sense that it applies to each element of its argument, the new axes disclosed becoming the final axes of the result. For example, using the matrix M of the preceding paragraph, $\langle M \rangle$ is a simple array of shape 2 4 3 5, and $\wedge /, \langle M \rangle = 0 \ 2 \ 1 \ 3 \ \& A$.

The axial operators can also be applied to the disclose function to determine the axes to be disclosed. For example, $\langle \langle 1 \ M \rangle \rangle$ produces a result of shape 2 4 5 whose elements are enclosed 3-element vectors.

The disclose function applied to a simple array A produces a result identical to A . Thus $\langle A \rangle = \langle \langle A \rangle \rangle$ is a test for whether A is simple.

The expression $F \vec{\vee}$ produces a derived function which applies the function F to its argument in an "item-wise" fashion, by disclosing each element of the argument, applying F , and enclosing the result to produce the corresponding element of the overall result. Thus, if $R \leftarrow F \vec{\vee} A$ and Q is a particular scalar element of A , then the corresponding element of R is $\langle F \rangle Q$. In particular, ρR equals ρA .

11. DERIVATIVE OPERATOR

Derivatives apply only to monadic functions, but to functions of any rank R , the derivative $F \Delta$ yielding a monadic function of rank $(+/R), R$. Thus:

	<u>Rank of F</u>	<u>Rank of FΔ</u>	
(Scalar function)	0 0	0 0	(Scalar function)
(Vector function of a scalar)	1 0	1 1	
(Scalar function of a vector)	0 1	1 1	(gradient)
(Vector function V)	1 1	2 1	(Matrix of partials)
(VΔ)	2 1	3 1	(VΔ Δ)

The shape of $F\Delta$ is determined as follows: if U is a unit of F , then $\rho F\Delta U$ is $(\rho F U), \rho U$.

The derivative operator together with composition and scalar operators can be used to express the familiar differentiation rules as follows:

$$\begin{aligned}
 F\bar{\nabla}G\Delta &\leftrightarrow F\Delta\bar{\nabla}(G\Delta) \\
 F\bar{\times}G\Delta &\leftrightarrow F\Delta\bar{\times}G\bar{\nabla}(G\Delta\bar{\times}F) \\
 F''G\Delta &\leftrightarrow F\Delta''G\bar{\times}(G\Delta) \\
 \div''G\Delta &\leftrightarrow \div''(G\Delta)\bar{\nabla}(G\bar{\times}2) \\
 *''N\Delta &\leftrightarrow *''(N-1)\bar{\times}N \\
 * \Delta &\leftrightarrow *
 \end{aligned}$$

12. DIFFERENCE OPERATOR

The derivative operator Δ discussed in the preceding section actually yields a divalent result, the dyadic form being the difference function defined as follows:

$$S F\Delta X \leftrightarrow ((F X+S)-F X)\div S$$

This function is also completed at zero so that the expression $0 F\Delta X$ equals the derivative $F\Delta X$.

13. DYADIC REDUCTION AND SCAN

If F is a function, then $F/$ and $F\backslash$ produce divalent derived functions whose dyadic forms are defined here. If $R \leftarrow K F/X$ and $(K \geq 0) \wedge K \leq \rho X$, then ρR is $1 + (\rho X) - K$, and $R[I]$ is $F/K \uparrow I \uparrow X$. Moreover, $(-K)F/X$ is the dual of $K F/X$ with respect to reversal, in the following sense:

$$(-K)F/X \leftrightarrow \phi K F / \phi X$$

Thus $\overset{-}{2} -/X$ and $\overset{-}{2} -/X$ yield backward and forward differences of X , respectively. More generally, $\overset{-}{2} F/X$ and $\overset{-}{2} F/X$ yield "pairwise" applications of F . For example, $\overset{-}{2} \div/T$ yields the ratios of terms in a series, and $\wedge/2 = /V$ determines whether all elements of V are equal.

The new coordinate of the result produced by the dyadic form precedes those of the monadic form. Thus if S is the shape of F/X , then the shape of $K F / X$ has the form N, S . Moreover, $(\rho X) F/X$ is equivalent to F/X except that the shape of the result has a leading element of 1.

The dyadic form of $F \setminus$ is defined similarly, differing only in that it includes the reductions over all leading prefixes so as to yield a result of the same length as the argument X . More precisely, if $R \leftarrow K F \setminus X$, then ρR is ρX and:

$$R[I] \leftrightarrow F / ((|K)|I+1) + (0[I - (|K) - 1]) \setminus X$$

For example, $\overset{-}{2} - \setminus X$ yields the forward differences of X but includes the leading element of X so that $+ \setminus$ is a true inverse, that is,

$$+ \setminus \overset{-}{2} - \setminus X \leftrightarrow X$$

$$\overset{-}{2} - \setminus + \setminus X \leftrightarrow X$$

14. THE VARIANT OPERATOR

Certain functions have variants, in the sense that there exist other closely related functions. For example, the sine of an argument in radians and the sine of an argument in degrees are variants. Moreover, in current APL each of the functions dependent on index origin has two variants (chosen by $\square IO$) and each relation has many variants (specified by $\square CT$).

We now introduce a dyadic variant operator, denoted by $\bar{\cdot}$ and called mark, whose right argument specifies the particular variant. Functions are treated in seven classes as follows:

- 1) Dyadic circular functions with left arguments 1, 2, 3, $\bar{1}$ $\bar{2}$, $\bar{3}$.

$$o \bar{\cdot} D \omega \leftrightarrow o \omega \div D \div 0.5$$

Thus the right argument D gives the number of divisions in a right angle and, in particular, $o \bar{\cdot} 90$ is the circular function for arguments in degrees, and $o \bar{\cdot} 100$ is for arguments in grads.

2) Relations

$F\bar{K}$ uses comparison tolerance K . For example,
 $X\leq\bar{1}E\bar{6} Y$.

3) Indexing (monadic and dyadic \uparrow and subscripting)

$\uparrow\bar{K}$ and $[\bar{K}$ (as in $M[\bar{1};I;J]$) specify origin K .

4) Random (monadic and dyadic ?)

The right argument is a function which specifies the distribution or a character which selects one of several standard distribution as follows:

Distribution

R	Rectangular
G	Gauss
P	Poisson

This variant also takes an integer right argument to determine the origin of the population, as in $?\bar{0}$, or $?\bar{1}$, or $?\bar{'G'}\bar{1}$.

5) Residue arithmetic (dyadic $+, -, \times, *$, and monadic $-$)

$X F\bar{K} Y \leftrightarrow K | X F Y$

In particular, since $0 | X \leftrightarrow X$, we have $F\bar{0} \leftrightarrow F$

6) The functions take and expand insert at certain positions a "fill element" (blank for characters, and zero for numeric); the variant operator allows the specification of this element, as in $\uparrow\bar{1}$ and $L\backslash\bar{'*'}\bar{1}$, and $L\backslash\bar{0}$.

7) The conformability requirements for catenation can be relaxed by "padding" either argument along the axes complementary to the catenation axis, using overtakes and some specified fill element. This is provided by the variant operator, the right argument being the fill element, or a jot indicating the default fill elements blank or zero. For example, $M,\bar{0} N$ and $M,\bar{'*'}\bar{0} N$.

For any function F dominated by the variant operator, the function F alone is treated as a default case, the default value being given by a system variable. For the relations, this system variable is $\square CT$, and therefore $\leq \leftrightarrow \leq \bar{\square CT}$.

The default for the functions $\uparrow, ?$, and $[\bar{K}$ is determined by $\square IO$, and for the remaining cases we introduce new system variables as follows:

Circular Functions:	$\square CF$	0.5
Residue Arithmetic:	$\square RA$	0
Type of Distribution:	$\square DT$	'R'
Fill Element:	$\square FE$	0

The rightmost column shows the values in a clear workspace.

15. BOOLEAN OPERATOR

The boolean operator \ast is an example of a useful monadic function having a numeric argument. Applied to an integer $I \in \{1, 16\}$ it produces the " I th boolean function" according to the following definition:

$$I \leftrightarrow 2 \uparrow, 0 \ 1 \circ. (I \ast) 0 \ 1$$

Thus $1 \ast \leftrightarrow \wedge$ and $7 \ast \leftrightarrow \vee$, and in general, $\sim (I \ast) \leftrightarrow (15 - I) \ast$. The use of a vector I produces a vector function.

16. SET FUNCTIONS

We define seven set functions as follows:

<u>Nub</u>	uW	$((\uparrow \rho W) = W \uparrow W) / W \uparrow, W$
<u>Ordered Nub</u>	nW	$W [\uparrow W \leftarrow uW]$
<u>Distribution</u>	θW	$(uW) \circ. = W$
<u>Ordered Distribution</u>	nW	$(nW) \circ. = W$
<u>Union</u>	$V \cup W$	$(, V), , W$
<u>Intersection</u>	$V \cap W$	$((, V) \epsilon, W) / , V$
<u>Difference</u>	$V \sim W$	$(\sim(, V) \epsilon, W) / , V$
<u>Inclusion</u>	$V \subseteq W$ $V \supseteq W$	$\wedge / (, V) \epsilon W$ $W \subseteq V$
<u>Strict Inclusion</u>	$V \subset W$ $V \supset W$	$(V \subseteq W) \wedge \sim V \supseteq W$ $W \subset V$

Practical and theoretical application of the distribution functions will now be illustrated. If V is a vector with repeated elements and F is a scalar function, then every distinct element of $F V$ is contained in $F \cup V$, and these elements can be "distributed" to the positions their arguments occupied in V by the expression $(F \cup V) \uparrow. \times \theta V$. Thus:

$$F V \leftrightarrow (F \cup V) + . \times \cup V \quad [16.1]$$

If the evaluation of F is time-consuming, and if $\rho \cup V$ is considerably less than ρV , this identity can provide an efficient algorithm for the evaluation of $F V$. Similar remarks apply to the ordered nub and the ordered distribution matrix.

The product of the sum of two vectors A and B (that is, $\times/A+B$) can be "expanded" to an equivalent sum of $2 \times \rho B$ terms, in which a typical term is a product of ρB factors, one chosen from each position of either A or B . Formally, each term is of the form $(A \times . * \sim L) \times (B \times . * L)$, where L is a logical vector of shape ρB . All terms are therefore obtained by replacing L by the matrix $T \leftarrow (K \rho 2) \tau_{12} * K \leftarrow \rho B$ of all such logical vectors. We therefore have the identity:

$$\times/A+B \leftrightarrow (A \times . * \sim T) + . \times (B \times . * T \leftarrow (K \rho 2) \tau_{12} * K \leftarrow \rho B)$$

For a scalar X , the expression $\times/X-R$ yields the value of a polynomial with roots R evaluated at X . By substituting X for A and $-R$ for B in the foregoing identity, and using the fact that (for a scalar X) the expressions $X \times . * L$ and $X * + / L$ are identical, we can derive a sequence of equivalent expressions which finally yield a relation between the coefficients and the roots of a polynomial:

$$\begin{aligned} & \times/X-R \\ & (X \times . * \sim T) + . \times (B \times . * T \leftarrow (K \rho 2) \tau_{12} * K \leftarrow \rho B \leftarrow -R) \\ & (X * S \leftarrow + / \sim T) + . \times (B \times . * T) \\ & ((X * n S) + . \times n S) + . \times (B \times . * T) \quad 16.1 \\ & (X * n S) + . \times ((n S) + . \times (B \times . * T)) \quad + . \times \text{ is assoc} \\ & (X * 11 + \rho B) + . \times ((n S) + . \times (B \times . * T)) \quad n S \leftarrow 11 + \rho B \\ & ((n S) + . \times (B \times . * T)) \text{ POL } X \quad \text{POL in 8.1} \end{aligned}$$

The left argument of POL above is therefore the coefficient vector of the polynomial whose roots are R . Consequently, a function CFR , which yields the coefficients when applied to the vector of roots, may be defined as follows:

$$CFR \leftarrow (n + / \sim T) + . \times (-\omega) \times . * T \leftarrow (K \rho 2) \tau_{12} * K \leftarrow \rho \omega'$$

The definition of CFR is, of course, a definition of Newton's symmetric functions.

As a final example of the use of the (ordered) distribution function, we state the shape of the transpose $I \cup A$ as follows:

$$\rho I \cup A \leftarrow (n I) \downarrow . (\downarrow * \circ) \sim \rho A$$

17. INDEXING

The range of indexing will now be extended to include negative numbers in a manner which makes it possible to refer to an index position by two different numbers, one relative to its position as counted forward, and one (non-positive) relative to its position as counted backward: an axis of length N may be indexed by any of the elements of $(12 \times N) - N$, the position indexed by I being $I + N \times I < 0$. Thus:

```

      N+4
      □IO+1
      □←I←(12×N)-N
-3 -2 -1 0 1 2 3 4
      I+N×I<0
1 2 3 4 1 2 3 4
      □IO+0
      □←I←(12×N)-N
-4 -3 -2 -1 0 1 2 3
      I+N×I<0
0 1 2 3 0 1 2 3
    
```

For example, in 0-origin, $X[-1] \leftrightarrow X[-1+\rho X]$, and selects the last element of X . Similarly, in 1-origin, $X[0]$ selects the last element.

We also introduce a form of indexing called from, denoted by \square , and of rank $0, 1, RR$, where R is the rank of the right argument. The basic definition is:

$$I \square A \leftrightarrow (, A) [\square(\rho A) \perp \square I]$$

The function \square distributes over any scalar function; thus, $I \square A + B \leftrightarrow (I \square A) + (I \square B)$.

The right argument of the axis operator applied to \square is of the form $I, _ , J$. For example:

```

      M+3 4ρ112
      M
0  1  2  3
4  5  6  7
8  9 10 11
      2 2 □ M
10
      2 □ ¨_ 1 M
2 6 10
      2 0 1 □ ¨_ 1 M
2 4 9
      (3 2ρ3|16) □ M
1 8 6
    
```

The use of the indexing function will be illustrated in a proof of the useful identity $I \square J \square A \leftrightarrow I [J] \square A$. We first

state the definition of the transpose as follows: if K is any vector index of $J \otimes A$, then

$$K \otimes J \otimes A \leftrightarrow K[J] \otimes A$$

Then:

$$\begin{aligned} K \otimes I \otimes J \otimes A \\ K[I] \otimes J \otimes A \\ (K[I])[J] \otimes A \\ K[I[J]] \otimes A \\ K \otimes I[J] \otimes A \end{aligned}$$

REFERENCES

1. APL Language, Form Number GC26-3847, IBM Corporation.
2. Iverson, K. E., Elementary Analysis, APL Press, 1976.