Graph Notation for Arrays

Hans G. Ehrbar
Economics Department, University of Utah
1645 Campus Center Drive
Salt Lake City, UT 84112-9300
U.S.A.
+1 801 581 7797
ehrbar@econ.utah.edu

July 24, 2000

Abstract

A graph-theoretical notation for array concatenation represents arrays as bubbles with arms sticking out, each arm with a specified number of "fingers." Bubbles with one arm are vectors, with two arms matrices, etc. Arrays can only hold hands, i.e., "contract" along a given pair of arms, if the arms have the same number of fingers. There are three array concatenations: outer product, contraction, and direct sum. Special arrays are the unit vectors and the diagonal array, which is the branching point of several arms. Outer products and contractions are independent of the order in which they are performed and distributive with respect to the direct sum. Examples are given where this notation clarifies mathematical proofs.

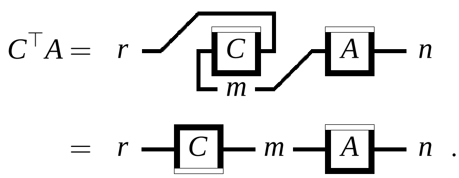
1 Introduction

Besides scalars, vectors, and matrices, also higher arrays are necessary in statistics; for instance, the "covariance matrix" of a random matrix is really an array of rank 4, etc. Usually, such higher arrays are avoided in the applied sciences because of the difficulties to write them on a two-dimensional sheet of paper. The following symbolic notation makes the structure of arrays explicit without writing them down element by element. It is hoped that this makes arrays easier to understand, and that this notation leads to simple high-level user interfaces for programming languages manipulating arrays.

2 Informal Survey of the Notation

Each array is symbolized by a rectangular tile with arms sticking out, similar to a molecule. Tiles with one arm are vectors, those with two arms matrices, those with more arms are arrays of higher rank (or "valence" as in [SS35], [Mor73], and [MS86, p. 12]), and those without arms are scalars. The arrays considered here are rectangular, not "ragged," therefore in addition to their rank we only need to know the dimension of each arm; it can be thought of as the number of fingers associated with this arm. Arrays can only hold hands (i.e., "contract" along two arms) if the hands have the same number of fingers.

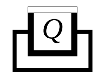
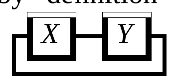
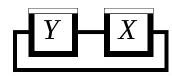
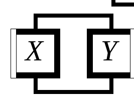
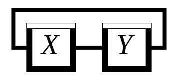
Sometimes it is convenient to write the dimension of each arm at the end of the arm, i.e., a $m \times n$ matrix A can be represented as $m -\!\boxed{A}\!- n$. Matrix products are represented by joining the obvious arms: if B is $n \times q$, then the matrix product AB is $m -\!\boxed{A}\!- n -\!\boxed{B}\!- q$ or, in short, $-\!\boxed{A}\!-\!\boxed{B}\!-$. The notation allows the reader to always tell which arm is which, even if the arms are not marked. If $m -\!\boxed{C}\!- r$ is $m \times r$, then the product $C^\top A$ is



(1)

In the second representation, the tile representing C is turned by 180 degrees. Since the white part of the frame of C is at the bottom, not on the top, one knows that the West arm of C, not its East arm, is concatenated with the West arm of A. The transpose of $m -\!\boxed{C}\!- r$ is $r -\!\boxed{C}\!- m$, i.e., it is not a different entity but the same entity in a different position. The order in which the elements are arranged on the page (or in computer memory) is not a part of the definition of the array itself. Likewise, there is no distinction between row vectors and column vectors.

Vectors are usually, but not necessarily, written in such a way that their arm points West (column vector convention). If $-\!\boxed{a}$ and $-\!\boxed{b}$ are vectors, their scalar product $a^\top b$ is the concatenation $\boxed{a}\!-\!\boxed{b}$ which has no free arms, i.e., it is a scalar, and their outer product $ab^\top$ is $-\!\boxed{a}\ \ \boxed{b}\!-$, which is a matrix. Juxtaposition of tiles represents the outer product, i.e., the array consisting of all the products of elements of the arrays represented by the tiles placed side by side.

The trace of a square matrix $-\!\boxed{Q}\!-$ is the concatenation , which is a scalar since no arms are sticking out. In general, concatenation of two arms of the same tile represents *contraction*, i.e., summation over equal values of the indices associated with these two arms. This notation makes it obvious that $\mathrm{tr}XY = \mathrm{tr}YX$, because by definition there is no difference between  and . Also  or  etc. represent the same array (here array of rank zero, i.e., scalar). Each of these tiles can be evaluated in essentially two different ways. One way is

1. Juxtapose the tiles for X and Y, i.e., form their outer product, which is an array of rank 4 with typical element $x_{mp}y_{qn}$.

2. Connect the East arm of X with the West arm of Y. This is a contraction, resulting in an array of rank 2, the matrix product XY, with typical element $\sum_p x_{mp}y_{pn}$.

3. Now connect the West arm of X with the East arm of Y. The result of this second contraction is a scalar, the trace $\mathrm{tr}XY = \sum_{p,m} x_{mp}y_{pm}$.

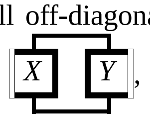
An alternative sequence of operations evaluating this same graph would be

1. Juxtapose the tiles for X and Y.

2. Connect the West arm of X with the East arm of Y to get the matrix product YX.

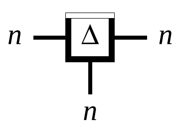
3. Now connect the East arm of X with the West arm of Y to get $\mathrm{tr}YX$.

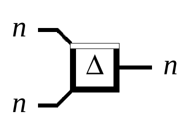
The result is the same, the notation does not specify which of these alternative evaluation paths is meant, and a computer receiving commands based on this notation can choose the most efficient evaluation path. Probably the most efficient evaluation path is given by (13) below: take the element-by-element product of X with the transpose of Y, and add all the elements of the resulting matrix.

If the user specifies $\mathrm{tr}(XY)$, the computer is locked into one evaluation path: it first has to compute the matrix product XY, even if X is a column vector and Y a row vector and it would be much more efficient to compute it as $\mathrm{tr}(YX)$, and then form the trace, i.e., throw away all off-diagonal elements. If the trace is specified as 

, the computer can choose the most efficient of a number of different evaluation paths transparently to the user. This advantage of the graphical notation is of course even more important if the graphs are more complex.

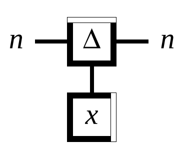
There is also the "diagonal" array, which in the case of rank 3 can be written



or


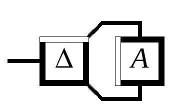
(2)

or similar configurations. It has 1's down the main diagonal and 0's elsewhere. It can be used to construct the diagonal matrix $\mathrm{diag}(x)$ of a vector (the square matrix with the vector in the diagonal and zeros elsewhere) as

$\mathrm{diag}(x) =$ 

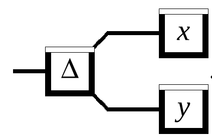
, (3)

the diagonal vector of a square matrix (i.e., the vector containing its diagonal elements) as



, (4)

and the "Hadamard product" (element-by-element product) of two vectors $x * y$ as

$x * y =$ 

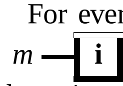
. (5)

The graphical representation of all these different matrix operations uses only a small number of atomic operations, which will be enumerated in Section 3, and each such graph can be evaluated in a number of different ways. In principle, each graph can be evaluated as follows: form the outer product of all arrays involved, and then contract along all those pairs of arms which are connected. For practical implementations it is more efficient to develop functions which connect two arrays along one or several of their arms without first forming outer products, and to add the arrays one by one, performing all contractions as early as possible.

3 Axiomatic Development of Array Operations

The following sketch shows how this axiom system might be built up. I apologize for presenting a half-finished theory, and thank the conference organizers and referees for allowing me to do this. Since I am an economist I do not plan to develop the material presented here any further. Others are invited to take over. If you are interested in working on this, I would be happy to hear from you; email me at `ehrbar@econ.utah.edu`

There are two kinds of special arrays: unit vectors and diagonal arrays.

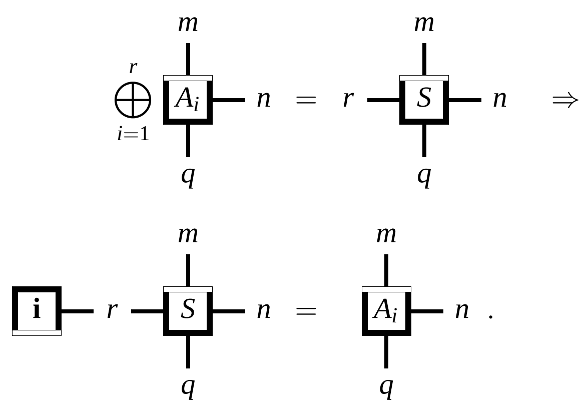
For every natural number $m \geq 1$, m *unit vectors* 

($i = 1, \ldots, m$) exist. Despite the fact that the unit vectors are denoted here by numbers, there is no intrinsic ordering among them; they might as well have the names "red, green, blue, … " (From (9) and other axioms below it will follow that each unit vector can be represented as a m-vector with 1 as one of the components and 0 elsewhere.)

For every rank ≥ 1 and dimension $n \geq 1$ there is a unique *diagonal array* denoted by Δ. Their main properties are (6) and (7). (This and the other axioms must be formulated in such a way that it will be possible to show that the diagonal arrays of rank 1 are the "vectors of ones" ι which have 1 in every component; diagonal arrays of rank 2 are the identity matrices; and for higher ranks, all arms of a diagonal array have the same dimension, and their $ijk\cdots$ element is 1 if $i = j = k = \cdots$ and 0 otherwise.) Perhaps it makes sense to define the diagonal array of rank 0 and dimension n to be the scalar n, and to declare all arrays which are everywhere 0-dimensional to be diagonal.

There are only three operations of arrays: their outer product, represented by writing them side by side, contraction, represented by the joining of arms, and the direct sum, which will be defined now:

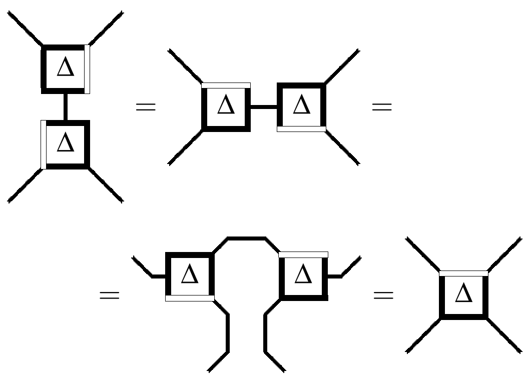
The direct sum is the operation by which a vector can be built up from scalars, a matrix from its row or column vectors, an array of rank 3 from its layers, etc. The direct sum of a set of r similar arrays (i.e., arrays which have the same number of arms, and corresponding arms have the same dimensions) is an array which has one additional arm, called the reference arm of the direct sum. If one "saturates" the reference arm with the ith unit vector, one gets the ith original array back, and this property defines the direct sum uniquely:



It is impossible to tell which is the first summand and which the second, direct sum is an operation defined on finite sets of arrays (where different elements of a set may be equal to each other in every respect but still have different identities).

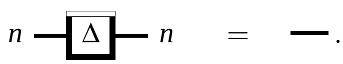
There is a broad rule of associativity: the order in which outer products and contractions are performed does not matter, as long as the at the end, the right arms are connected with each other. And there are distributive rules involving (contracted) outer products and direct sums.

Additional rules apply for the special arrays. If two different diagonal arrays join arms, the result is again a diagonal array. For instance, the following three concatenations of diagonal three-way arrays are identical, and they all evaluate to the (for a given dimension) unique diagonal array or rank 4:

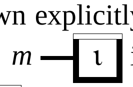
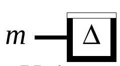
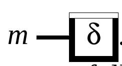


(6)

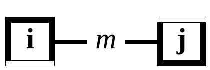
The diagonal array of rank 2 is neutral under concatenation, i.e., it can be written as



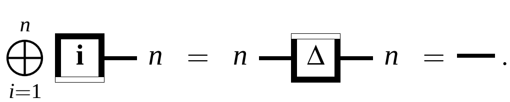
(7)

because attaching it to any array will not change this array. (6) and (7) make it possible to represent diagonal arrays simply as the branching points of several arms. This will make the array notation even simpler. However in the present introductory article, all diagonal arrays will be shown explicitly, and the vector of ones will be denoted  instead of  or perhaps .

Unit vectors concatenate as follows:

 $= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$ (8)

and the direct sum of all unit vectors is the diagonal array of rank 2:



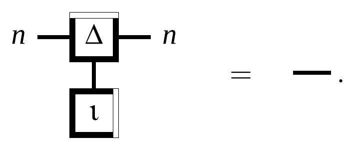
(9)

I am sure there will be modifications if one works it all out in detail, but if done right, the number of axioms should be fairly small. Element-by-element addition of arrays is not an axiom because it can be derived: if one saturates the reference arm of a direct sum with the vector of ones, one gets the element-by-element sum of the arrays in this direct sum. Multiplication of an array by a scalar is also contained in the above system of axioms: it is simply the outer product with an array of rank zero.

Problem 1. *Show that the saturation of an arm of a diagonal array with the vector of ones is the same as dropping this arm.*

Answer. Since the vector of ones is the diagonal array of rank 1, this is a special case of the general concantenation rule for diagonal arrays. □

Problem 2. *Show that the diagonal matrix of the vector of ones is the identity matrix, i.e.,*



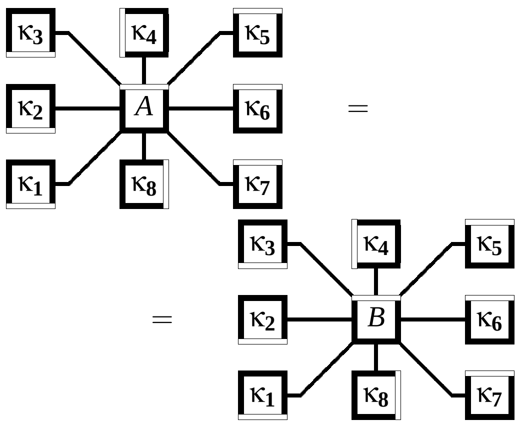
(10)

Answer. In view of (7), this is a special case of Problem 1. □

Problem 3. *A trivial array operation is the addition of an arm of dimension 1; for instance, this is how a n-vector can be turned into a $n \times 1$ matrix. Is this operation contained in the above system of axioms?*

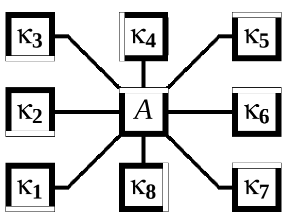
Answer. It is a special case of the direct sum: the direct sum of one array only, the only effect of which is the addition of the reference arm. □

From (9) and (7) follows that every array of rank k can be represented as a direct sum of arrays of rank $k-1$, and recursively, as iterated direct sums of those scalars which one gets by saturating all arms with unit vectors. Hence the following "extensionality property": if the arrays A and B are such that for all possible conformable choices of unit vectors $\kappa_1 \cdots \kappa_8$ follows



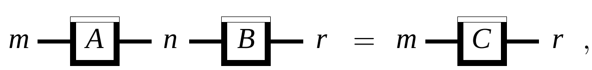
(11)

then $A = B$. This is why the saturation of an array with unit vectors can be considered one of its "elements," i.e.,



$= a_{\kappa_1\kappa_2\kappa_3\kappa_4\kappa_5\kappa_6\kappa_7\kappa_8}$. (12)

From (8) and (9) follows that the concatenation of two arrays by joining one or more pairs of arms consists in forming all possible products and summing over those subscripts (arms) which are joined to each other. For instance, if



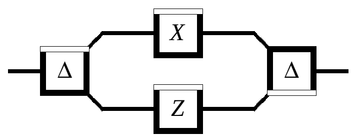
then $c_{\mu\rho} = \sum_{\nu=1}^{n} a_{\mu\nu} b_{\nu\rho}$. This is one of the most basic facts if one thinks of arrays as collections of elements. From this point of view, the proposed notation is simply a graphical elaboration of Einstein's summation convention. But in the holistic approach taken by the proposed system of axioms, which is informed by category theory, it is an implication; it comes at the end, not the beginning.

Instead of considering arrays as bags filled with elements, with the associated false problem of specifying the order in which the elements are packed into the bag, this notation and system of axioms consider each array as an abstract entity, associated with a certain finite graph. These entities can be operated on as specified in the axioms, but the only time they lose their abstract character is when they are fully saturated, i.e., concatenated with each other in such a way that no free arms are left: in this case they become scalars. An array of rank 1 is not the same as a vector, although it can be *represented* as a vector—after an ordering of its elements has been specified. This ordering is not part of the definition of the array itself. (Some vectors, such as time series, have an intrinsic ordering, but I am speaking here of the simplest case where they do not.) Also the ordering of the arms is not specified, and the order in which a set of arrays is packed into its direct sum is not specified either. These axioms therefore make a strict distinction between the abstract entities themselves (which the user is interested in) and their various representations (which the computer worries about).

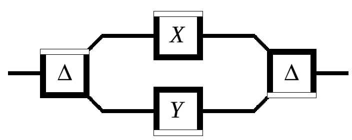
Maybe the following examples may clarify these points. If you specify a set of colors as $\{\text{red}, \text{green}, \text{blue}\}$, then this representation has an ordering built in: red comes first, then green, then blue. However this ordering is not part of the definition of the set; $\{\text{green}, \text{red}, \text{blue}\}$ is the same set. The two notations are two different representations of the same set. Another example: mathematicians usually distinguish between the outer products $A \otimes B$ and $B \otimes A$; there is a "natural isomorphism" between them but they are two different objects. In the system of axioms proposed here these two notations are two different representations of the same object, as in the set example. This object is represented by a graph which has A and B as nodes, but it is not apparent from this graph which node comes first. Interesting conceptual issues are involved here. The proposed axioms are quite different than e.g. [Mor73].

Problem 4. *The trace of the product of two matrices can be written as* $\operatorname{tr}(XY) = \iota^\top(X * Y^\top)\iota$. *Use tile notation to show that this gives indeed* $\operatorname{tr}(XY)$.

Answer. In analogy with (5), the Hadamard product of the two matrices X and Z, i.e., their element by element multiplication, is

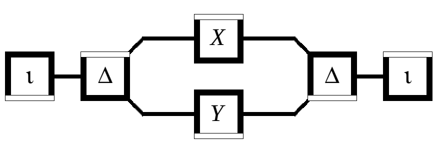
$X * Z =$ 

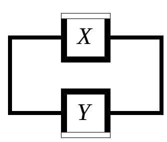
If $Z = Y^\top$, one gets

$X * Y^\top =$  .

$\iota^\top(X * Y^\top)\iota$

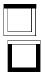
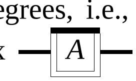
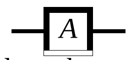
Therefore one gets, using (10):

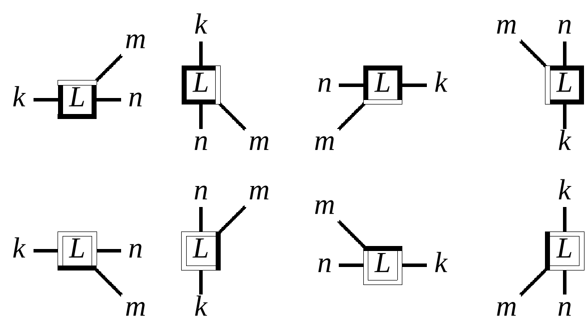
 $=$

$=$  $= \operatorname{tr}(XY)$ (13)

□

4 An Additional Notational Detail

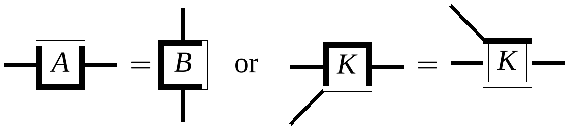
Besides turning a tile by 90, 180, or 270 degrees, the notation proposed here also allows to flip the tile over. The tile  (here drawn without its arms) is simply the tile  laid on its face; i.e., those parts of the frame, which are black on the side visible to the reader, are white on the opposite side and vice versa. If one flips a tile, the arms appear in a mirror-symmetric manner. For a matrix, flipping over is equivalent to turning by 180 degrees, i.e., there is no difference between the matrix  and the matrix . Since sometimes one and sometimes the other notation seems more natural, both will be used. For higher arrays, flipping over arranges the arms in a different fashion, which is sometimes convenient in order to keep the graphs uncluttered. It will be especially useful for differentiation. If one allows turning in 90 degree increments and flipping, each array can be represented in eight different positions, as shown here with a hypothetical array of rank 3:



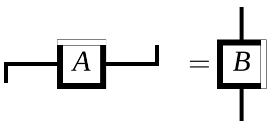
The black-and-white pattern at the edge of the tile indicates whether and how much the tile has been turned and/or flipped over, so that one can keep track which arm is which. In the above example, the arm with dimension k will always be called the West arm, whatever position the tile is in.

5 Equality of Arrays and Extended Substitution

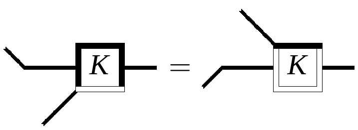
Given the flexibility of representing the same array in various positions for concatenation, specific conventions are necessary to determine when two such arrays in generalized positions are equal to each other. Expressions like



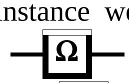
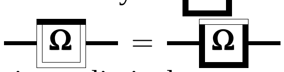
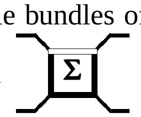
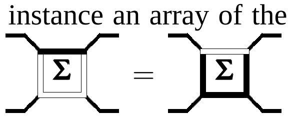
are not allowed. The arms on both sides of the equal sign must be parallel, in order to make it clear which arm corresponds to which. A permissible way to write the above expressions would therefore be



and

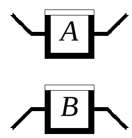


One additional benefit of this tile notation is the ability to substitute arrays with different numbers of arms into an equation. This is also a necessity since the number of possible arms is unbounded. This multiplicity can only be coped with because each arm in an identity written in this notation can be replaced by a bundle of many arms.

Extended substitution also makes it possible to extend definitions familiar from matrices to higher arrays. For instance we want to be able to say that the array  is symmetric if and only if . This notion of symmetry is not limited to arrays of rank 2. The arms of this array may symbolize not just a single arm, but whole bundles of arms; for instance an array of the form  satisfying  is symmetric according to this definition, and so is every scalar. Also the notion of a nonnegative definite matrix, or of a matrix inverse or generalized inverse, or of a projection matrix, can be extended to arrays in this way.

6 Vectorization and Kronecker Product

One conventional generally accepted method to deal with arrays of rank > 2 is the Kronecker product. If A and B are both matrices, then the outer product in tile notation is



(14)

Since this is an array of rank 4, there is no natural way to write its elements down on a sheet of paper. This is where the Kronecker product steps in. The Kronecker product of two matrices is their outer product written again as a matrix. Its definition includes a protocol how to arrange the elements of an array of rank 4 as a matrix. Alongside the Kronecker product, also the vectorization operator is useful, which is a protocol how to arrange the elements of a matrix as a vector, and also the so-called "commutation matrices" may become necessary. Here are the relevant definitions:

6.1 Vectorization of a Matrix

If A is a matrix, then $\text{vec}(A)$ is the vector obtained by stacking the column vectors on top of each other, i.e.,

$$\text{if} \quad A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \quad \text{then} \quad \operatorname{vec}(A) = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}. \tag{15}$$

The vectorization of a matrix is merely a different arrangement of the elements of the matrix on paper, just as the transpose of a matrix.

Problem 5. *Show that* $\operatorname{tr}(B^\top C) = (\operatorname{vec} B)^\top \operatorname{vec} C$.

Answer. Both sides are $\sum b_{ji} c_{ji}$. (26) is a proof in tile notation which does not have to look at the matrices involved element by element. □

6.2 Kronecker Product of Matrices

Let A and B be two matrices, say A is $m \times n$ and B is $r \times q$. Their Kronecker product $A \otimes B$ is the $mr \times nq$ matrix which in partitioned form can be written

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{16}$$

This convention of how to write the elements of an array of rank 4 as a matrix is not symmetric, so that usually $A \otimes C \neq C \otimes A$. Both Kronecker products represent the same abstract array, but they arrange it differently on the page. However, in many other respects, the Kronecker product maintains the properties of outer products.

Problem 6. *[JHG+88, p. 965] Show that*

$$\operatorname{vec}(ABC) = (C^\top \otimes A) \operatorname{vec}(B). \tag{17}$$

Answer. Assume A is $k \times m$, B is $m \times n$, and C is $n \times p$. Write $A = \begin{bmatrix} a_1^\top \\ \vdots \\ a_k^\top \end{bmatrix}$ and $B = \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}$. Then $(C^\top \otimes A) \operatorname{vec} B =$

$$= \begin{bmatrix} c_{11}A & c_{21}A & \cdots & c_{n1}A \\ c_{12}A & c_{22}A & \cdots & c_{n2}A \\ \vdots & \vdots & \ddots & \vdots \\ c_{1p}A & c_{2p}A & \cdots & c_{np}A \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} =$$

$$= \begin{bmatrix} \begin{bmatrix} c_{11}a_1^\top b_1 + c_{21}a_1^\top b_2 + \cdots + c_{n1}a_1^\top b_n \\ c_{11}a_2^\top b_1 + c_{21}a_2^\top b_2 + \cdots + c_{n1}a_2^\top b_n \\ \vdots \\ c_{11}a_k^\top b_1 + c_{21}a_k^\top b_2 + \cdots + c_{n1}a_k^\top b_n \end{bmatrix} \\ \begin{bmatrix} c_{12}a_1^\top b_1 + c_{22}a_1^\top b_2 + \cdots + c_{n2}a_1^\top b_n \\ c_{12}a_2^\top b_1 + c_{22}a_2^\top b_2 + \cdots + c_{n2}a_2^\top b_n \\ \vdots \\ c_{12}a_k^\top b_1 + c_{22}a_k^\top b_2 + \cdots + c_{n2}a_k^\top b_n \end{bmatrix} \\ \vdots \\ \begin{bmatrix} c_{1p}a_1^\top b_1 + c_{2p}a_1^\top b_2 + \cdots + c_{np}a_1^\top b_n \\ c_{1p}a_2^\top b_1 + c_{2p}a_2^\top b_2 + \cdots + c_{np}a_2^\top b_n \\ \vdots \\ c_{1p}a_k^\top b_1 + c_{2p}a_k^\top b_2 + \cdots + c_{np}a_k^\top b_n \end{bmatrix} \end{bmatrix}.$$

One obtains the same result by vectorizing the matrix $ABC =$

$$\begin{bmatrix} a_1^\top b_1 & a_1^\top b_2 & \cdots & a_1^\top b_n \\ a_2^\top b_1 & a_2^\top b_2 & \cdots & a_2^\top b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_k^\top b_1 & a_k^\top b_2 & \cdots & a_k^\top b_n \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{bmatrix}$$

$$= \left[\begin{matrix} a_1^\top b_1c_{11} + a_1^\top b_2c_{21} + \cdots + a_1^\top b_nc_{n1} & \cdots \\ a_2^\top b_1c_{11} + a_2^\top b_2c_{21} + \cdots + a_2^\top b_nc_{n1} & \cdots \\ \vdots & \ddots \\ a_k^\top b_1c_{11} + a_k^\top b_2c_{21} + \cdots + a_k^\top b_nc_{n1} & \cdots \end{matrix}\right.$$

$$\begin{matrix} \cdots & a_1^\top b_1c_{12} + a_1^\top b_2c_{22} + \cdots + a_1^\top b_nc_{n2} & \cdots \\ \cdots & a_2^\top b_1c_{12} + a_2^\top b_2c_{22} + \cdots + a_2^\top b_nc_{n2} & \cdots \\ \ddots & \vdots & \ddots \\ \cdots & a_k^\top b_1c_{12} + a_k^\top b_2c_{22} + \cdots + a_k^\top b_nc_{n2} & \cdots \end{matrix}$$

$$\left.\begin{matrix} \cdots & a_1^\top b_1c_{1p} + a_1^\top b_2c_{2p} + \cdots + a_1^\top b_nc_{np} \\ \cdots & a_2^\top b_1c_{1p} + a_2^\top b_2c_{2p} + \cdots + a_2^\top b_nc_{np} \\ \ddots & \vdots \\ \cdots & a_k^\top b_1c_{1p} + a_k^\top b_2c_{2p} + \cdots + a_k^\top b_nc_{np} \end{matrix}\right].$$

The main challenge in this automatic proof is to fit the many matrix rows, columns, and single elements involved on the same sheet of paper. Among the shuffling of matrix entries, it is easy to lose track of how the result comes about. Later, in equation (27), a compact and intelligible proof will be given in tile notation.

□

6.3 The Commutation Matrix

Besides the Kronecker product and the vectorization operator, also the "commutation matrix" [MN88, pp. 46/7], [Mag88, p. 35] is needed for certain operations involving arrays of higher rank. Assume A is $m \times n$. Then the commutation matrix $K^{(m,n)}$ is the $mn \times mn$ matrix which transforms $\operatorname{vec} A$ into $\operatorname{vec}(A^\top)$:

$$K^{(m,n)} \operatorname{vec} A = \operatorname{vec}(A^\top) \tag{18}$$

The main property of the commutation matrix is that it allows to commute the Kronecker product: For any $m \times n$ matrix A and $r \times q$ matrix B follows

$$K^{(r,m)}(A \otimes B)K^{(n,q)} = B \otimes A \tag{19}$$

Problem 7. *Use* (18) *to compute* $K^{(2,3)}$.

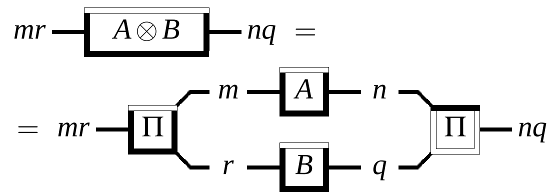
Answer.

$$K^{(2,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

□

6.4 Kronecker Product and Vectorization in Tile Notation

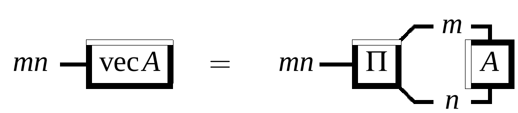
The Kronecker product of $m \,\boxed{A}\, n$ and $r \,\boxed{B}\, q$ is the following concatenation of A and B with members of a certain family of three-way arrays $\Pi^{(i,j)}$:



(21)

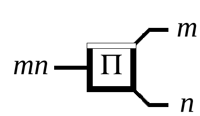
Strictly speaking we should have written $\Pi^{(m,r)}$ and $\Pi^{(n,q)}$ for the two Π-arrays in (21), but the superscripts can be inferred from the context: the first superscript is the dimension of the Northeast arm, and the second that of the Southeast arm.

Vectorization uses a member of the same family $\Pi^{(m,n)}$ to convert the matrix $n \,\boxed{A}\, m$ into the vector

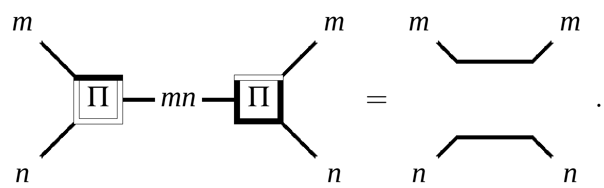


(22)

This equation is a little awkward because the A is here a $n \times m$ matrix, while elsewhere it is a $m \times n$ matrix. It would have been more consistent with the lexicographical ordering used in the Kronecker product to define vectorization as the stacking of the row vectors; then some of the formulas would have looked more natural.

The array $\Pi^{(m,n)} =$  exists for every $m \geq 1$ and $n \geq 1$. The dimension of the West arm is always the product of the dimensions of the two East arms. The elements of $\Pi^{(m,n)}$ will be given in (28) below; but first I will list three important properties of these arrays and give examples of their application.

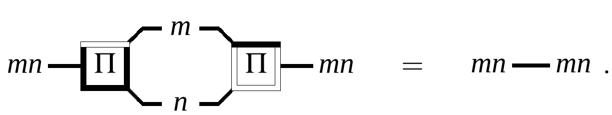
First of all, each $\Pi^{(m,n)}$ satisfies



(23)

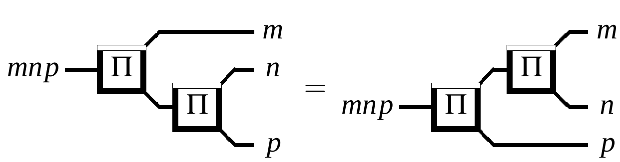
Let us discuss the meaning of (23) in detail. The lefthand side of (23) shows the concatenation of two copies of the three-way array $\Pi^{(m,n)}$ in a certain way that yields a 4-way array. Now look at the righthand side. The arm m — m by itself (which was bent only in order to remove any doubt about which arm to the left of the equal sign corresponds to which arm to the right) represents the neutral element under concatenation (i.e., the $m \times m$ identity matrix). Writing two arrays next to each other without joining any arms represents their outer product, i.e., the array whose rank is the sum of the ranks of the arrays involved, and whose elements are all possible products of elements of the first array with elements of the second array.

The second identity satisfied by $\Pi^{(m,n)}$ is



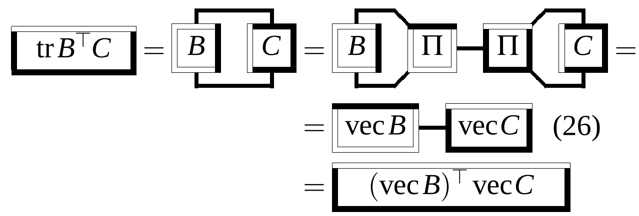
(24)

Finally, there is also associativity:



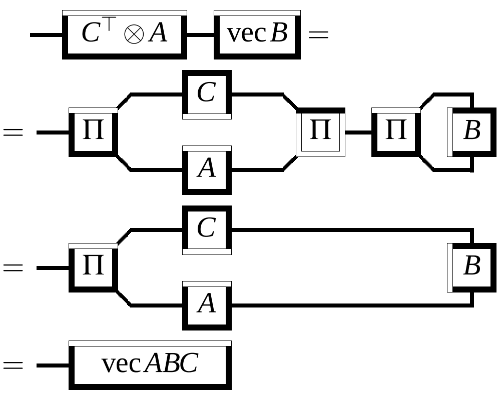
(25)

Here is the answer to Problem 5 in tile notation:



(26)

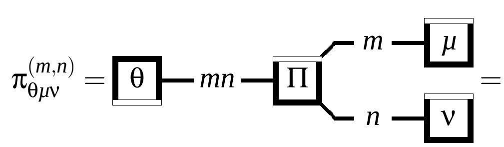
Equation (23) was central for obtaining the result. The answer to Problem 6 also relies on equation (23):



(27)

6.5 Looking Inside the Kronecker Arrays

It is necessary to open up the arrays from the $\boxed{\Pi}$-family and look at them "element by element," in order to verify (21), (22), (23), (24), and (25). The elements of $\Pi^{(m,n)}$, which can be written in tile notation by saturating the array with unit vectors, are



$$= \begin{cases} 1 & \text{if } \theta = (\mu - 1)n + \nu \\ 0 & \text{otherwise.} \end{cases} \qquad (28)$$

Note that for every θ there is exactly one μ and one ν such that $\pi^{(m,n)}_{\theta\mu\nu} = 1$; for all other values of μ and ν, $\pi^{(m,n)}_{\theta\mu\nu} = 0$.

Writing $\boxed{\nu}-\boxed{A}-\boxed{\mu} = a_{\nu\mu}$ and $\boxed{\theta}-\boxed{\mathrm{vec}A} = c_\theta$, (22) reads

$$c_\theta = \sum_{\mu,\nu} \pi^{(m,n)}_{\theta\mu\nu} a_{\nu\mu}, \tag{29}$$

which coincides with definition (15) of $\mathrm{vec}A$.

One also checks that (21) is (16). Calling $A \otimes B = C$, it follows from (21) that

$$c_{\phi\theta} = \sum_{\mu,\nu,\rho,\kappa} \pi^{(m,r)}_{\phi\mu\rho} a_{\mu\nu} b_{\rho\kappa} \pi^{(n,q)}_{\theta\nu\kappa}. \tag{30}$$

For $1 \leq \phi \leq r$ one gets a nonzero $\pi^{(m,r)}_{\phi\mu\rho}$ only for $\mu = 1$ and $\rho = \phi$, and for $1 \leq \theta \leq q$ one gets a nonzero $\pi^{(n,q)}_{\theta\nu\kappa}$ only for $\nu = 1$ and $\kappa = \theta$. Therefore $c_{\phi\theta} = a_{11} b_{\phi\theta}$ for all elements of matrix C with $\phi \leq r$ and $\theta \leq q$. Etc.

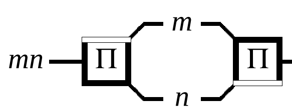
The proof of (23) uses the fact that for every θ there is exactly one μ and one ν such that $\pi^{(m,n)}_{\theta\mu\nu} \neq 0$:

$$\sum_{\theta=1}^{\theta=mn} \pi^{(m,n)}_{\theta\mu\nu} \pi^{(m,n)}_{\theta\omega\sigma} = \begin{cases} 1 & \text{if } \mu = \omega \text{ and } \nu = \sigma \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

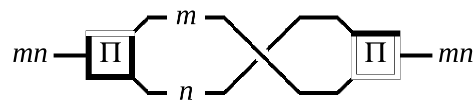
Similarly, (24) and (25) can be shown by elementary but tedious proofs. The best verification of these rules is their implementation in a computer language, see Section 9 below.

6.6 The Commutation Matrix in Tile Notation

The simplest way to represent the commutation matrix $K^{(m,n)}$ in a tile is

$K^{(m,n)} =$  . (32)

This should not be confused with the lefthand side of (24): $K^{(m,n)}$ is composed of $\Pi^{(m,n)}$ on its West and $\Pi^{(n,m)}$ on its East side, while (24) contains $\Pi^{(m,n)}$ twice. We will therefore use the following representation, mathematically equivalent to (32), which makes it easier to see the effects of $K^{(m,n)}$:

$K^{(m,n)} =$  .

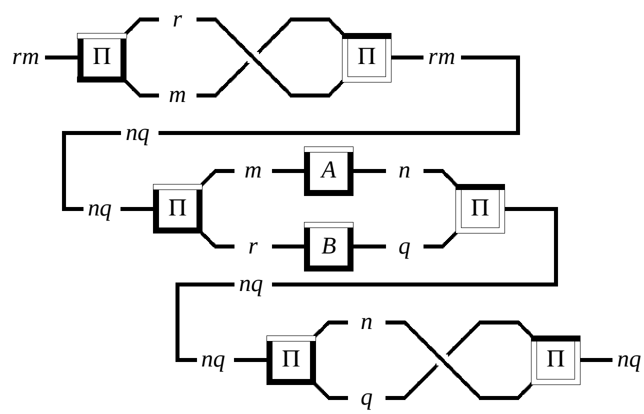
(33)

Problem 8. *Using the definition* (33) *show that* $K^{(m,n)}K^{(n,m)} = I_{mn}$*, the* $mn \times mn$ *identity matrix.*

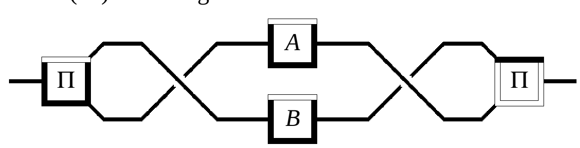
Answer. You will need (23) and (24). □

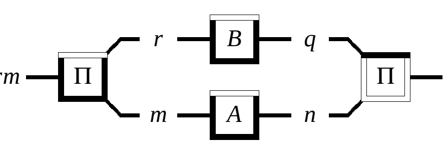
Problem 9. *Prove* (19) *in tile notation.*

Answer. Start with a tile representation of $K^{(r,m)}(A \otimes B)K^{(n,q)}$:



Now use (23) twice to get

$=$ 

$= rm$  nq .

□

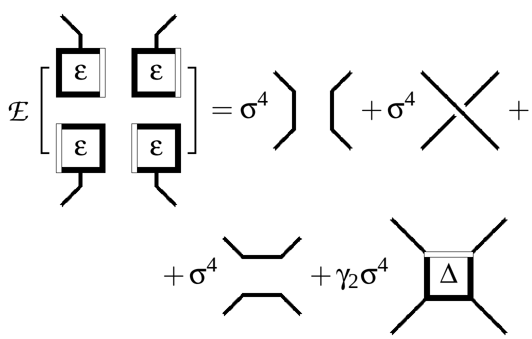
7 Higher Moments of Random Vectors

7.1 Identically Distributed, not Necessarily Normal

Given a random vector ε of independent variables ε_i with zero expected value $\mathrm{E}[\varepsilon_i] = 0$ and identical second and fourth moments. Call $\mathrm{var}[\varepsilon_i] = \sigma^2$ and $\mathrm{E}[\varepsilon_i^4] = \sigma^4(\gamma_2 + 3)$, where γ_2 is the kurtosis. Then the following holds for the fourth moments:

$$\mathrm{E}[\varepsilon_i\varepsilon_j\varepsilon_k\varepsilon_l] = \begin{cases} \sigma^4(\gamma_2+3) & \text{if } i=j=k=l \\ \sigma^4 & \text{if } i=j\neq k=l \\ & \text{ or } i=k\neq j=l \\ & \text{ or } i=l\neq j=k \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

It is not an accident that (34) is given element by element and not in matrix notation. It is not possible to do this, not even with the Kronecker product. But it is easy in tile notation:

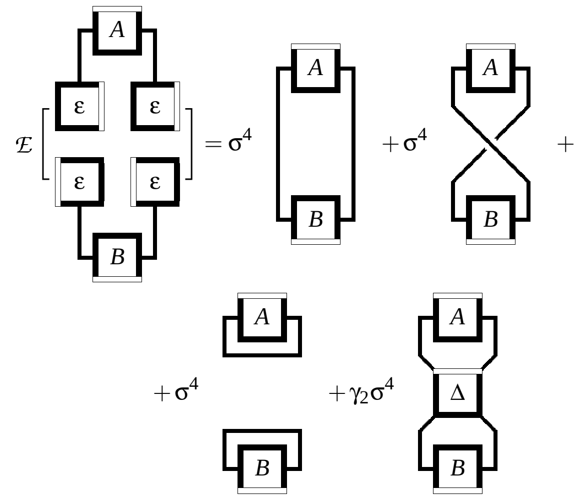


(35)

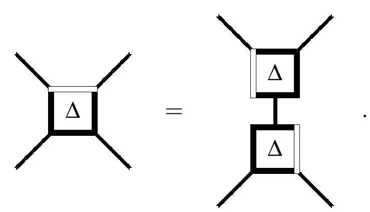
Problem 10. *[Seb77, pp. 14–16 and 52] Show that for any symmetric $n \times n$ matrices A and B, whose vectors of diagonal elements are a and b,*

$$\mathrm{E}[(\varepsilon^\top A\varepsilon)(\varepsilon^\top B\varepsilon)] = \\ = \sigma^4\Big(\mathrm{tr}A\,\mathrm{tr}B + 2\,\mathrm{tr}(AB) + \gamma_2 a^\top b\Big). \tag{36}$$

Answer. (36) is an immediate consequence of (35); this step is now trivial due to linearity of the expected value:



The first term is $\mathrm{tr}AB$. The second is $\mathrm{tr}AB^\top$, but since A and B are symmetric, this is equal to $\mathrm{tr}AB$. The third term is $\mathrm{tr}A\,\mathrm{tr}B$. What is the fourth term? Diagonal arrays exist with any number of arms, and any connected concatenation of diagonal arrays is again a diagonal array, see (6). For instance,

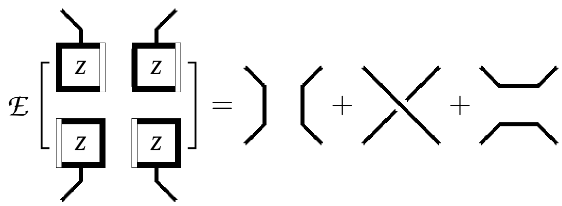


(37)

From this together with (4) one can see that the fourth term is the scalar product of the diagonal vectors of A and B. □

7.2 Multivariate Normal Distribution

I will give a brief overview in tile notation of the higher moments of the multivariate standard normal z. All odd moments disappear, and the fourth moments are

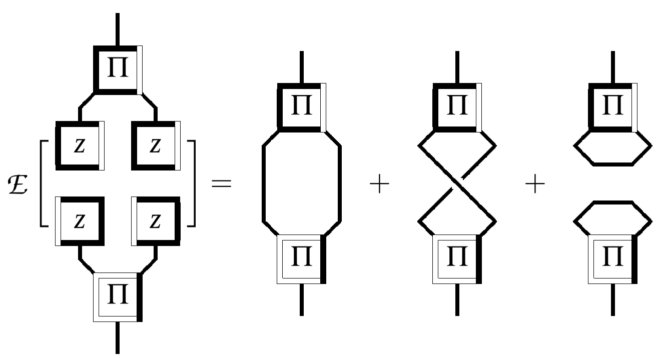


(38)

Compared with (35), the last term, which depends on the kurtosis, is missing. What remains is a sum of outer products of unit matrices, with every possibility appearing exactly once. In the present case, it happens to be possible to write down the four-way arrays in (38) in terms of Kronecker products and the commutation matrix $K^{(n,n)}$ introduced in (19): It is

$$\mathcal{E}[(zz^\top)\otimes(zz^\top)] = I_{n^2} + K^{(n,n)} + (\mathrm{vec}[I_n])(\mathrm{vec}[I_n])^\top \tag{39}$$

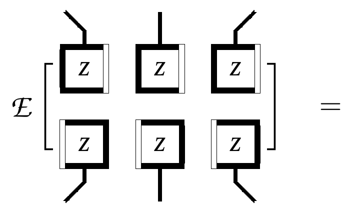
Compare [Gra83, 10.9.2 on p. 361]. Here is a proof of (39) in tile notation:



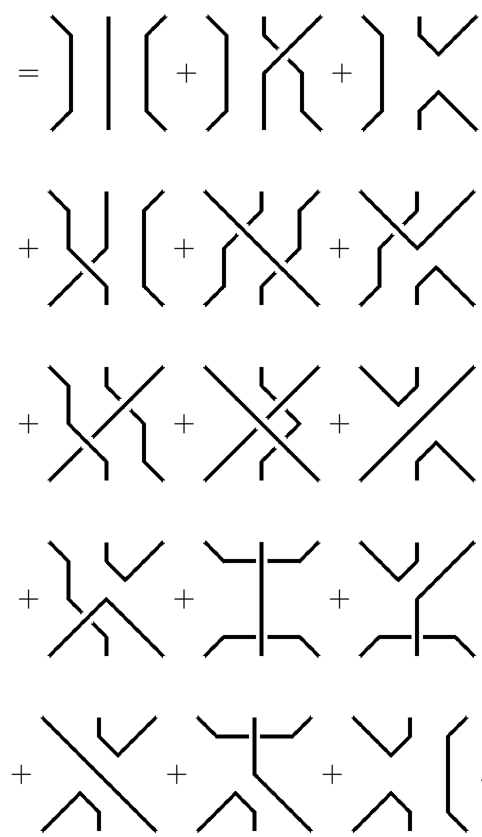
(40)

The first term is I_{n^2} due to (24), the second is $K^{(n,n)}$ due to (33), and the third is $(\mathrm{vec}[I_n])(\mathrm{vec}[I_n])^\top$ because of (22). Graybill [Gra83, p. 312] considers it a justification of the interest of the commutation matrix that it appears in the higher moments of the standard normal. In my view, the commutation matrix is ubiquitous only because the Kronecker-notation blows up something as trivial as the crossing of two arms into a mysterious-sounding special matrix.

The sixth moments of the standard normal, in analogy to the fourth, are the sum of all the different possible outer products of unit matrices:



(41)



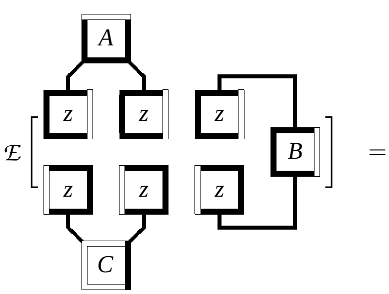
Here is the principle how these were written down: Fix one branch, here the Southwest branch. First combine the Southwest branch with the Northwest one, and then you have three possibilities to pair up the others as in (38). Next combine the Southwest branch with the North branch, and you again have three possibilities for the others. Etc. This gives 15 possibilities altogether.

This can no longer be written as a Kronecker product, see [Gra83, 10.9.4 (3) on p. 363]. However (41) can be applied directly, for instance in order to show (42), which is [Gra83, 10.9.12 (1) on p. 368]:

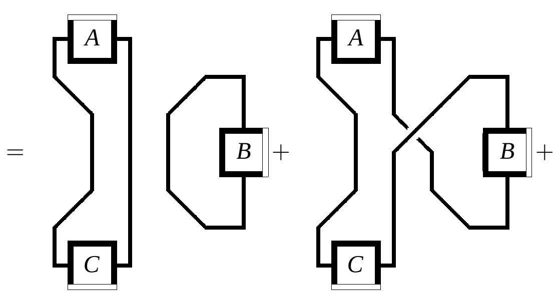
$$\begin{aligned}&\mathrm{E}[(z^\top Az)(z^\top Bz)(z^\top Cz)] =\\ &= \mathrm{tr}(A)\,\mathrm{tr}(B)\,\mathrm{tr}(C) + 2\,\mathrm{tr}(A)\,\mathrm{tr}(BC) + 2\,\mathrm{tr}(B)\,\mathrm{tr}(AC) +\\ &\qquad + 2\,\mathrm{tr}(C)\,\mathrm{tr}(AB) + 8\,\mathrm{tr}(ABC). \qquad (42)\end{aligned}$$

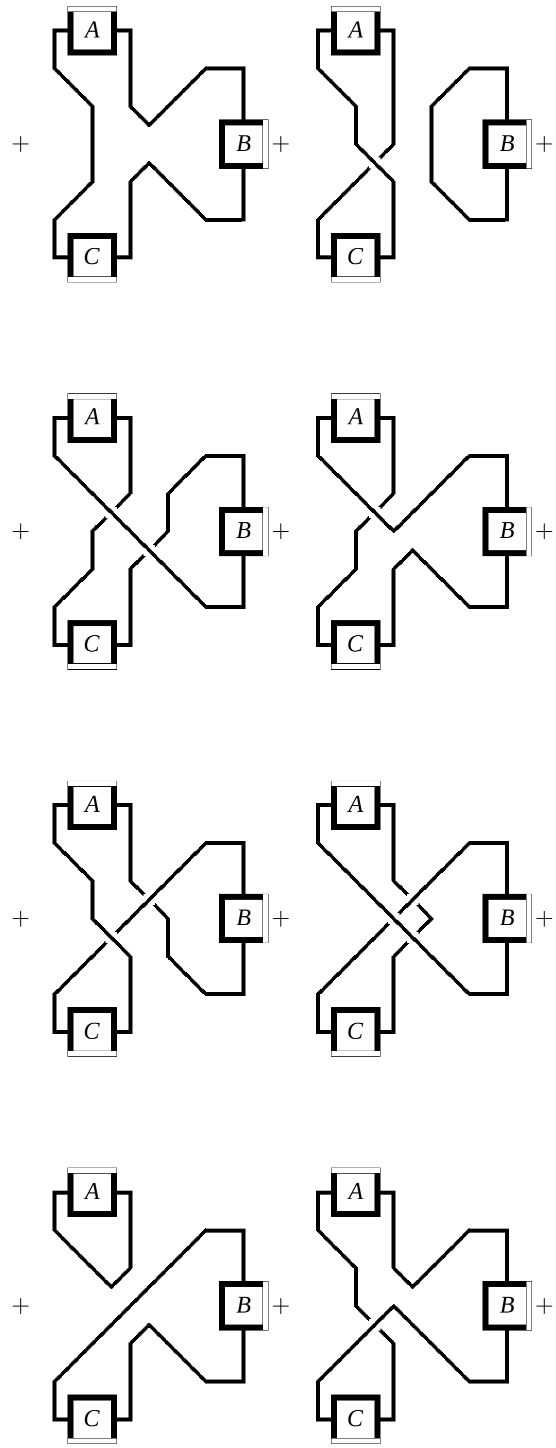
Problem 11. *Assuming that A, B, and C are symmetric matrices, prove* (42) *in tile notation.*

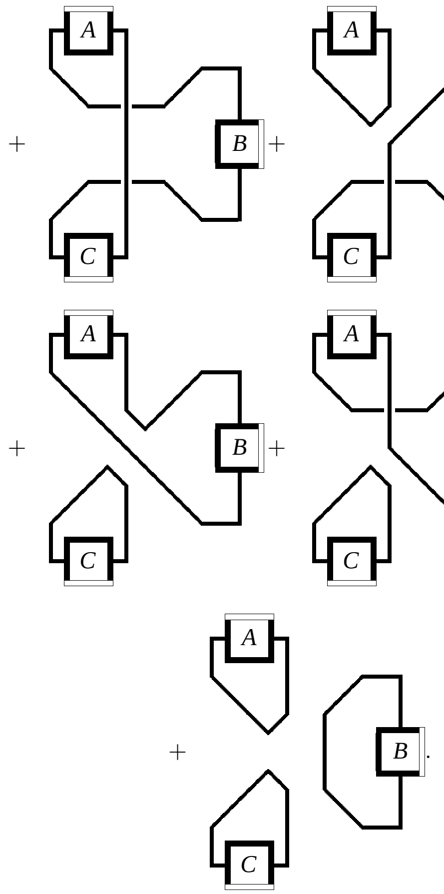
Answer.



(43)





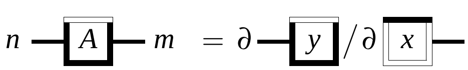


These 15 summands are, in order, $\operatorname{tr}(B)\operatorname{tr}(AC)$, $\operatorname{tr}(ABC)$ twice, $\operatorname{tr}(B)\operatorname{tr}(AC)$, $\operatorname{tr}(ABC)$ four times, $\operatorname{tr}(A)\operatorname{tr}(BC)$, $\operatorname{tr}(ABC)$ twice, $\operatorname{tr}(A)\operatorname{tr}(BC)$, $\operatorname{tr}(C)\operatorname{tr}(AB)$ twice, and $\operatorname{tr}(A)\operatorname{tr}(B)\operatorname{tr}(C)$.

□

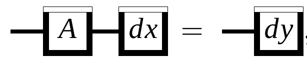
8 Array Differentiation

Here is the tile notation for matrix differentiation: If n —$\boxed{y}$ depends on m —$\boxed{x}$, then



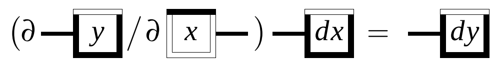
(44)

is that array which satisfies



(45)

i.e.,



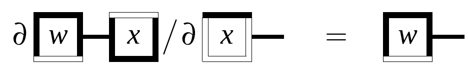
(46)

Extended substitutability applies here: n —$\boxed{y}$ and m —$\boxed{x}$ are not necessarily vectors; the arms with dimension m and n can represent different bundles of several arms.

The simplest matrix differentiation rule, for $y = w^\top x$, is

$$\partial w^\top x / \partial x^\top = w^\top \tag{47}$$

In tiles it is

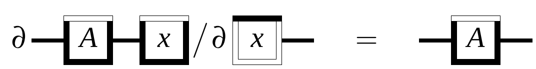


(48)

Here is the most basic matrix differentiation rule: if $y = Ax$ is a linear vector function, then its derivative is that same linear vector function:

$$\partial Ax / \partial x^\top = A, \tag{49}$$

or in tiles

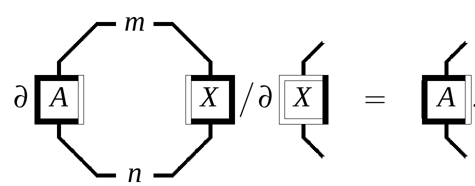


(50)

Problem 12. *Show that*

$$\frac{\partial \operatorname{tr} AX}{\partial X^\top} = A. \tag{51}$$

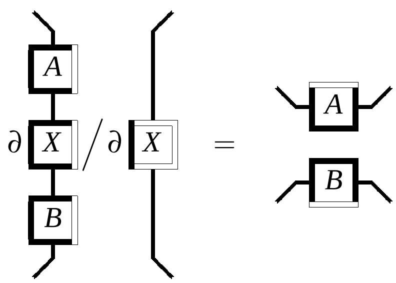
In tiles it reads



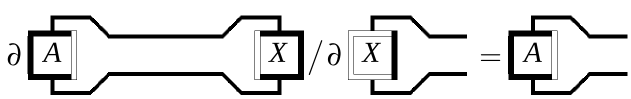
(52)

Answer. $\mathrm{tr}(AX) = \sum_{i,j} a_{ij}x_{ji}$ i.e., the coefficient of x_{ji} is a_{ij}. □

Here is a differentiation rule for a matrix with respect to a matrix, first written element by element, and then in tiles: If $Y = AXB$, i.e., $y_{im} = \sum_{j,k} a_{ij}x_{jk}b_{km}$, then $\frac{\partial y_{im}}{\partial x_{jk}} = a_{ij}a_{km}$, because for every fixed i and m this sum contains only one term which has x_{jk} in it, namely, $a_{ij}x_{jk}b_{km}$. In tiles:

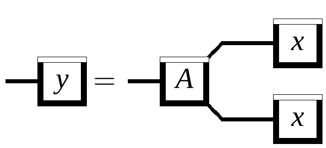

(53)

Equations (52) and (53) can be obtained from (48) and (50) by extended substitution, since a bundle of several arms can always be considered as one arm. For instance, (52) can be written

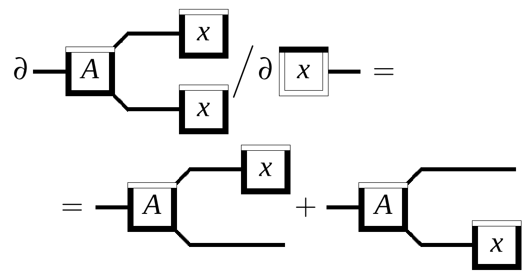


and this is a special case of (48), since the two parallel arms can be treated as one arm. With a better development of the logic underlying this notation, it will not be necessary to formulate them as separate theorems; all matrix differentiation rules given so far are trivial applications of (50).

Here is one of the basic differentiation rules for a *bilinear* array concatenation: if

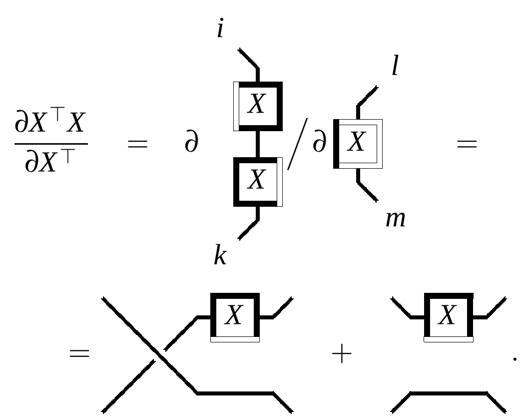

(54)

then


(55)

Proof. $y_i = \sum_{j,k} a_{ijk}x_jx_k$. For a given i, this has x_p^2 in the term $a_{ipp}x_p^2$, and it has x_p in the terms $a_{ipk}x_px_k$ where $p \neq k$, and in $a_{ijp}x_jx_p$ where $j \neq p$. The derivatives of these terms are $2a_{ipp}x_p + \sum_{k\neq p} a_{ipk}x_k + \sum_{j\neq p} a_{ijp}x_j$, which simplifies to $\sum_k a_{ipk}x_k + \sum_j a_{ijp}x_j$. This is the i,p-element of the matrix on the rhs of (55). □

But there are also other ways to have the array X occur twice in a concatenation Y. If $Y = X^\top X$ then $y_{ik} = \sum_j x_{ji}x_{jk}$ and therefore $\partial y_{ik}/\partial x_{lm} = 0$ if $m \neq i$ and $m \neq k$. Now assume $m = i \neq k$: $\partial y_{ik}/\partial x_{li} = \partial x_{li}x_{lk}/\partial x_{li} = x_{lk}$. Now assume $m = k \neq i$: $\partial y_{ik}/\partial x_{lk} = \partial x_{li}x_{lk}/\partial x_{lk} = x_{li}$. And if $m = k = i$ then one gets the sum of the two above: $\partial y_{ii}/\partial x_{li} = \partial x_{li}^2/\partial x_{li} = 2x_{li}$. In tiles this is


(56)

This rule is helpful for differentiating the multivariate Normal likelihood function.

A computer implementation of this tile notation should contain algorithms to automatically take the derivatives of these array concatenations.

9 Internet Resources

The TEX-macros to typeset the tiles are available at

`www.econ.utah.edu/ehrbar/arca.sty`.

My Econometrics class notes

`www.econ.utah.edu/ehrbar/ecmet.pdf`

(5 Megabytes) contain more examples of this notation. A pilot implementation of this type of array concatenation in R is available as R-package on my web site. It can be downloaded using the following R-command: `install.packages("arca", contriburl = "http://www.econ.utah.edu/ehrbar/R", lib = "/usr/lib/R/library")` (you may need a different `lib` argument for your system). Besides special functions building the Π-arrays for Kronecker products, this package has one function `arca` which takes as arguments several arrays, together with a vector indicating which arms of which arrays are to be joined together. Several contractions can be specified in the same function call. This is not a production version; I merely used it to check the identities in Section 6.

References

[Gra83] Franklin A. Graybill. *Matrices with Applications in Statistics*. Wadsworth and Brooks/Cole, Pacific Grove, CA, second edition, 1983.

[JHG+88] George G. Judge, R. Carter Hill, William E. Griffiths, Helmut Lütkepohl, and Tsoung-Chao Lee. *Introduction to the Theory and Practice of Econometrics*. Wiley, New York, second edition, 1988.

[Mag88] Jan R. Magnus. *Linear Structures*. Oxford University Press, New York, 1988.

[MN88] Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, Chichester, 1988.

[Mor73] Trenchard More, Jr. Axioms and theorems for a theory of arrays. *IBM Journal of Research and Development*, 17(2):135–175, March 1973.

[MS86] Parry Hiram Moon and Domina Eberle Spencer. *Theory of Holors; A Generalization of Tensors*. Cambridge University Press, 1986.

[Seb77] G. A. F. Seber. *Linear Regression Analysis*. Wiley, New York, 1977.

[SS35] J. A. Schouten and Dirk J. Struik. *Einführung in die neuren Methoden der Differentialgeometrie*, volume I. 1935.