Journal of Cybernetics, 1972, Vol. 2, No.1, pp. 50-65

Further Towards a Triadic Calculus

Part 1

Christopher R. Longyear [*]

Foreword

This paper, presented in two parts, attempts to augment an initial formulation of a calculus for triadas by Warren McCulloch and Roberto Moreno-Diaz, published as Quarterly Progress Report No. 84 (QPR 84), from the Research Laboratory of Electronics, Massachusetts Institute of Technology, January 15, 1967, pp. 335-346.

I have tried to follow closely the organization, the material, and even the wording[1] of that paper wherever possible, though further developments have made many of the conclusions reached here quite different from those in QPR 84. Major points of departure in the present paper are in the graphical notation, which makes the ordering of the elements of a triada much more visible and manipulable geometrically, and in the two additional restrictions imposed on complex triadas involving interfaces and internal connections. The section on computer-aided triada logic development is also entirely new.

Warren McCulloch introduced me to his and Moreno-Diaz' work on their calculus for triadas when I was a research fellow in Mathematical Linguistics at Harvard from 1967 to 1969. My interests were linguistics and computers, particularly in how to develop data base structures so that they could represent adequately the rich information conveyed in natural language. Warren McCulloch was generous enough to encourage my attempts to adapt the graphical devices I had used to represent complex relations in my computer data bases, all the more so when certain discrepancies in my derivations compared with theirs came to light.

Not until after his fatal heart attack have I had a chance to resume work on this rotation, and I, as much as many of his friends of longer standing, sorely miss both his rich encouragement and his sharp but always stimulating criticism. Joseph Goguen of the University of Chicago commented on QPR 84, and his suggestion of using transistor and socket has been accepted as well as a modification of his representation of the identities I_c, I_λ, and I_ρ. Remaining error and infelicities are mine.

On a Calculus for Tridas

*Introduction**

De Morgan, obstructed by his terminology, thought the construction of a logic of relations impossible. A quarter of a century later, C. S. Peirce initiated it. Repeated attempts to understand him failed because in every paper he changed his terminology. It was not until we attempted to formulate family relations in Gilstrap's matricial calculus that he and we were able to understand Peirce, who had actually invented such a calculus and extended it to three-dimensional arrays which we call "mints." It is now clear what he had done and what stopped him. He also used a symbolism in molecular diagrams which is transparent. Finally, he interpreted these in terms of sentences containing n blanks to be filled by the names of things in the universe of discourse. Whether these be real or imaginary is immaterial to this calculus, which therefore can cope with intension, not merely extension, and hence is of value in psychophysiological contexts. Many theorems not involving negation can now be proved, but negation is not simple and we are struggling to discover its multifarious consequences. At the moment, we want to present the following useful results.

---

* Requests for reprints should be sent to Dr. Christopher R. Longyear, English Dept., University of Washington, Seattle, Washington 98195.

1 Sections of this paper, identified by an asterisk* and small print, were reproduced with permission from the above-mentioned report QPR 84.

Triadas

The triada is a structure of any kind involving three elements or members of a given set at a time. For example, "a gives b to c" is a triada, G , involving the objects a , b, and c . Peirce suggested different ways to develop a calculus for triadas, i.e., "an art of drawing inferences." For cases in which triadas are of the nature of the previously mentioned example, i.e., of the nature of a sentence or phrase with three blanks that are to be filled by particular members of a given set, a calculus may be developed that is similar to the calculus of functional propositions of three arguments – or you have Boolean tensors of rank 3 – but that is richer in possibilities and consequences. One of the ways to develop such a calculus is to consider two kinds of variables or symbols, one for the elements of the set where the triadas apply (here lower-case letters are used), and the other for the triadas themselves (represented here by upper-case letters). A calculus involving only upper-case letters will be called a "proper calculus for triadas."

In the process of constructing the calculus, operations on or among triadas are defined which have a definite meaning. The object of the calculus is then to combine the operations and to obtain conclusions or theorems about the combined operations of triadas. We concern ourselves here only with closed operations, i.e., operations on or among triadas, which again generate triadas.

Definitions and Operations

A triada is a sentence or phrase with three blanks that are to be filled with specific names of objects, or members of a given set, in order for the sentence to have meaning. For example, if in the sentence "a gives b to c ," we delete the names a , b , and c , we end with the triada

"__gives__to__" (S1)

We denote by i, j, and k the first, second, and third blanks, respectively. Furthermore, we represent the triada by G_{ijk}, i.e., G_{ijk} means

"__gives__to__" (S2)

If we want to express the fact that the particular member a gives the particular member b to the particular one c, we shall write G_{abc}. Therefore, the subscripts are regarded as variables, as are the blanks. Somewhere in the calculus we shall be able to delete subscripts without confusion, to obtain the calculus proper.

Two triadas are said to be equal if they have the same meaning, i.e., they originate equivalent sentences, when applied to any three objects in the same order. We represent the equality of two triadas by separating them with the sign =. In any expression in which triadas appear, any of them can be replaced by an equivalent one. For example, the triadas

"__gives__to__" (S3)

and

"__is given to__by__" (S4)

are not equal because when applied to objects a, b, and c in this order the resulting sentences do not have the same meaning; however, the triadas

"__gives__to__" (S5)

and

"__is identical to the one who gives__to__" (S6)

are equal.

end of *

Frequently, a graphical representation will aid us in making a point. These diagrams are further developments from those used by Peirce, and are similar to those developed by Longyear in the different context of "triangle" data base representation. The important difference between Peirce's conventions and those used here is that we mark explicitly the *direction* of progression.

For example, if two triadas, G_{abc} and E_{abc}, are:

G_{abc} = "a gives b to c" (S7)

and

E_{abc} = "a gives to b by c" (S8)

then the meanings are not equal; graphically, we may diagram the two triadas and their inequality as shown in Fig. 1.

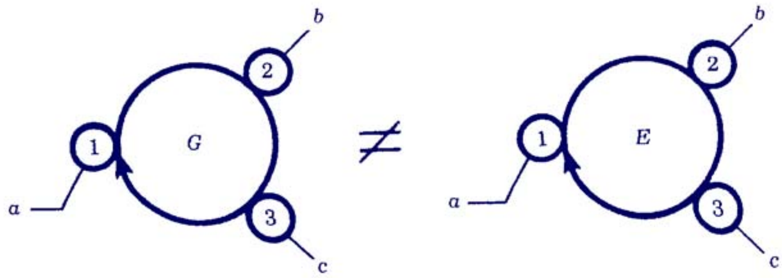

FIG. 1.

But if

F_{abc} = "a is identical to the one who gives b to c" (S9)

then the meanings of triadas G_{abc} and F_{abc} are equal. This equality is shown graphically in Fig. 2.

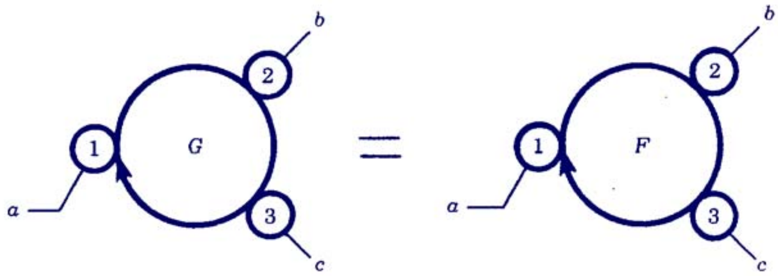

FIG. 2.

Note also that although the meanings of the two following sentences are equivalent, the order of the elements is not the same, and therefore triadas G_{abc} and G'_{bca} are *not* equal.

G_{abc} = "a gives b to c" (S10)

G'_{bca} = "b gives c to a" (S11)

These are shown diagrammatically in Fig. 3.

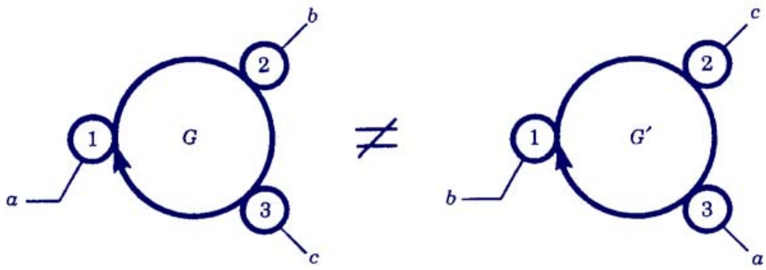

FIG. 3.

For one n-ada to equal another, seven requirements must be met:

1. Both n-adas must have n external elements.[2] (A triada may equal a triada, but in general, a triada does not equal a dyada.)
2. Both n-adas must share the *same* n elements. (a, b, c could = a, b, c; but a, b, c does not generally = a, b, d.)
3. Both n-adas must have the same *order* of elements (a, b, c could = a, b, c; but a, b, c does not generally = a, c, b.)
4. Both n-adas must share the same internal *structure*.[3] (A Δ-product could = another Δ-product; but in general, a Δ-product does not = a Δ-sum.)
5. Both n-adas must share the same meanings.[4]
6. Both the internal and the external part of an arm passing through the envelope of a complex n-ada must be of the same relative order for such an interface to be well-formed. (A second arm relative to an outsiden-ada must also be a second arm relative to its insiden-ada.)
7. Any internal connections (colligative terms) or relative connections (binary operations) must always be of the same relative order to be well-formed. (A third arm relative to its n-ada may connect only to a third arm of *its* n-ada.)

For triadas, then, the seven requirements are met when:

1. Both triadas have *three* external elements.
2. Both triadas have the *same* three external elements.
3. Both triadas have the same *order* of their three elements.
4. Both triadas have the same *internal* structure.
5. Both triadas have the same *meaning*.
6. Both sides of a "*shift*" have arms of the same order.
7. Both sides of an internal connection have arms of the same order.

For our graphic representation, we should note that the only significant features are the name of the triad (e.g., "G"), the number of the elements to which it relates (i.e., the direction of the arrow as well as the point of origin of the triad, shown graphically by the arrowhead and by the first arm bent toward the direction of the second element). All the representations of Fig. 4 have the same name, the same origin, and the same sequence of elements. They are therefore entirely equivalent, as are any other representations which maintain name, origin, and sequence of elements.

---

[2] In complex triadas (which will become more apparent below, when we look at non-relative and relative sums and products), there may be internal as well as external elements.

[3] In complex triadas, there may be internal relationships which affect the meaning of the triada as a whole.

[4] In deciding whether two sentences have the same meaning, we need be concerned only with overtly expressed meanings, and not with such questions as rhetorical emphasis or gracefulness of expression. If one sentence is an accurate paraphrase of the other, they are said to have equal meanings. Thus, S7 and S9 are said to have the same meaning.

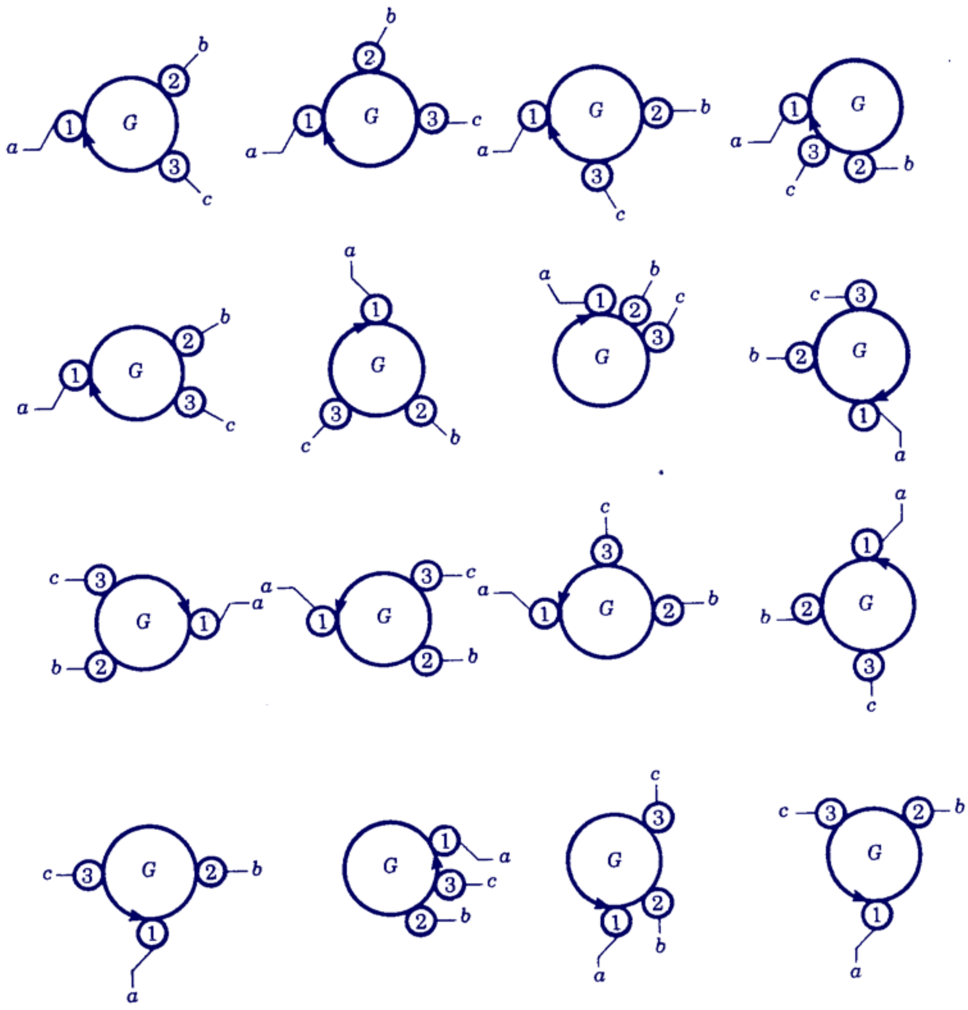

FIG. 4.

* We now distinguish three kinds of closed operations. These are unary operations, involving one triada; binary, or nonrelative, involving two triadas; and triadic, or relative, involving three triadas.

Unary Operations

Rotation is the clockwise rotation of the order of the blanks in the triada one step, keeping the meaning constant. For example, let

G_{ijk} be "__gives__to__" (S12)

Its rotation, represented by $\widehat{G_{ijk}}$, is the triada

"to__,__gives__", (S13)

According to the definition of equality, we may write

$$\widehat{G_{ijk}} = G_{kij} \qquad [E1]$$

which indicates that if G applies to objects a , b , and c in this order, then $\widehat{G}$ applies to them in the order c , a, b ; furthermore, the sentences representing G and $\widehat{G}$ must be paraphrases of each other in order to preserve meaning.

end of *

For example, if G_{abc} is the sentence,

"<u>a</u> gives <u>b</u> to <u>c</u>" (S14)

then $\widehat{G_{abc}}$ or G_{cab} is the sentence, "to $\underline{c}$, $\underline{a}$ gives $\underline{b}$" (S15)

Rotation is an operation to be carried out on the triada to which it is applied.

Thus, $\widehat{G_{ijk}}$ means "the triada G_{ijk} is to be rotated." Graphically, Fig. 5 becomes, when the operation is carried out, Fig. 6a or 6b.

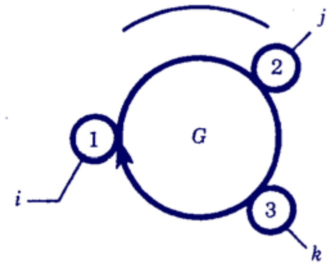

FIG. 5.

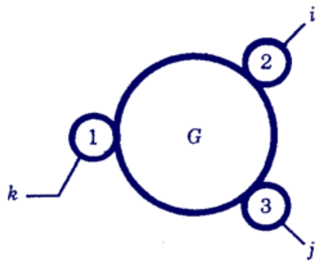

FIG. 6a.

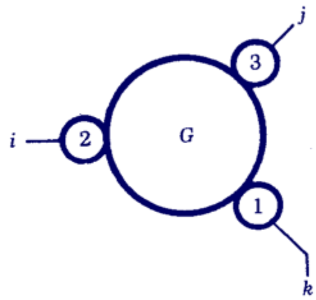

FIG. 6b.

Graphically, rotation is equivalent to rotating the "socket" of the "transistor" one step. (See Fig. 6c.)

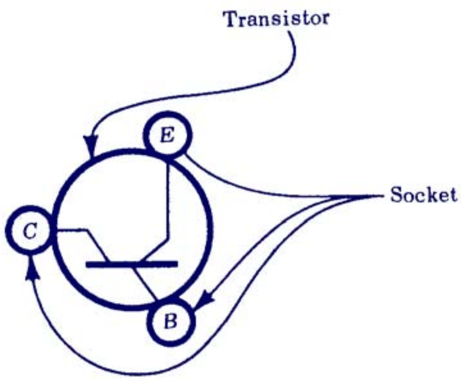

FIG. 6c.

If we think of the external elements to which the arms of the "transistor" lead as being the "socket," then we rotate the "socket" one notch in the direction of the arrow. This graphical procedure gives us the result shown in Fig. 6a. If, on the other hand, we rotate the "transistor" instead, we must rotate it one notch *against* the direction of the arrow. The graphical result is shown in Fig. 6b, which is, of course, entirely equivalent to that of Fig. 6a.

It is useful to keep separate the operations to be done from those which have already been carried out. G, in our Figs. 5 and 6, are not the same sentences until the rotation of Fig. 5 has been carried out. As long as there is some obvious way in which we may recognize the original order of the elements, difficulties may be minimized. But when we get to complex triadas below, such ordering becomes less clearly visible, and we may want sometimes to refer to a triada to which some operation is to be carried out (as $\widehat{G_{ijk'}}$ above), and sometimes to a triada which notes that some operation has been carried out on it already. For this purpose, we adopt the following convention: Diagrammed triadas on which operations are to be carried out are marked *outside* the circle of the graphic representation, suggesting the same sort of diacritical notation as for the subscripted algebraic representation.

Thus, Fig. 5 means that the triada G_{ijk} is to be rotated, just as $\widehat{G_{ijk}}$ means that the triada G_{ijk} is to be rotated. Sometimes, when we want to keep the "history" of some operation that has already been carried out, we may put a diacritical mark *inside* the circle of the graphic representation. Thus, we could also diagram Fig. 6a, for example, as Fig. 7.

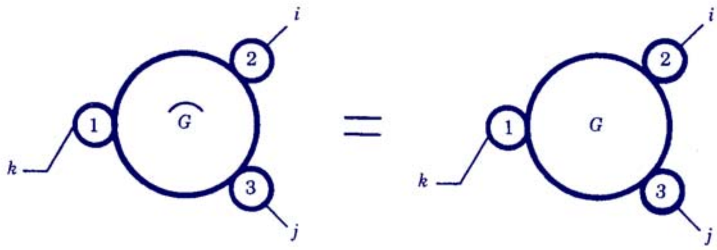

FIG. 7.

To repeat: diacritical marks *inside* the circle are basically meaningless, and refer only to some history of earlier operations. In other words, the elements to which the arms point in the designated sequence define the triada in its actual manifestation. In the case of a diacritical mark *outside* the circle, however, the operation implied by the mark is to be carried out on the triada, including its elements. Note that it is *not* the case that $\overset{\frown}{G_{ijk}} = G_{ijk}$ *nor* is it the case that $\overset{\frown}{G_{kij'}} = G_{ijk}$. In fact, the rotation of $G_{kij'}$ symbolized $\overset{\frown}{G_{kij}}$ is equivalent to the double rotation of Giik' $\overset{\frown}{\overset{\frown}{G_{ijk}}}$ and is, in fact, equal to G_{jki}. This double rotation is shown graphically in Fig. 8, which illustrates

$$\overset{\frown}{\overset{\frown}{G_{ijk}}} = G_{jki} \qquad \text{[E2]}$$

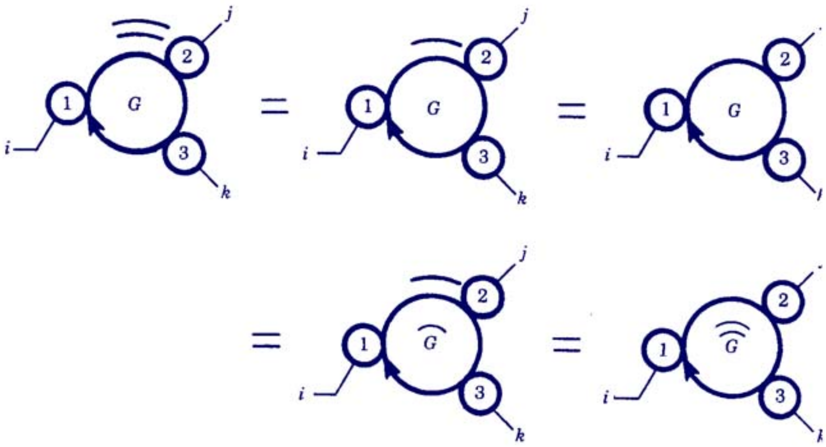

FIG. 8.

Reflection is the "flipping over" of the triada, using the second element as the axis of rotation. In effect, the first and the third elements change positions. For example, the reflection of G_{ijk}

"__gives__to__" (S16)

is $G_{ijk'}$ or

"__is given__by__" (S17)

We write

$$= \overset{\smile}{G_{ijk}} \quad G_{kji} \qquad \text{[E3]}$$

and represent it graphically as in Fig. 9.

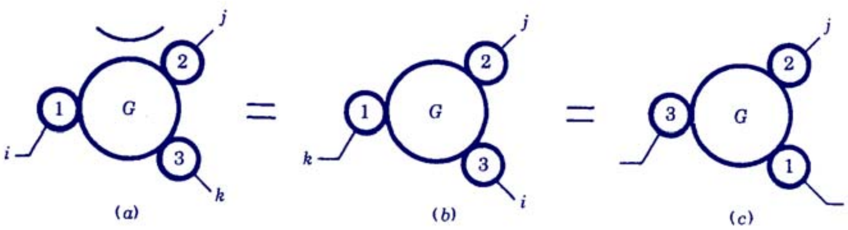

FIG. 9.

The graphic Fig. 9b represents the graphical operation of flipping over the "socket" of the "transistor" so that the third and the first are interchanged, leaving the middle one untouched. Figure 9c is equivalent to this, but is achieved graphically by flipping over the "transistor" so that its origin is now connected to what used to be its third element. To persuade oneself that the two figures, 9b and 9c, are equivalent, one needs only to imagine looking at Fig. 9b from underneath the page, seeing Fig. 9(c).

* By iteratively applying each unary operation to a triada, it is easy to see that

$$G_{ijk} = \breve{\breve{G}}_{ijk} \qquad [E4]$$

and that

$$G_{ijk} = \overset{\frown}{\overset{\frown}{\overset{\frown}{G}}}_{ijk} \qquad [E5]$$

end of *

Graphically, these equalities are shown in Fig. 10.

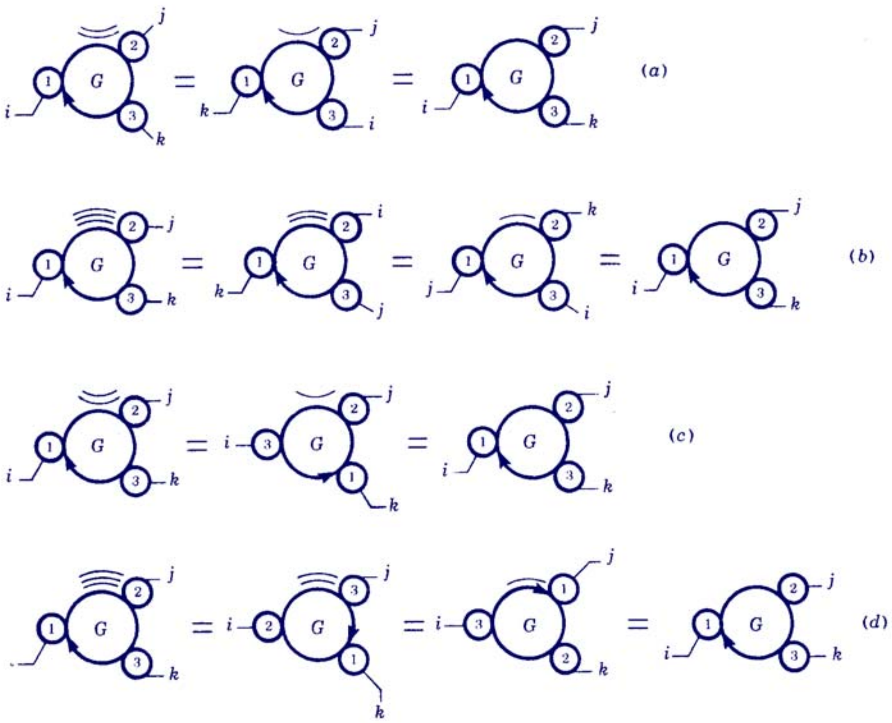

FIG. 10.

Figure 10a shows a double reflection graphically executed by flipping over the "socket", while Fig. 10c shows the same effect due to flipping over the "transistor." Figure lob shows the effect of rotating the "socket" three times in the direction of the arrow, while Fig. 10d shows what happens when the "transistor" is rotated three times in the direction against the arrow. Figures 10a and l0c are entirely equivalent as are 10b and 1 0d. In this particular instance, in fact every diagram on Fig. 10 is equal to every other one.

For the record, let us apply the test of equality of triadas (as listed above Clearly, all the triadas illustrated in Fig. 10a-d have the same three, extern elements, and the same internal structure (which is simple, for all of them). To check the sameness of the meaning, we note the following sentences, used examples.

G_{ijk}: $\underline{a}$ gives $\underline{b}$ to $\underline{c}$ (S18)

G_{kji}: to $\underline{c}$ is given $\underline{b}$ by $\underline{a}$ (S19)

G_{kij}: to $\underline{c}$, $\underline{a}$ gives $\underline{b}$ (S20)

G_{jki}: $\underline{b}$ is given to $\underline{c}$ by $\underline{a}$ (S21)

We note further that the reflection of G_{kji} results in a sentence that is identical with (S18), and that the rotation of G_{kij} gives G_{jki}, which rotated in turn gives $G_{ijk'}$ again resulting in a sentence indistinguishable from sentence (S18).

We need only assure ourselves that the order of the elements is identical before we claim that triadas are equal in general. In our case, since i, j, k is the order in all the first and last triadas in the series of Fig. 10a-d we may indeed remove the subscripts from G_{ijk}, G_{ijk}, and G_{ijk}, concluding in general that

$$\breve{G} = G \text{ and that } \hat{G} = G \qquad [\text{ E6 }]\ [\text{ E7 }]$$

Interfaces and Shifts

When a triada is complex, we note that one or more "arms" of a triada may cross through the envelope of a triada containing one or more internal triadas. Thus, in Fig. 11 triada G' contains triada G. The order of the "arms" of G_{ijk} is the same order as the arms of G'_{ijk}. Only when such sameness is observed in complex triadas may the complex triadas be considered *well formed*. We thus need to add the sixth requirement to the lists above: for a complex structure, any "arm" crossing an "envelope" must have the same relative *number* (i.e., first, second, or third) for the triada which is the envelope *and* for any triada inside, to which it leads. To move from an "external" structure or triada to an "internal" triada or other structure, we shift our focus from the outside of the envelope to the inside. Such a shift of focus, in either direction, we shah term simply "shift". To distinguish moving out or moving in through such envelopes, we may use the terms shift out (or shift up) for the former case and the terms shift in (or shift down) for the latter.

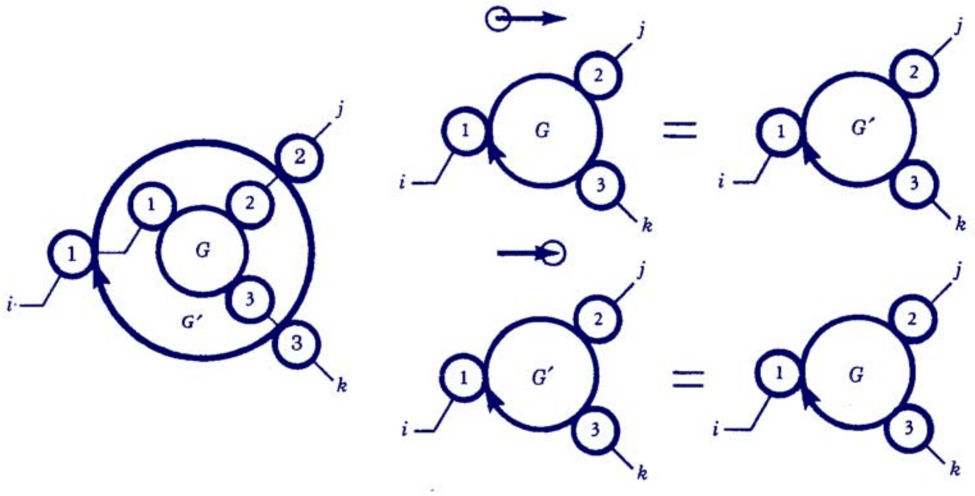

FIG. 11.

Unary shifts are perhaps trivial, but we shall have need to assure ourselves that complex triadas are well-formed before assuming that we can freely shift.

For graphic convenience, we use an arrow pointing *out* from a circle to indicate a shift *outward* and an arrow pointing *into* a circle to indicate a shift i*nward*.

Thus, if G and G' of Fig. 11 are the same semantically, then

$$\overset{\circ\!\!\rightarrow}{G}_{ijk} = G'_{ijk} \quad \text{and} \quad G'_{ijk} = \overset{\rightarrow\!\circ}{G}_{ijk} \qquad [\text{E8, E9}]$$

or, since the subscripts are identical,

$$\overset{\circ\!\rightarrow}{G} = G' \quad \text{and} \quad G' = \overset{\rightarrow\!\circ}{G} \qquad [E8, E9]$$

Illustrative sentences are

G: __gives__to__ (S22)

G': __is identical with one who gives one identical with __to one identical with__ (S23)

Binary Operations (or Nonrelative Operations)

NonrelativeProduct. * The nonrelative product of two triadas is a triada obtained after joining the two original triadas with the logical connective "and," and making the subscripts in both triadas the same. For example, let C_{ijk} mean

"__gives__ to__" (S24)

and L_{ijk} mean

"__lies in between__and__" (S25)

The nonrelative product, represented by G_{ijk}. L_{ijk} is the triada

"__gives__to__, and the first lies in between the second and the third." (S26)

It follows that

$$Gijk \bullet Lijk = L_{ijk} \bullet G_{ijk} \qquad [E12]$$

Graphically, nonrelative products are shown as in Fig. 12.

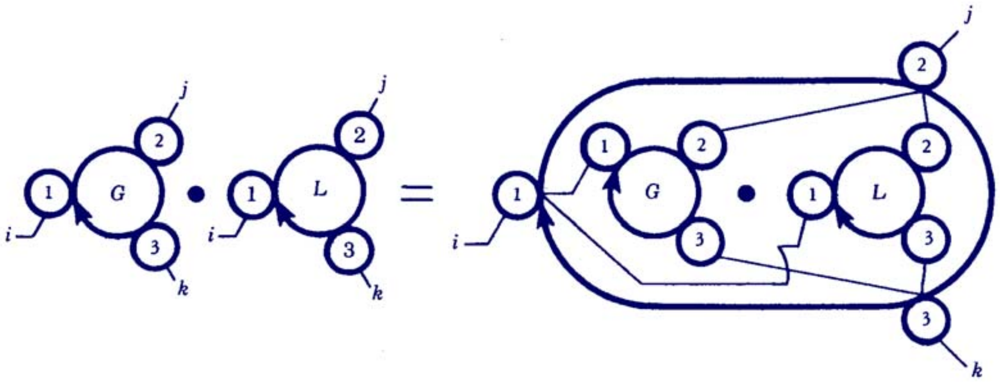

FIG. 12.

*end of **

Whether G_{ijk} or L_{ijk} is the left-most figure inside the oblong does not matter. As long as the first arm of G and the first arm of L both are also the first arm of the triada resulting from their product, and similarly for their second and third arms, then any shuffling around that preserves these relations does not affect the result. Thus, all the figures in Fig. 13 are equivalent and equal. We note that the joining at the border of the outer oblong indicates shared elements, and that the dot in the middle of the figure indicates a prodùct (the result of a logical "and" operation).

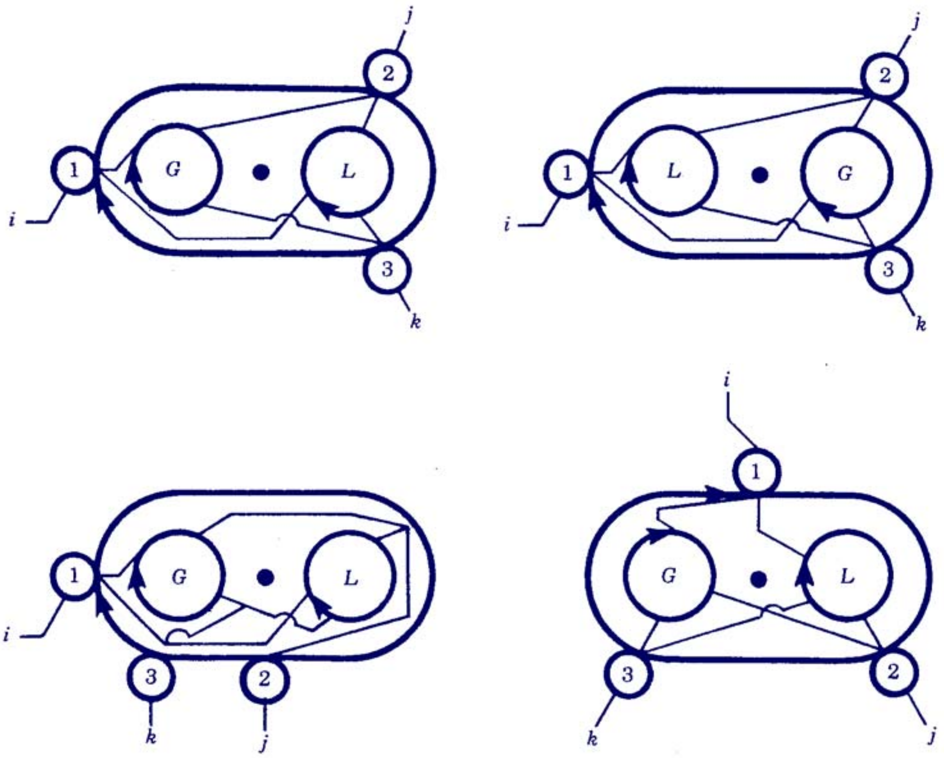

FIG. 13.

Nonrelative Sum. * The nonrelative sum of two triadas is the triada obtained after joining the two original triadas with the logical connective "or" (inclusive or) and making the subscripts in both triadas the same. For example, the nonrelative sum of C_{ijk} and L_{ijk} is the triada

"__gives__to__, or the first lies in between the second and the third." (S27)

We represent it by $G_{ijk} + L_{ijk}$. It is clear that

$$G_{ijk} + L_{ijk} = L_{ijk} + G_{ijk}. \quad \text{[E13]}$$

Graphically, nonrelative products and nonrelative sums differ only in products being marked with a "•" (and) and sums with a "+" (or). Figure 14 shows the nonrelative sum $G_{ijk} + L_{ijk}$.

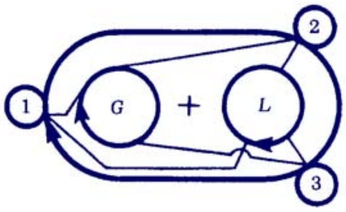

FIG. 14.

*end of **

If we now choose to call the binary complex triada

"__gives__to__, and the first lies in between the second and the third." (S28)

by some other name, such as H, then we note that

$$\overset{\rightarrow\!\circ}{H}_{ijk} = (G_{ijk} \bullet L_{ijk}) \quad \text{[E14]}$$

and that

$$\overset{\circ\!\longrightarrow}{(G_{ijk} \bullet L_{ijk})} = H_{ijk} \quad \text{[E15]}$$

Graphically, this is indicated in Fig. 15.

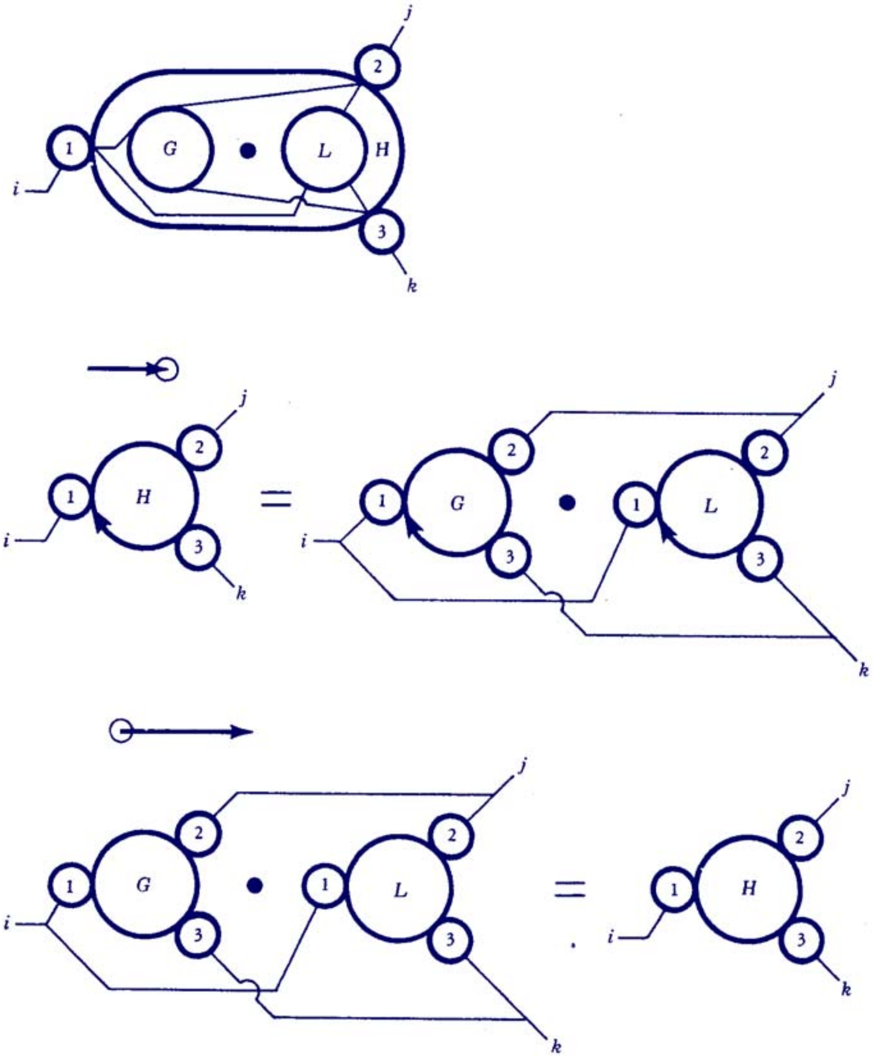

FIG. 15.

vordenker
ISSN 1619-9324

How to cite:
Christopher R. Longyear: Further Towards a Triadic Calculus – part 1, in: www.vordenker.de (Edition: Winter 2008/09), J. Paul (Ed.), URL: < http://www.vordenker.de/ggphilosophy/longyear-part_1.pdf > — originally published in: Journal of Cybernetics, 1972, Vol. 2, No.1, pp. 50-65.