Cybernetics And Human Knowing, Vol.5, no.3, 1998, pp. 63–70

Virtual Logic — Symbolic Logic and The Calculus of Indications

by Louis H. Kauffman

This is the sixth column in this series on "Virtual Logic". In this column we shall give a short exposition of how symbolic logic is illuminated by the calculus of indications. Columns four and five began an introduction to the calculus of indications. Nevertheless, we shall be self-contained here and recall this construction in section 1.

I. Finding the Calculus of Indications

We take as given the idea of a distinction.
Let O denote one side of the distinction (the outside).
Let I denote the other side of the distinction (the inside).

Let < S > denote the side opposite to S.
Thus
<Outside>= Inside
<Inside>= Outside
which we abbreviate to
< O > = I and < I > = O.

There is no necessity for the symbol I. Let the unmarked state denote the I. Then the crossing from the inside to the outside becomes the equation <   > = O, and the crossing from the outside to the inside becomes the equation <O> =   .

Our equations have simplified:
<   > = O
< O > =   .

The equation <   > = O suggests that we identify the name of the outside with the mark (bracket) itself. Call the state indicated by the empty bracket the marked state. We then have only one equation:

<< >> =   .

"The value of a crossing made again is not the value of the crossing."

Crossing from the marked state yields the unmarked state. The brackets now hold the double function of noun and verb, name and operator. The empty bracket as verb is the action of crossing from the unmarked state. The action of crossing from the unmarked state indicates the marked state. Action and name fit together.

Now let the mark itself be the distinction that is being described.

The mark < > is no longer distinct from the original distinction. The mark is the name of the original distinction. Hence, the mark is its own name. We have found a sign that stands for itself.

There is another equation:

< > < > = < >

"The value of a call made again is the value of the call."

In this equation, the two adjacent marks can be regarded as naming each other, but since the mark names itself, the two become one.

Without letting go of the distinction altogether this marks the bottom of the descent. What remains is to identify all other apparent distinctions (insofar as they are distinctions) with the mark itself. Finally, we identify distinction with the form of distinction.

For the cybernetic context this language of distinctions is significant in that the notation itself becomes a system that embodies its own self-description. Notice that we said that we "found" a sign (the mark) that stands for itself. One might be tempted to say that we "constructed" a sign that stands for itself. However, the use of the word "constructed" suggests the "making" of a formal system in which the sign stands for itself. Such constructions are done - for example by Godel numbering and related devices in formal systems that contain the structure of arithmetic. However, here we have "deconstructed" (i.e. made formally simpler) an originally very simple system (<O>=I, <I>=O) and "found" in the process of deconstruction a sign that stands for itself. In a similar way, we as self-observing systems find that the sign "I" (in referring to the self) stands for itself in the context of self-reference ("I am the one who says I."). This occurs through simplification at the level of self-description that identifies "the observer of observing the self" with the self. Self-reference arises ("is found") through the acceptance of such identifications.

Of course, we could also say that the very small formal system generated by the two equations

<<>> =
<><> = <>

is a construction of a formal context in which the sign <> stands for itself. Viewed this way the "construction" has the appearance of a "leap" rather than a carefully considered process of building. This happens even though the sign arose from a carefully considered process of un-building! Here we are confronted with the issues of invention and discovery in a context that sheds new light on the matter. By “uncovering” the calculus of indications we have the opportunity to “find” it and to realize that it has always been within the structure of our discourse. The process of discovery is a process of seeing the presence and universality of a structure. The process of invention is inherently tactical, depending upon choices available in the moment. Nevertheless, an invention can be seen as a discovery (always!) and a discovery can sometimes be seen as an invention.

II. The Calculus of Indications

The two equations

<<>> =
<><> = <>

become the basis for a calculus of indications where we evaluate more complex expressions such as <<<<><><>>>> by using these equations to make forms simple or more complex. This section will show how this calculus works and the next will show how it applies to logic.

Consider a space that can be divided, giving rise to a distinction in that space. For example, a circle makes a distinction in the plane and a pair of left and right brackets makes a distinction in a line. In the case of the pair of brackets, the structure of the brackets themselves tells the observer whether she is inside or outside the disinction so formed.

(You look at the pair of brackets and if you find that they are not pointed sharply at you, then you stand inside the distinction formed by the brackets.)

Outside < Inside > Outside

Nota Bene. The distinction made by a pair of brackets in the line divides that space into an inside and an outside *and* it divides the outside into a left part and a right part. Thus, strictly speaking, the pair of brackets creates three spaces and two interlocked distinctions. This is the actual complexity of the typographical distinctions that we utilize in reading and writing the English language and in mathematics. In particular, it allows us to distinguish AB from BA since in one case, A is to the left of B, and in the other case, A is to the right of B.

It is convenient to use brackets and to consider distinctions that are produced in a linear space. The reader may enjoy imagining versions of what we now do in higher dimensional spaces such as the plane, other surfaces, three dimensional space, four dimensional space and other spaces as well. This theme of distinctions relative to a containing space will be the subject of other columns in this series.

If we draw our distinctions as circles in the plane, then each circle makes a distinction between inside and outside, but there is no longer any distinction between right and left. In the plane it is not possible to say in general that one circle is to the right or left of another. This means that the field of distinctions drawn in a plane space is simpler than the always- given field of typographical distinctions in the line. Nevertheless, we shall here stick to the work in the line.

In any space divided by a mark of distinction (such as the brackets), we can make further distinctions by drawing new marks of distinction either in the inside of the given mark or outside that mark The possibilities that result from one more distinction are shown below.

<> <>- each is outside the other.
<<>> - one is in inside the other.

If we make a third distinction in one of the divisions already created by two distinctions we obtain the following possibilites.

<> <> <>- each is outside the other two.
<<>> <>- one is inside another on the left.
<> <<>> - one is inside another on the right.
<<><>>- two are outside each other and inside a third.
<<<>>>- one is inside another that is inside another.

Systems of distinctions in the line that are produced in this way are called *expressions*. An expression is any well-formed (in the sense of parentheses) pattern of brackets. The definition of well-formedness is encapsulated in the (recursive) rules below.

0. The empty expression is well-formed.
1. <> is well-formed.
2. If A is well-formed then <A> is well-formed.
3. If A and B are well-formed, then AB is well-formed.
4. All well-formed expressions are obtained in this way from a finite number of brackets.

Thus <<<>>> and <<>><<>> are well-formed. The brackets come in pairs. For each pair of brackets it can be determined whether they are outside or inside any other pair of brackets.

The pattern <<<>> is not well-formed, nor is >><<.

Lemma. Every finite well-formed expression can be reduced by a sequence of applications of calling (<><> = <>) and crossing (<<>> =) to either a marked state or an unmarked state.

Proof. Just note that by delving into the inside of an expression you can always find an instance of calling (<><>) or crossing (<<>>). The instance of calling can be reduced from two adjacent marks to one mark. The instance of crossing can be reduced from two included marks to none. Continuing this process eventually reduces the expression to one mark or none.

For example <<<>>> = <   > = <>.
For example <<>><<>><<>> = <<>><<>> = <<>> =   .
For example <<>><<<<>>>><> = <<<<>>>><> = << >><>=<>.

Note that in these examples we look for an isolated appearance of calling (<><>) or crossing (<<>>) and then apply either condensation or cancellation. It is not hard to see that there always will be an isolated appearance of calling or of crossing. This is what makes the Lemma work! If we had allowed infinite expressions then this lemma would not hold. For example, if

J = <<<<<...>>>>>

is an infinite nest of crossings, then there is no available simplification and we see that J is not equivalent to either the marked state or to the unmarked state. Such infinite expressions are quite significant.

J = <J> in the sense that an exact copy of J sits inside itself. Note also that we can say of J that it is a structure that reenters its own indicational space (It sits inside itself.). Here we enter into the realm of “constructions” for self-reference. But once again this idea goes beyond the usual meaning of construction, for an infinity is not a finite construction. If we insist on constructing J step by step then it will take an infinite number of steps. In the usual sense of the word, J cannot be constructed. Nevertheless, we have “found” J in the world of mathematical imagination.

In a previous column we constructed gadgets like J by an operational process. We called the operator G with the property Gx = < xx > for any x, the “Gremlin”. We allowed G to operate on itself and found GG = < GG >. Thus GG reenters its own indicational space just as does J, but GG does not require an excursion to infinity! Nevertheless, the existence of an operator G and the assumption that G can be applied to itself constitute a construction of a formal system (the lambda calculus) that allows self-reference to occur. Self-reference is infinity in finite quise.

Theorem. There is no finite sequence of steps, via calling and crossing, connecting the marked state to the unmarked state.

Proof. The interested reader should consult [1]or [2]. The proof is worth discussing and we shall do that here, leaving out some details. First we construct a new and very simple formal system of values denoted by M (marked) and U (unmarked). We assume the equations

MM= M
MU = M
UM = M
UU= U.

(If you let U be actually unmarked and M be the marked state, then these equations express the laws of calling and the condition of identity.)

Now take a finite expression E and label it according to the following rules:

0. An empty expression is labelled U.
1. Label each deepest space U.
2. If a space S is labelled U and there is a cross around it, label the space exterior to the cross by M .
3. If a space S is labelled M and there is a cross around it, label the space exterior to the cross by U .
4. If a space has more than one label then find the value of that space (i.e. its label) by using the four equations above.
5. Continue from the deepest spaces in the expression upwards until a value for the entire expression is obtained as a label for its outermost space.
6. Define the value V(E) to be M or U according to the labelling of the outermost space in E.

It is not hard to see that this labelling process assigns a unique value to every finite expression E. Note that V( ) = U by the above and that V(< >) = V (<U> M) = M. Thus the unmarked state has value U and the marked state has value M.

Furthermore, it is easy to see that if an expression E' is obtained from E by an application of calling or crossing, then V(E') =V(E). This part follows from the properties of labelling calling and crossing:

In labelling the form of calling we find <U>M <U>M hence the outer space of < > < > is labelled by M as is the outer space of < >. This means that the labelling of an expression containing < > < > will be the same as the labelling of an expression containg < > in its place. The same remarks apply to << >> with the labelling <<U>M>U.

Now suppose that there were a sequence of callings and crossings that transformed the unmarked state into the marked state. Then it would follow that V( ) = V(<>) hence that U=M. This is a contradiction. Since we started with the distinction between U and M, we conclude that there is no way to transform the unmarked state to the marked state by a sequence of calling and crossing. This completes the proof.

Example

For example, if we start with the expression E = < <> <> <<>> > then we have

1. < <U> <U> <<U>> >
2. < <U>M <U>M < <U>M > >
3. < <U>M <U>M < <U>M >U >

In the labelling so far the space in the expression that is one crossing away from the outside has labels M,M,U hence it is labelled MMU = MU = M. Therefore, the ouside is labelled U and we have the final labelling < <U>M <U>M < <U>M >U >U. In fact E is equivalent to the unmarked state, as is easy to see by directly applying the calculus of indications.

Remark

Note how this proof depends upon the idea of a distinction. If U can be distinguished from M (as letters!) then the marked and unmarked states in the calculus of indications are distinct in that calculus. Such a proof almost assumes what it is intending to prove!

In fact, in order to construct the proof, we used a portion of the calculus of indications (in the form of the calculus of U and M) in order to prove a result about the whole. In that portion it is indubitable that a product such as UUMMMUMUMUMMU has a unique value since that value is either M if M is present and U if M is absent in the product. Thus we base the properties of our more complex calculus of indications on the simple abilities of the observer in the formal system to discriminate between two letters and to detect the presence or the absence of a symbol in a finite list of symbols. These are basically the properties of an observer that are required in the operation of idealisations of computing such as Turing machines. It is important to continue this discussion of the nature of observing systems and to discuss possibly more subtle modes of observation that might be incorporated in generalisatons of the Turing paradigm.

III. Logic and Symbolic Logic

We have seen that the calculus of indications can be used to form the basis for systems that embody a binary distinction. In particular there is the possibility that this calculus will apply to logic. This is the case, and we will here show how to translate elementary symbolic logic into the calculus of indications.

First it is worth pointing out some general patterns in the calculus of indications:

0. ab = ba for any expressions a and b.
1. <<a>> = a for any expression a.
2. a<> = <> for any expression a.
3. p <p> = <> for any expression p.

At this stage, the reader should be able to verify these statements for herself.

In logic we have the operatons *and*, *or*, *not* and *implies* and the values *true* and *false*. These expressions can be modelled in the calculus of indications as follows:

(we use a —> b for "a implies b")
True= <>
False =
a OR b= ab
a AND b= <<a><b>>
a—> b = <a>b .

Let's see how this works out in an example:

a Implies b = a—> b = <a>b = b<a> = <<b>><a>
= Not(b) —> Not(a)
= Not(B) Implies Not(a).

It is well known that "a Implies b" and the contrapositive "Not(b) Implies Not(a)" are logically equivalent. The model for logic that is given by the calculus of indications shows how this equivalence comes about by giving a form, <a>b, that the two propositional structures have in common. The calculus of indications provides a structural domain that is deeper than ordinary symbolic logic and from which the structure of that logic can be seen with clarity. In the deeper domain a multiplicity of logical forms share explicit common features and admit transformations among themselves.

It is natural to ask "what is the source of reason" or "what are the laws of thought". The idea of a distinction and the possibility of following this idea in the complexity of its representations provides a way to investigate these questions and to build intuition leading toward their answers.

It is apparent even from this tidbit of logic in relation to the calculus of indications that the patterns of ordinary symbolic logic are superficial in relation to this formalism and in respect of our own ability to conceive reason.

The next column will be devoted to a deeper foray into logic in relation to form. We close this column with an exercise for the reader:

Transcribe
" ((a Implies b) and (b Implies c)) Implies (a Implies c) " into the calculus of indications and verify that it always has the marked value in that calculus.

References

1. G. Spencer-Brown, Laws of Form, George Allen and Unwin Pub. London (1969).
2. L. Kauffman, Arithmetic in the Form, Cybernetics and Systems Vol.26 (1995), 1-57.