The Robbins Problem - Computer Proofs and Human Proofs

by Louis H. Kauffman

I. Introduction

This paper is devoted to a discussion of the relation between computer proof and human proof. It is a discussion of the relationship of persons and machines. In Section 2 we discuss these issues in general and specifically in regard to the recent solution of the Robbins Problem via a proof generated by computer. Section 3 gives the mathematical background for the Robbins problem.

The Robbins problem was a longstanding open problem about axioms for Boolean algebra. One point of this paper is to show that the proof of the Robbins conjecture, generated by a computer, can be filled in and understood by human beings. We accomplish this aim in the present paper by presenting a notational reformulation of Boolean algebra and the Robbins Problem in Sections 4 and 5. This reformulation is called "box notation". Section 6 discusses cybernetic and semiotic issues. Section 7 is a formal presentation of a proof, in box notation, that Robbins algebras are Boolean.

It should be mentioned that the notational reformulation given herein is most effective for a person when he or she actually uses the notation, writing out and making formulas and diagrams by hand. This activity allows the combination of hand and eye to correlate the patterns inherent in the formulas. Sections 5 and 7 should be regarded as guides and injunctions for such personal work. As far as this author knows, such work is indispensable for reaching understanding. Each person who encounters a piece of mathematics must hold a conversation with it. This conversation almost always includes the development of individual language and notation to enable that person to grasp the mathematics. This paper is an invitation to enter that domain of linguistic creativity through which mathematical understanding can arise.

The issue developed here is of cybernetic, linguistic and semiotic interest. The ideas behind our notational reformulation of the Robbins Problem go back to inventions/discoveries of Charles Sanders Peirce [Peirce,C.S. (1898/1992)], [Ketner, Kenneth Laine (1990)] and G. Spencer-Brown [Spencer-Brown,George (1969)]. The specific reformulation used here had already been investigated by this author in [Kauffman, L.H. (1990)]. It is our contention that mathematics can behave non-trivially under change of notation. Change of notation is change of language. In the present case the change of language afforded by an appropriate change of notation makes a mathematical domain accessible to human beings that has heretofore been only accessible to computers.

As computers accept more and more work in the domain of demonstrating mathematical theorems, it will become increasingly necessary to discover new languages that intermediate between the human domain of understanding and the computational domain of symbolic manipulation.

The present paper is an expanded version of a column by the author that appeared in the journal "Cybernetics and Human Knowing" [Kauffman,L.H. (1997b)].

II. Proofs

Can a computer discover the proof of a theorem in mathematics? Some would say that this is certainly possible, since the computer has only to find the right steps. After all, proving a theorem is like solving a problem in chess, and we are all aware that computers can play quite a good game of chess. On the other hand, I say that a proof is not a proof until a person is convinced by it. In fact a mathematical proof is exactly an argument that is completely convincing to a mathematician! In this sense, a computer does not , can not produce a proof. The computer is not convinced of anything. The computer is set to search for something - a certain series of steps according to some rules. The computer can indicate the state desired by its programmers. It does not know the proof. It only finds the steps.

It is a human judgement that propels the result of the computer's search into a statement that the computer has "found a proof". Indeed the computer has helped us and it has found something that we can examine. If we judge that to be a proof, then it is a proof (for us). How do we judge?

This last question is the real question about proof. How do we know when we have a proof? We need to look at this question from many angles. One view on it will come forth as we look at a specific problem.

In 1996 a computer program at Argonne National Labs in Argonne, Illinois found a proof for a famous unsolved problem in Boolean algebra, known as the *Robbins Problem*. (Incidentally, Robbins is the name of a mathematician Herbert Robbins, who introduced the problem. We shall not use an apostrophe on the "s" in Robbins when we say the Robbins problem.) The problem can be elegantly stated, and we shall state it. But first a story:

The computer proof of the Robbins Problem was completed on October 10,1996 by a program named "EQP". The proof was successfully verified by another program named "Otter." It was already known that it would be sufficient for the computer to deduce a certain equational pattern from the Robbins Axioms. EQP succeeded in locating this form. The computer output of the final demonstration is not long, but it contains many formulas that are exceedingly difficult for a human being to read. For example, one line of the proof reads:

~(~(~(~(~(x) + x) + ~(~(x) + x) + x + x + x + x) + ~(~(~(x) + x) + x + x + x) + x) + x) = ~(~(~(x) + x) + ~(~(x) + x) + x + x + x + x).

This run on the Robbins problem was initiated by William McCune, a member of a group of researchers in automated theorem proving at the Argonne Labs. The head of the group is Larry Wos. Credit for the initiative goes to Dr. McCune and to the team who set the computer on the track of the problem.

It might seem that the demonstration of the Robbins Problem by the program EQP is out of human reach. This is not so! It is possible to change the representation of the problem in such a way that a human can check the proof and even appreciate the form of reasoning used by EQP. In fact, this mode of representation creates an arena in which one can continue the investigation either by hand, or with the aid of the computer.

EQP's human guides were asked how they knew the demonstration was correct. Their very good answer is that, along with checking it by hand, they set another program (Otter) to checking and rederiving EQP's results and that it is confirmed. I was very curious about this and found their web page [McCune,W. (1996)]. There, reproduced, were the output lists of EQP and Otter. Could a human being follow the proof? I decided to give it a try. The first part of this task, entailed a simplification and translation of the notation used by the computer. It is very difficult for a human being to keep track of nested parentheses, but not too hard to look at patterns of nested boxes. Accordingly I translated the steps in the proof into a nested box notation. I had previously worked on the Robbins problem using this notation [Kauffman, L.H. (1990)].

The translation of EQP's proof was not trivial. I had to find derivations of the statements in the EQP proof. These were not given, just the sequence of propositions leading to the final result. It *is* possible to humanly comprehend the EQP proof! We will look at these details in later sections of this paper.

It is quite fascinating to contemplate the present situation. EQP produced a proof that could have been inaccessible to human beings but is in fact understandable to us with a little effort in communication. EQP's proof is much more than a calculation. The proof depends upon a successful search among a realm of possibilities and the skillful application of pattern recognition and the application of axioms. This is very close to the work of a human mathematician. EQP, being able to handle many possibilities and great depths of parentheses has an advantage over her human colleagues. I understood EQP's proof with an enjoyment that was very much the same as the enjoyment that I get from a proof produced by a human being. Understanding a proof is like listening to a piece of music - or like improvising a piece of music from an incomplete score.

There will be successors to EQP. Computers will prove theorems with proofs that are very long and complex. These proofs will need translations into language that human beings can understand. Here is the beginning of a new field of mathematics in the interface between logic, communication and the power to reason and understand. The machines will help us move forward into new and powerful ways to explore mathematical terrain.

III. Background for the Robbins Problem

In order to understand the Robbins problem it is necessary to know about *Boolean algebra*. Boolean algebra was discovered/invented by George Boole [Boole, George (1847)] as an *algebra for logic.* Boole had the happy idea that one could use the notation of ordinary algebra, but re-interpret it to express the meanings of symbolic logic while retaining the calculating power of the algebra. I will describe this idea of Boole in modern notation.

Boolean algebra is the algebra that describes the simple properties of a single distinction. This represents the simplicity of on/off, outside/inside, true/false, present/absent. In the case of true and false, Boolean logic assumes a simplicity that idealizes language and experience. I shall recall Boolean algebra by first recalling its interpretation for the logic of true and false.

In Boolean algebra ~A means "not A" while A*B denotes "A or B" and A#B denotes "A and B". With this interpretation in mind, many rules of logic become identities in the algebra. For example,

$$A \# B = \sim((\sim A) * (\sim B)).$$

The statements "A and B." and "It is not the case that not A or not B." are logically equivalent.

This rule is called "DeMorgan's Law" (after the logician Augustus DeMorgan 1806-1871). It makes explicit the fact that we can express "and" in terms of the operations "not" and "or". Incidentally, we use here the *inclusive or* that means "A, B, or both". In Boole's original work he used the *exclusive or* that means "A or B but not both".

There are many rules in Boolean algebra. One wants an axiom system of small size from which all the true equations can be deduced. There are many such systems of axioms.
Here is a standard version:

Standard Axioms for a Boolean Algebra

0. The Boolean algebra B is a set that is endowed with a binary operation denoted "*" and a unary operation denoted "~". The set is closed under these operations. That is, given a and b in B, then a*b is in B and ~a is in B.

1. The operation + is commutative and associative. That is a*b = b*a for all a and b in B, and (a*b)*c = a*(b*c) for all a,b,c in B.

2. There is a special element called "zero" and denoted by the symbol "1" such that 1*a = a for all a in B. ~1 is denoted by the symbol "0".

3. ~~a = a for all a in B.

4. ~(a * ~a) = 1 for all a in B.

5 a * ~(~b * ~c) = ~(~(a*b) * ~(a*c)) for all a,b,c in B.

From these axioms one can deduce all the other facts that are true in Boolean algebra.

Commutativity and associativity are part of the simplification of ordinary language to the mathematics of this model. We say "scotch and soda" and we do not say "soda and scotch". In this sense ordinary language is not commutative. In the ideal of pure logic, we imagine that it is possible to list a set of propositions and to disregard their order. In this sense, the axioms of commutativity and associativity are like the assumption of frictionless surfaces and ideal gases in physics.

For the purpose of logic, 1 means "false" and 0 means "true". The operation a*b is interpreted as "a or b". Thus we have

$$0*1 = 1*0 = 0,$$
$$0*0=0,$$
$$1*1=1\ ,$$

in the context of this interpretation since "a or b" is true exactly when either a is true or b is true or both a and b are true.

Note that in this Boolean algebra we have chosen the rules for combination of 0 and 1 with respect to * so that they look like the usual rules for multiplication of 0 and 1 in ordinary arithmetic. In some treatments of Boolean algebra, one uses the symbol "+" as "or". In this case the roles of 0 and 1 are reversed. This is a matter of convention since 0 and 1 in Boolean algebra simply stand for two distinct states whose interpretation depends upon the context. See section 6 for more discussion of this point.

Axiom 3 says the familiar law of double negation: "Not not a is equivalent to a."

Axiom 4 says ~(~a *a) = 1. Thus Axiom 4 says that "It is false that it is not the case that either not a or a." In English this translates to the algebraic equivalent (~a)*a = 0: "It is the case that either a or not a." Thus Axiom 4 is a statement that says that for every element in the Boolean algebra, either a is true or ~a is true.
Again, we are in the idealization that assumes that there is no ambiguity, no statements beyond true and false in the system B. (It is beyond the scope of this paper to discuss systems of multiple valued logic or of fuzzy or modal logic where ambiguity is partially formalized. Such systems are also in the natural evolution started by Boole.)

Finally, Axiom 5 is better understood if we use the definition of "and" (as described above): a#b = ~((~a) *(~b)). Then Axiom 5 reads

a *(b#c) = (a*b)#(a*c):

The statement " it is the case that a or it is the case that b and c." is equivalent to the statement
"it is the case that a or b and it is the case that a or c."

We tend to understand these statements of logic, once they are stated in English, by comparing them with images of familiar situations. The algebra knows nothing but the rules that have been given to it. In this way the algebra acquires a life of its own. For example, *it is possible to prove Axiom 3 by using the other axioms.* In order to do this you must take a very formal attitude. The well-known interpretations of the algebra must not be used. The only rules that one can rely upon are the axioms themselves and the "usual" rules for substitution and replacement that apply in all algebraic situations.

Here is the demonstration of Axiom 3 from the other Axioms:

1. ~~a

2. = ~~a * 1 (Axiom 2)

3. = ~~a * ~(~(a) * ~(~a)) (Axiom 4)

4. = ~(~(~~a *a) * ~(~~a * ~a)) (Axiom 5)

5. = ~(~(a * ~~a) * ~(~a * ~~a)) (Axiom 1)

6. = ~(~(a* ~~a) * 1) (Axiom 2)

7. = ~(~(a* ~~a) * ~(a * ~a)) (Axiom 2)

8. = a * ~(~~a * ~a) (Axiom 5)

9. = a * ~(~a * ~~a) (Axiom 1)

10. = a * 1 (Axiom 4)

11. = a (Axiom 2)

Note that the proof as list of steps and reasons leaves out any discussion of strategy in relation to making these particular steps. We also use the convention that the operation of negation (~) takes precendence over the operation combination (*).

In making a proof we may believe that we have shown all the steps and all the reasons. There is still work for the reader. He or she has to be able to *see* the reason for taking a given step! In this sense every proof, *as a text* , is full of holes. The holes are filled in by the person who reads and comprehends the text.

We shall return to this proof in the next section .

How far can you simplify the axioms for a Boolean algebra? Huntington made a discovery in 1933 [Huntington,E.V. (1933)]. He found an axiom system for Boolean algebra that uses only one axiom beyond commutativity and associativity.

Huntington's Axioms for Boolean Algebra

0. The Boolean algebra B is a set that is endowed with a binary operation denoted "*" and a unary operation denoted "~". The set is closed under these operations. That is, given a and b in B, then a*b is in B and ~a is in B.

1. The operation + is commutative and associative. That is a*b = b*a for all a and b in B, and (a*b)*c = a*(b*c) for all a,b,c in B.

2. For all a and b in the set B, a = ~(~a * b) * ~(~a * ~b).

In these axioms it is understood that the operation ~ takes precedence over the operation *. Thus ~a*~b = (~a)*(~b).

This discovery of Huntington means that it is possible to get the whole structure of Boolean algebra from just one special axiom (plus commutativity, associativity and the conventions of one binary and one unary operation). Now things are beginning to get interesting. It is quite a puzzle to derive everything from Huntington's special axioms. For example, the first thing that Huntington proves is that ~(a*~a) = ~(b *~b) for any a and b in B. After he shows this, he can *define* 1 by the equation 1= ~(a*~a). He then has to work quite hard to show that 1 behaves like 1! (For the reader who consults the paper by Huntington: Huntington uses the notation "+" where we use "*" so that his notations for 0 and 1 are reversed. This is simply a notational convention as we have discussed above. It does not affect the mathematical content of the work.)

In the 1930's Herbert Robbins, a Professor of Mathematics at Rutgers University in New Brunswick, New Jersey, asked the seemingly innocent question: What if we replace Huntington's Axiom 2. by the following

Robbins Axiom 2. For all a and b in the set B,
a = ~(~(a*b) + ~(a*~b)).

Does this again yield precisely Boolean algebra?! That question is the Robbins Problem.

It may seem an easy matter to decide this problem. If you think that this is so, please give it a try. Everyone who has made the attempt so far has found great difficulty. The difficulty appears at first to be notational. The big parenthesis surrounding the whole expression *~(a*b) *~(a*~b)* makes it difficult to work with. Then difficulty becomes frustration and frustration despair as the problem recedes from view and the mathematician drowns in a sea of unruly paper. On top of this, the rewards seem very slim. Boolean algebra is well understood.

Why should we worry about a puzzle like this?

Mathematics throws strange and magic puzzles up to its practitioners. These puzzles sometimes become whole fields of study. It is hard to predict. It is rare to find a challenge that engages the mind just so. Eventually such puzzles acquire enough associations so that one can explain why they are important or interesting.

IV. Reformulation - Parentheses and Brackets

In this section I will explain a notational reformulation that applies to both Boolean and Robbins algebra. The idea for this reformulation is inherent in the work of C. S. Pierce [Peirce,C.S. (1898/1992)], [Ketner, Kenneth Laine (1990)] and in the work of G. Spencer-Brown [Spencer-Brown,George (1969)].

In this notation we will let **<a>** stand for **~(a)** in the standard notation. Sharp brackets around an expression denotes its negation. We shall write **ab** where before we wrote **a*b**. The result of this condensation of notation is that one writes fewer symbols, and the sharp brackets replace parentheses. In the next section we will make a further step by adopting a box notation that joins the two ends of a bracket (left end < and right end >) to make a single form for each bracketing.

Here is the rewrite of the axioms for Boolean Algebra in the bracket notation:

Standard Axioms for a Boolean Algebra in Bracket Notation

0. The Boolean bracket algebra B is a set that is endowed with a binary operation indicated by ab for a and b in B, and a unary operation denoted <a> for a in B. The set is closed under these operations. That is, given a and b in B, then ab is in B and <a> is in B.

1. The binary operation is commutative and associative. That is ab = ba for all a and b in B, and (ab)c = a(bc) for all a,b,c in B.

2. There is a special element called "unity" and denoted by the symbol "1" such that 1a = a for all a in B. <1> is denoted by the symbol "0".

3. <<a>> = a for all a in B.

4. $\langle a \langle a \rangle \rangle = 1$ **for all a in B.**

5. $a \langle \langle b \rangle \langle c \rangle \rangle = \langle \langle ab \rangle \langle ab \rangle \rangle$ **for all a,b,c in B.**

One of the advantages of the bracket notation for Boolean algebra is that we can usually avoid using any parentheses other than the brackets themselves. This simplifies the notation for transcribing proofs and demonstrations.

For example, we can now prove that $\langle \langle A \rangle \rangle = A$ is a consequence of the other axioms:

$\langle \langle A \rangle \rangle = \langle \langle A \rangle \rangle 1$ **by Axiom 2**

$= \langle \langle A \rangle \rangle \langle \langle A \rangle \langle \langle A \rangle \rangle \rangle$ **by Axiom 4**

$= \langle \langle {\scriptstyle \langle \langle A \rangle \rangle} A \rangle \langle {\scriptstyle \langle \langle A \rangle \rangle} \langle A \rangle \rangle \rangle$ **by Axiom 5**

$= \langle \langle {\scriptstyle \langle \langle A \rangle \rangle} A \rangle 1 \rangle$ **by Axiom 4**

$= \langle \langle \langle \langle A \rangle \rangle A \rangle \langle \langle A \rangle A \rangle \rangle \rangle$ **by Axiom 5**

$= A \langle \langle \langle \langle A \rangle \rangle \rangle \langle \langle A \rangle \rangle \rangle$ **by Axioms 5 and 1**

$= A\ 1$ **by Axiom 4**

$= A$ **by Axiom 2.**

The essential theme of this proof is that everything goes back to the two basic axioms:

4. $\langle a \langle a \rangle \rangle = 1$ **and**

5. $a \langle \langle b \rangle \langle c \rangle \rangle = \langle \langle ab \rangle \langle ac \rangle \rangle$ **.**

Axiom 4 allows one to create from an unmarked state (i. e. from **1**) and to take certain forms back to that unmarked state. I refer to **1** as an unmarked state because it has the neutral property that **A1 = 1A = A** for any **A**. Axiom 5 permits an interaction between forms and it includes the possibility of duplicating a form (as a is duplicated in going from the left hand side of 5. to the right hand side).

These axioms allow creation from the unmarked state, interaction and replication or condensation of forms. We keep these actions in mind when composing proofs, and the proofs become a mixture of the combinatorics of the formalism and these concepts of creation and interaction.

This proof hinges on the identity

$$< < < A > > < A > > = 1 = < < A > A > = < A < A > >$$

Inside this identity $< < A > >$ can be replaced by **A**. The interaction of Axioms 4 and 5 yields the result in its full generality.

Once we begin to look at it this proof seems like a kind of miracle. It is a beautiful combination of creations and interactions that couple together in a quite unexpected way. It is very difficult to obtain an intuition for this kind of mathematics, and it is in just this domain that the Robbins Problem lives.

In the next section, we will translate the Robbins Problem into a bracket and graphical formalism. Then it is possible to begin a human look at the difficulties of the problem.

IV. Box Notation

The contraction of notation in the last section does not address the problem of reading nested parentheses. Instead of using the bracket notation, we can replace each pair of brackets (forming a notational enclosure) by a box. Now look at the graphics of boxes as shown in Figure 0. This Figure illustrates in box notation the basic Axioms 4 and 5 of the previous section. The box axioms take the form of allowed replacements for patterns of boxes with algebraic labels in their spaces. As before, commutativity and associativity of the juxtaposition of these forms are taken for granted.

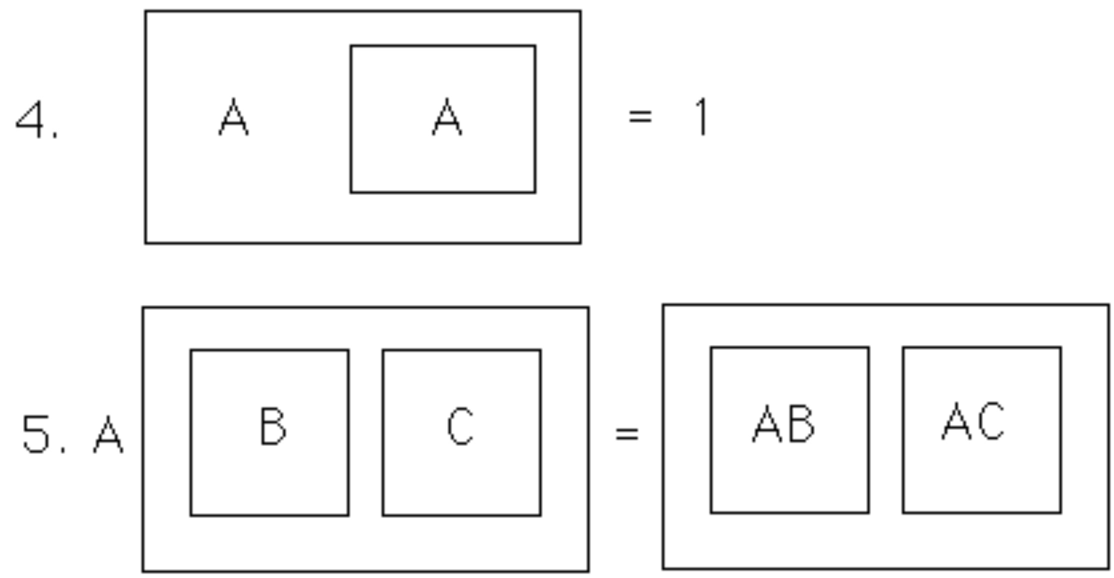

Figure 0

In Figure 1, we use box notation to illustrate the proof of $\langle \langle \mathbf{a} \rangle \rangle = \mathbf{a}$. With the box notation it is easier for the investigator to concentrate on the patterns generated by applications of Axioms 4 and 5. The eye is immediately met with the regions delineated by the boxes and hence by the corresponding enclosures of parentheses. By the same token it is useful to become adept at doing the algebra in the box notation. The reader would benefit from attempting to do the derivation of Figure 1 with only the basics in Figure 0 for reference!

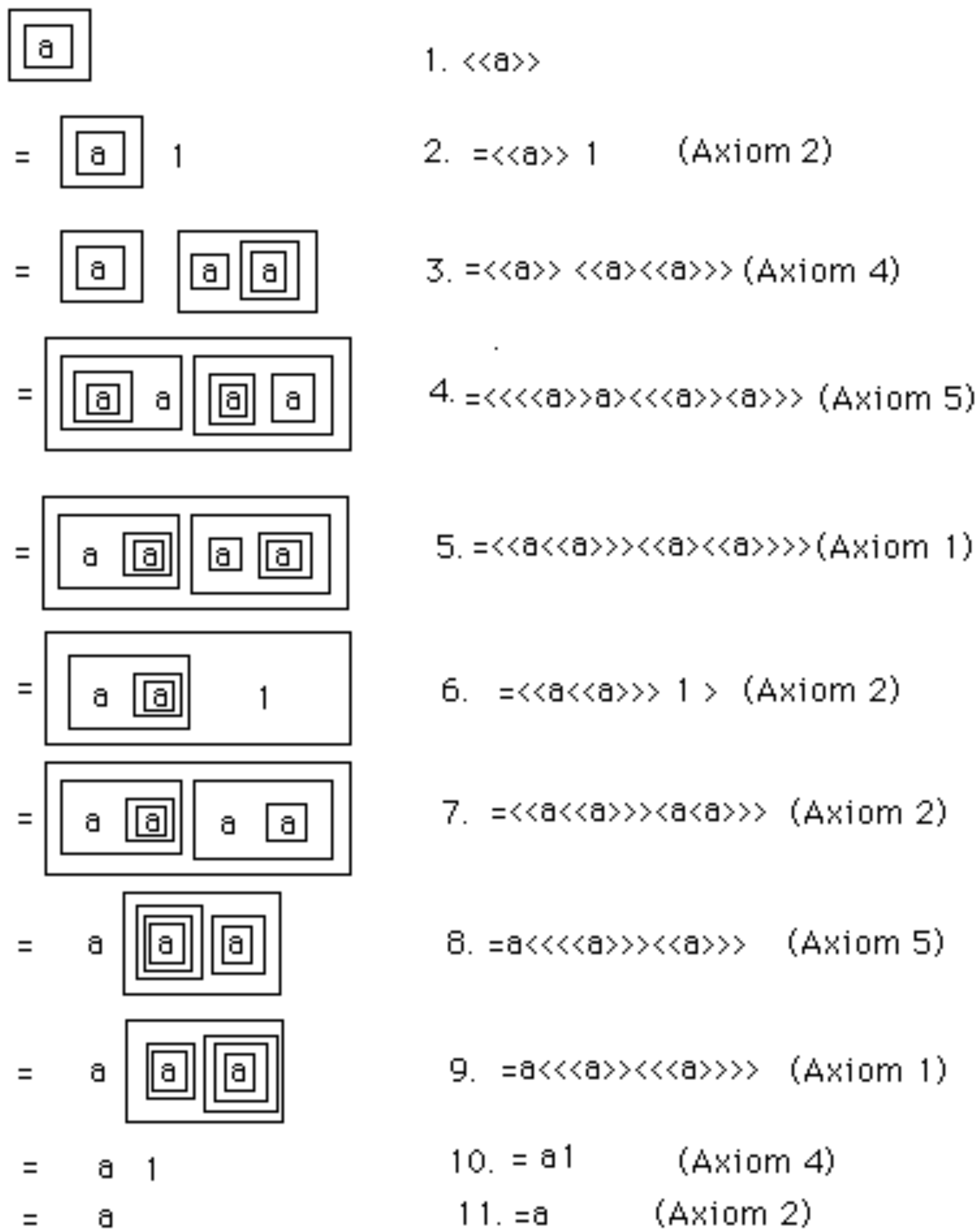

Figure 1. A Boolean Box Derivation

Remark. Note that this box notation is quite distinct from the Venn diagrams that are commonly used to illustrate Boolean properties. Box notation is designed to facilitate the algebra itself rather than to provide (as do the Venn diagrams) a visual domain whose description is provided by the algebra. We can use box notation in a context where the algebra is not Boolean or not known to be Boolean. In the box notation a geometric notation is seen to interface seamlessly with the algebraic language. The reader who is familiar with Venn diagrams will enjoy exploring the similarities and differences.

In Figure 2 we illustrate the corresponding box notation for the Robbins axiom. In dealing with Robbins algebra in this notation, we use associativity and commutativity without mention. This is a one-axiom system (with commutativity and associativity taken for granted). What does this axiom allow its user to do?

Of course, the axiom can always be applied to make a given expression more complex. It can also be applied to simplify an expression if that expression has the right pattern. The pattern that we see is a pattern of two adjacent boxes inside a larger box. The two adjacent boxes each contain two adjacent expressions, one of the form $< \mathbf{X} > \mathbf{Y}$ and the other of the form **XY.** Here $< \mathbf{X} >$ stands for **X** with a box around it. Thus in bracket notation the Robbins Axiom reads $< < < \mathbf{X} > \mathbf{Y} > < \mathbf{XY} > > = \mathbf{Y}$.

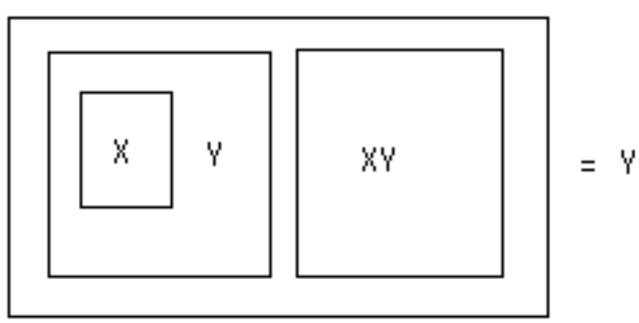

Figure 2 - Robbins Axiom A1 in Box Notation

One way to use this axiom is shown below in bracket form:

$$< < < X > Y < XY > > Y > =$$
$$< < < X > Y < XY > > < < < X > Y > < XY > > > =$$
$$< XY >$$

In this use of the Robbins axiom the **Y** in the first expression is expanded (by the axiom) in such a way that the entire new expression can be simplified by a global application of the axiom. We get the new identity

$$< < < X > Y < XY > > Y > = < XY >.$$

The method by which we derived this consequence illustrates a general principle of work with patterns. To wit - it may be necessary to use an expanded version of a condensed form (as in putting $< < < X > Y > < XY > >$ in place of **Y**) in order to see how the condensed form can interact with its environment. We call this the

Principle of Sufficient Expansion: *In order to see how a given pattern can interact in the context of other patterns, the given pattern must be expanded to a state of potential interaction with those patterns in the context.*

The Principle of Sufficient Expansion is operative in all areas of thought. Any individual participant in a conversational domain can and does create "internal" dialogues, symbolism and imagery that are, in a condensed state, intelligible only to that participant. The participant who is seen as a good conversationalist is a one who can take that condensed and personal imagery and expand it to sufficient form for successful interaction in the global domain. Language itself is an instance of the Principle. Mozart comprehends his symphony in an instant. A mathematician grasps the essence of a proof at once, and spends a year in the writing out of that understanding.

The significance of the Robbins Problem is that it challenges us to see the limits of (algebraic) reasonings that depend only on (a formal image of) the Principle of Sufficient Expansion. The outer box in the Robbins Axiom precludes any direct action through juxtaposition. The only recourse is through interaction generated by Sufficient Expansion - producing "large" instances of the Axiom by the use of "small" instances that drive the expansion.

To see this formalism in action, view Figure 3. Figure 3 shows a few steps in the Robbins Box algebra. In these steps we derive an equality between two expressions that are identical except that there is a $< < X > >$ in one expression where there is only an x in the other. In bracket notation the result is

$$< Y < < < X > > \; Y < < X > Y > > > \; =$$

$$< Y < \quad X \quad\; Y < < X > Y > > > \, .$$

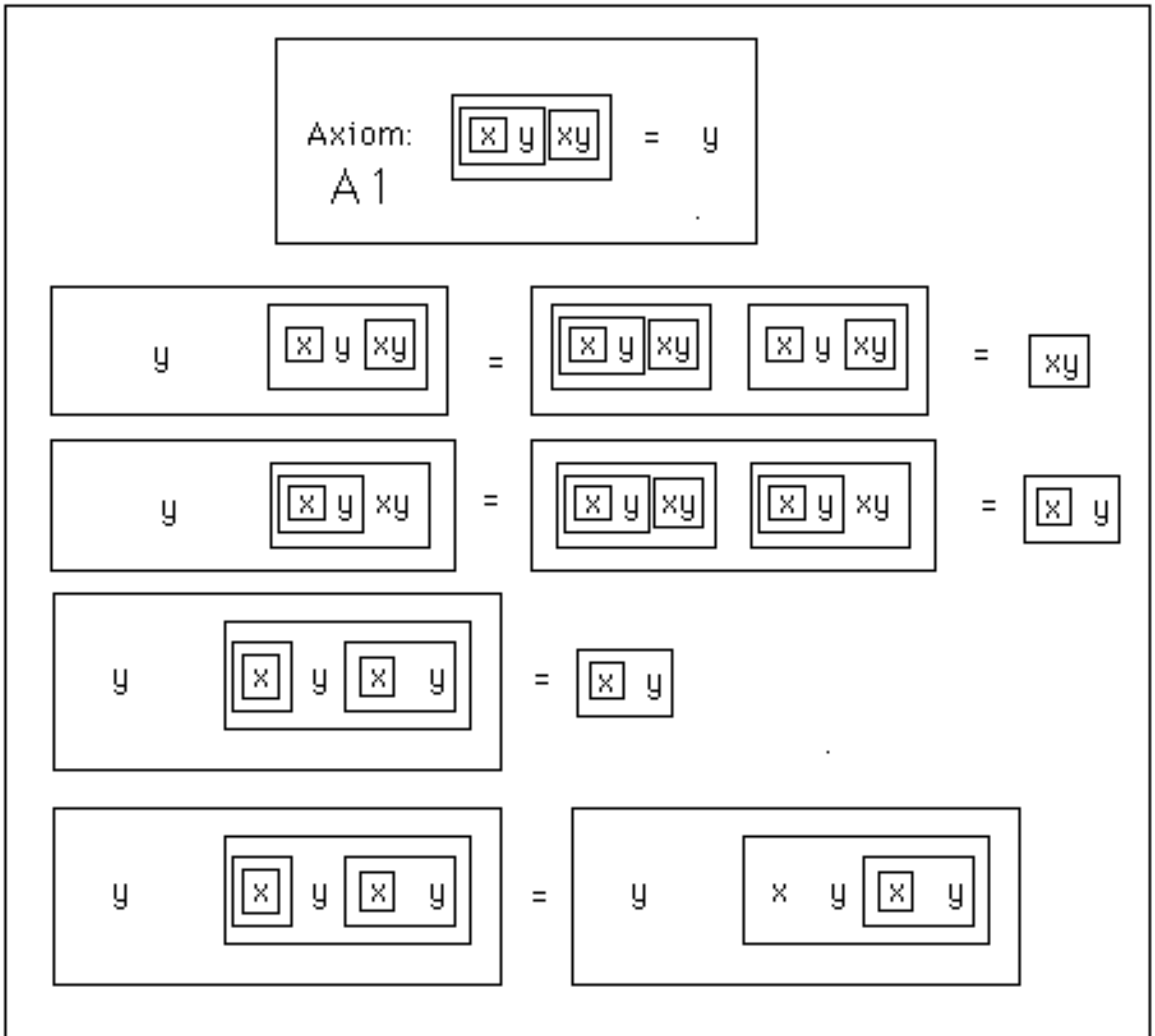

Figure 3 - A Few Steps in Robbins Box Algebra

This is a little hint that it may be possible to prove that
< < X > > = X for any x from the Robbins axioms. A direct proof of this fact would solve the Robbins problem by a result of Steve Winker [Winker, S. (1992)].

In order to solve the Robbins Problem, one must show that the axioms 4 and 5 (or equivalently the Boolean Box Axioms of Figure 0) follow from the Robbins Axiom. (All in the context of commutativity and associativity of the juxtaposition of forms.) In [Winker, S. (1992)] Steven Winker shows that to prove this, it is sufficient to find any instance of **< < X > > = X** or any instance of **< X Y >

$= \langle\ \mathbf{X}\ \rangle$. We shall call the latter identity *absorbtion*, and the former identity *reflection*. By Winker's results it is not necessary to prove either reflection or absorbtion in general. All that is needed is to find an instance of either one of these identities for some elements **X** and **Y** in the Robbins algebra. In this context **X** and **Y** must be non-empty as formal expressions. That is we cannot, in framing the Robbins problem, allow the empty word (the blank) as an element of the algebra. If the blank is allowed then the algebra contains a unit element and it is easy to see that it is Boolean if it contains a unit.

The successful strategy in the computer proof of the Robbins problem is the location of an instance of absorbtion. The computer was set to search for such an instance as a consequence of the axioms, and it succeeded. We will give a version of that proof via the Robbins Box notation in section 7 of this paper

.VI. Cybernetics, Semiotics, Boolean Algebra

One of the initial insights in the development of Boolean algebra was the astonishing idea (at the time) [Boole, George (1847)] that the formalism of ordinary algebra could be applied to elucidate the non-arithmetical subject of logic. Subsequent developments showed that George Boole's insight was correct way beyond all expectations. Boole's algebra marked a beginning of the understanding that entire (mathematical) subjects could be viewed as patterns to be mapped to one another and to different fields of application.

In the case of Boolean algebra, the field of possibility opened into practicality with Claude Shannon's ground-breaking paper [Shannon, C. E. (938)] showing the applicability of Boolean algebra to the structure and design of switching circuits. From that time onward Boolean algebra became the basis for the design of circuits and (later) digital computers. From Aristotelian logic to computer design, the algebra became the most useful and the simplest area of applied mathematics.

There is a beautiful balance in the now-standard formulation of Boolean algebra with two binary operations * (OR) and # (AND). Each can be expressed in terms of the other:

$$a*b = \sim(\sim a\#\sim b),$$

$$a\#b = \sim(\sim a*\sim b).$$

Each distributes over the other:

a*(b#c) = (a*b)#(a*c),
a#(b*c) = (a#b)*(a#c).

It is this symmetry (De Morgan's Law) that makes the choice of notation 0 or 1 for the two basic Boolean values a matter of convention (See the discussion in Section 2 in relation to this point.).

This symmetry is the result of the modern choice that uses "a or b, or both a and b" as the interpretation of "a OR b". Boole's original "or" was the exclusive or: "EXOR" = "a or b but not both a and b". Shifts of this kind are actually big shifts in concept.

In the box notation the operation * is given notational precendence by writing ab for a*b, and we have adopted a parenthesis <A> or a box around A to denote ~A. This has the advantage of reducing the number of distinct symbols in the system. In the case of standard Boolean algebra it has the interesting feature that we can use the absence of a character to denote the state 1. That is we have 1A =A so that 1 acts as though it was not present. We have ~1 = 0 (the other Boolean state) so that
in the parenthesis notation <1> = 0.

We divest ourselves of excessive notation by letting go of the notion that it is neccessary to write 1 and use a blank or unmarked state instead of 1. Then the equation 1A=A becomes simply A=A and this need not be said. And <1> = 0 becomes the equation < > = 0, saying that the empty brackets can be regarded as the stand-in for the other Boolean value 0.

In other words, we have the conception that there are two fundamental states. One state is unmarked and the other state is marked and designated by < >. The marked state is indicated by the process of transition (under negation) from the unmarked state. It is the indication of that transition that shows the new state. This is the underlying semiotics of Boolean algebra articulated by G. Spencer-Brown [Spencer-Brown, (1969)]. In this epistemology the act of transition from the unmarked state and the existence of the marked state are. in the form, identical.

How does the Robbin's problem appear from the point of view of Laws of Form? It is not a problem there! If we take as given an unmarked state then we can specialize the Robbins axiom

$$\mathbf{A} = \langle \langle \langle \mathbf{B} \rangle \mathbf{A} \rangle \langle \mathbf{BA} \rangle \rangle$$

to the special case where A is unmarked

$$= < < < B > > < B > > .$$

This means that any element of the form shown just above will act as a 1 in the algebra. It is then not hard to show that the resulting algebra is Boolean (See [Kauffman, L.H. (1990)].). The difficulty of the Robbins problem is based on the assumption that no variable symbol can *initially* be replaced by an unmarked state. It is then perhaps no wonder that it takes a great deal of work to produce a combination of symbols that acts as an unmarked state under the Robbins axioms. This is exactly one of the conseqences of the result that Robbins algebras are Boolean. Something can vanish after all!

The Robbins problem has the flavor of a fairy tale that shows that finally there is an escape from an artificial labyrinth made by insisting on existence rather than allowing non-existence.

Our discussion is a discussion of properties of domains of symbolic communication. The symbols themselves are imaginary. Silence and markedness achieve balance in the end.

In its fields of application Boolean algebra works hand in hand with matters of design and invention. The algebra also reaches its own limitation in these domains when direct methods or the needs of design context require a excursion beyond. For example, in designing digital circuitry it is neccessary to consider the effects of recursive applications of Boolean functions, and this is a deeper subject than the initial formalism of the algebra itself. By the same token one may consider circuits that internally embody a feedback mechanism. This results in stable and unstable local assignments of Boolean (two valued) states. Such systems can be modelled by recursions, but they can also be regarded as processes with states and transitions to be analysed synchronously or asynchronously. Here one needs to go beyond the Boolean algebra in order to understand the workings of the system as a whole.

In the case of the Robbins problem we see that its solution appeared to require an excursion beyond the usual human mathematical conversation into a conversation that incuded computing machinery. This placed the human beings in an extended system that incuded the machine. We have extended this conversation further by making new mathematics and changes of languge to facilitate the conversation/understanding as a whole. Understanding can arise in any conversation. We touch here on the unsolved problem of what contexts will most effectively generate understanding. There are also curious metophors that deserve further exploration. Do we understand the machine. Does the machine understand our instructions? Where is the understandng of the proof? Who understands this proof? Mathematics complicates and clarifies the issue with its formal nature. John von Neumann is said to have said "You do not understand mathematics; you just get used to it." It requires perseverance, creativity and hard work to get used to new mathematics. We get used to the loops and circularites of structure that constitute the terrain of our conversation. In this repect the map becomes the territory and the territory becomes the map. In that interlock we come to understanding.

A key underlying issue is the essential circularity of our (human) way of working. The simplest example of a system with circularity (feedback/reentry) is seen in the apparently simple Boolean equation

$$F = \sim F.$$

This innocent equation can be regarded as a circuit in which the signal (F) is fed back to itself through an inverter (the NOT:~). The result is that when F=0 then F BECOMES 1 and when F=1 then F BECOMES 0. The circuit behaves temporally as an oscillation between the Boolean values of 0 and 1. If we wish to assign a non-temporal solution to this equation then it is neccessary to consider domains of values beyond Boolean as in multiple valued logic or as in Spencer-Brown's concept of imaginary Boolean values. The point about the imaginary Boolean values is that they come up immediately as soon as we look at Boolean algebra in the context of an application. Boolean algebra in any cybernetic context cries for an extension into the imaginary. In the end this is the extension into the human domain.

Here we at last come in contact with the conversation and theory of Gordon Pask, particularly his later work involving nets and patterns of networks of communication. In this work he utilized topological pattern and imaginary values to indicate and model the complexities and levels of conversation, the weave of listeners speakers participants. It would take us too far afield to go into these matters here, but I am sure that the way in which a complex network of mathematics and communication evolved among humans and machines, just to understand the Robbins problem, would have fascinated Gordon and would have led to new conversation about conversation.

VII. Solution of The Robbins Problem

The derivation that follows consists of fourteen consequences of the Robbins Axiom, labelled C2,C3, ..., C15. We have started the numbering with 2 because the Axiom itself is labelled A1. All references in the derivations are to A1 and to previously derived consequences. Since we take commutativity and associativity for granted, there are no references to the applications of these implicit axioms. Consequence C15 is an instance of absorbtion, thus solving the Robbins Problem.

The consequences in this sequence are box reformulations of the consequences in the computer proof of the Robbins problem. The derivations are my own, and in the course of making them I simplified the computer proof of the Robbins problem by eliminating one of the computer's consequences. It would be very interesting to simplify this proof further, or to find a different and simpler proof.

Axiom:
A1

$$\boxed{\boxed{\boxed{x}\ y}\ \boxed{xy}} = y$$

C2. $\boxed{\boxed{\boxed{xy}\ \boxed{x}\ y}\ y} = \boxed{xy}$

Dem. $\boxed{\boxed{\boxed{xy}\ \boxed{x}\ y}\ y} = \boxed{\boxed{\boxed{xy}\ \boxed{x}\ y}\ \boxed{\boxed{xy}\ \boxed{\boxed{x}\ y}}}$ (A1)

$= \boxed{xy}$ (A1). //

C3. $\boxed{\boxed{\boxed{\boxed{x}\ y}\ xy}\ y} = \boxed{\boxed{x}\ y}$

Dem. $\boxed{\boxed{\boxed{\boxed{x}\ y}\ xy}\ y} = \boxed{\boxed{\boxed{\boxed{x}\ y}\ xy}\ \boxed{\boxed{\boxed{x}\ y}\ \boxed{xy}}}$ (A1)

$= \boxed{\boxed{x}\ y}$ (A1). //

C4. $\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}} = y$

Dem. $\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}} = \boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{\boxed{\boxed{x}\ y}\ xy}\ y}}$ (C3)

$= y$ (A1).//

C5. $\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ z}\ \boxed{yz}} = z$

Dem. Let $a = \boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}$. Then $\boxed{a} = y$ by (C4).

Therefore,

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ z}\ \boxed{yz}} = \boxed{\boxed{az}\boxed{yz}} = \boxed{\boxed{az}\ \boxed{\boxed{a}\ z}} = z,$$

using (A1). //

C6. $$\boxed{\boxed{\boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyy}\ y} = \boxed{\boxed{x}\ y}$$

Dem. $$\boxed{\boxed{\boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyy}\ y} \underset{(C4)}{=} \boxed{\boxed{\boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}}}$$

$$\underset{(A1)}{=} \boxed{\boxed{x}\ y}\ .//$$

C7. $$\boxed{\boxed{\boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyyy}\ \boxed{\boxed{x}\ y}} = y$$

Dem. Let $a = \boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyy$. Then $\boxed{\boxed{a}\ y} = \boxed{\boxed{x}\ y}$ by (C6).

$$\boxed{\boxed{\boxed{\boxed{x}\ y}\ \boxed{\boxed{x}\ y}\ xyyy}\ \boxed{\boxed{x}\ y}} = \boxed{\boxed{ay}\ \boxed{\boxed{x}\ y}} = \boxed{\boxed{ay}\ \boxed{\boxed{a}\ y}} = y.\ //$$

C8. $\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ \boxed{yz}\ z}\ z} \quad = \quad \boxed{yz}$

Dem. Let $a = \boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ z$. Then, by (C5), $\boxed{\boxed{a}\ \boxed{yz}} = z$.

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ \boxed{yz}\ z}\ z} \quad = \quad \boxed{\boxed{a\ \boxed{yz}}\ z}$$

$$= \quad \boxed{\boxed{a\ \boxed{yz}}\ \boxed{\boxed{a}\ \boxed{yz}}} \quad = \quad \boxed{yz} \qquad //$$

C9. $$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy\ \boxed{\boxed{x}\ y}\ \boxed{yz}\ z}\ zu}\ \boxed{\boxed{yz}\ u}} = u$$

Dem. Let $a = \boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ \boxed{yz}\ z}\ z$. Then $\boxed{a} = \boxed{yz}$ by (C8).

$$\boxed{\boxed{\boxed{\boxed{\boxed{\boxed{x}\ y}\ xyy}\ \boxed{\boxed{x}\ y}\ \boxed{yz}\ z}\ zu}\ \boxed{\boxed{yz}\ u}} = \boxed{\boxed{au}\ \boxed{\boxed{yz}\ u}} = \boxed{\boxed{au}\ \boxed{\boxed{a}\ u}}$$

$$= u \quad //$$

C10. $$\boxed{\boxed{\boxed{\boxed{x}\ x}\ xxx}\ x} = \boxed{\boxed{x}\ x}$$

Dem. In (C9), let $u = \boxed{\boxed{x}\ x}$ and $y = x$. Then (C9) reads:

$$\boxed{\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ xxx}\ \boxed{\boxed{x}\ x}\ \boxed{xx}\ x}\ x\ \boxed{\boxed{x}\ x}}\ \boxed{\boxed{xx}\ \boxed{\boxed{x}\ x}}} = \boxed{\boxed{x}\ x}.$$

Note that by (A1), $\boxed{\boxed{xx}\ \boxed{\boxed{x}\ x}} = x.$

Let $A = xx$, $B = \boxed{\boxed{x}\ x}\ x$. Then

$$\boxed{\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ xxx}\ \boxed{\boxed{x}\ x}\ \boxed{xx}\ x}\ x\ \boxed{\boxed{x}\ x}}\ \boxed{\boxed{xx}\ \boxed{\boxed{x}\ x}}}$$

$$= \boxed{\boxed{\boxed{\boxed{AB}\ \boxed{A}\ B}\ B}\ x} = \boxed{\boxed{AB}\ x} \qquad \text{(C2)}$$

$$= \boxed{\boxed{\boxed{\boxed{x}\ x}\ xxx}\ x}.$$

This completes the demonstration. //

C11. $\boxed{\boxed{\boxed{\boxed{x}\ \ x}\ \ xxxx}\ \boxed{\boxed{x}\ \ x}} \ = \ x$

Dem. Let $a = \boxed{\boxed{x}\ \ x}\ xxx$. Then, according to (C10),

$\boxed{\boxed{a}\ \ x} = \boxed{\boxed{x}\ \ x}$. Therefore,

$$\boxed{\boxed{\boxed{\boxed{x}\ \ x}\ \ xxxx}\ \boxed{\boxed{x}\ \ x}} = \boxed{\boxed{a\ \ x}\ \boxed{\boxed{x}\ \ x}} = \boxed{\boxed{a\ \ x}\ \ \boxed{\boxed{a}\ \ x}} = x. \ //$$

C12. $$\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ x} = \boxed{\boxed{x}\ x}$$

Dem. Let $a = \boxed{\boxed{x}\ x}\ xxxx$. Then (C11) asserts that $\boxed{\boxed{a}\ \boxed{\boxed{x}\ x}} = x$.

$$\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ x} = \boxed{\boxed{\boxed{\boxed{x}\ x}\ a}\ x}$$

$$= \boxed{\boxed{\boxed{\boxed{x}\ x}\ a}\ \boxed{\boxed{a}\ \boxed{\boxed{x}\ x}}}$$

$$= \boxed{\boxed{x}\ x}. \quad //$$

C13. $\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ xy}\ \boxed{\boxed{\boxed{x}\ x}\ y}} \ = \ y$

Dem. Let $a = \boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ x$.

According to (C12), $\boxed{a} = \boxed{\boxed{x}\ x}$.

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ xy}\ \boxed{\boxed{\boxed{x}\ x}\ y}} = \boxed{\boxed{ay}\ \boxed{\boxed{\boxed{x}\ x}\ y}}$$

$$= \boxed{\boxed{ay}\ \boxed{\boxed{a}\ y}} = y. \ //$$

C14. $$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ \ xxx}\ \ x}\ \ x} \quad = \quad \boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ \ xxxx}$$

Dem.

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ \ xxx}\ \ x}\ \ x} \quad = \quad \text{(C7)}$$

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ \ xxx}\ \ x}\ \boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ \ xxxx}\ \boxed{\boxed{x}\ \ x}}} \ \underset{\text{(C10)}}{=}$$

$$\boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ \ xxx}\ \ x}\ \boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{\boxed{x}\ x}\ \ xxx}\ \ x}}}$$

$$\underset{\text{(A1)}}{=} \ \boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ . \quad //$$

C15. $\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx} = \boxed{\boxed{\boxed{x}\ x}\ xxx}$

Dem.

$$\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx} = \boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ xxx}\ x}\ x} \qquad (C14)$$

$$= \boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ \boxed{\boxed{\boxed{x}\ x}\ xxx}\ x}\ \boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{\boxed{x}\ x}\ xxx}}} \qquad (C4)$$

$$= \boxed{\boxed{\boxed{\boxed{\boxed{x}\ x}\ \boxed{\boxed{x}\ x}\ xxxx}\ y\ x}\ \boxed{\boxed{\boxed{x}\ x}\ y}}$$

$$= y \qquad (C13)$$

$$= \boxed{\boxed{\boxed{x}\ x}\ xxx}\ . \quad //$$

References

1. Boole, George (1847). "The Mathematical Analysis of Logic", Cambridge.

2. Huntington,E.V. (1933) , Boolean Algebra. A Correction. *Trans. Amer. Math. Soc.* **35** , pp. 557-558.

3. Kauffman, L.H. (1990). Robbins Algebra. Proceedings of the Twentieth International Symposium on Multiple Valued Logic. 54-60, IEE Computer Society Press.

4. Kauffman,L.H. (1997a). Website: <http://math.uic.edu/~kauffman/>

5. Kauffman,L.H. (1997b). Column on Virtual Logic, in "Cybernetics and Human Knowing", Vol. 4, No. 1(1997), 105-109.

6. Ketner, Kenneth Laine (1990). "Elements of Logic - An Introduction to Peirce's Existential Graphs", Texas University Press.

7. McCune,W. (1996), Website: <http:www.mcs.anl.gov/home/mccune/ar/robbins>.

8. Peirce,C.S. (1898/1992). "Reasoning and the Logic of Things" - The Cambridge Conferences Lectures of 1898, Edited by Kenneth Laine Ketner, Harvard University Press, Cambridge Massachusetts and London, England (1992).

9. Shannon, C. E. (1938). A symbolic analysis of relay and switching circuits. Trans. Am. Inst. elec. Eng. Vol. 57 (1938), 713-723.

10. Spencer-Brown,George (1969). "Laws of Form", George Allen and Unwin Ltd., London.

11. Winker, S. (1992). Absorbtion and Idempotency Criteria for a Problem in Near-Boolean Algebras, *Journal of Algebra* **153**, 414-423.