The Algebra of States and Events
Author(s): Edmund C. Berkeley
Source: *The Scientific Monthly*, Vol. 78, No. 4 (Apr., 1954), pp. 232-242
Published by: American Association for the Advancement of Science
Stable URL: http://www.jstor.org/stable/21396
Accessed: 01/05/2014 09:31

The Algebra of States and Events

EDMUND C. BERKELEY

The author graduated summa cum laude in mathematics from Harvard University in 1930 and did actuarial work in life insurance companies until 1948. He is a Fellow of the Society of Actuaries, and for over five years was secretary of the Association for Computing Machinery. Since 1948 he has been operating his own business as consultant and publisher.

The Algebra of Classes, or Boolean Algebra

BY now a good many people have heard of the algebra of classes, also called Boolean algebra. This algebra has for its main operators AND (·), OR (v), NOT (′), in the same way as elementary algebra has for its main operators PLUS (+), MINUS (−), TIMES (·), and DIVIDED BY (/). Elementary algebra operates with numbers and with letters that stand for numbers; but Boolean algebra operates with classes or with statements and with letters that stand for classes or statements. The name comes from George Boole, a great English mathematician. (1815–64).

Boolean algebra has important applications in the outlining of rules and contracts, and the designing of on-off circuits for switching and computing; it is being studied and uesd widely in laboratories throughout the country. Boolean algebra is interesting and not difficult, and since it contains no numbers as coefficients as in $3m$ or as exponents as in b^4, it is essentially simpler than ordinary elementary algebra.

Examples of Boolean Algebra

An example of the use of Boolean alegbra for handling conditions is shown in the folowing problem (due to another Englishman, John Venn, about 1894).

Problem 1. A certain club has the following rules: (1) The financial committee shall be chosen from among the general committee. (2) No one shall be a member of both the general and library committees unless he is also on the financial committee. (3) No member of the library committee shall be on the financial committee. Simplify these rules.

Answer: The rules may be simplified as follows: (1) The financial committee shall be chosen from among the general committee. (2) No member of the general committee shall be on the library committee.

Another example of the use of Boolean algebra for dealing with circuit elements is shown in the following problem.

Problem 2. The relay D is energized if and only if all the switches A, B, and C are closed. The relay E is energized if and only if D is not energized and any one or more of the switches A, B, and C are closed. (See Fig. 1). Simplify the circuit for E.

For the benefit of those readers who may not be familiar with relay circuits, an electrical relay is essentially a device consisting of a coil of wire wound around a soft iron core having the property that a contact is switched or transferred when the core is magnetized by running current through the coil. In Fig. 2 the physical structure of a relay is shown: when current flows from

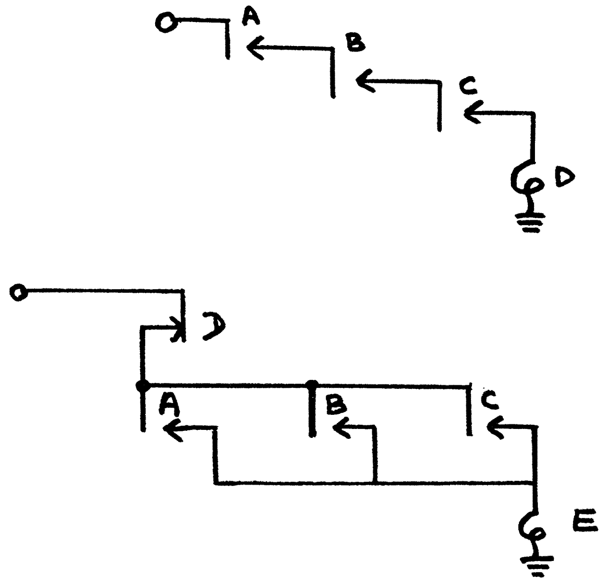


FIG. 1. Circuits to be simplified: the Boolean equations are (1) $D = A \cdot B \cdot C$; (2) $E = D'(A \vee B \vee C)$.

pickup P to ground, the magnetic force pulls down the iron flap or "armature" above, causing the transfer contact T to move away from the normally closed contact NC over to the normally open contact NO. Relays may have more than one set of contacts, T, NC, NO, all switched at once. In Fig. 3 is shown the simpler schematic for a relay and its contacts used in drawing circuits involving a number of relays and contacts.

Answer: The circuit may be simplified as shown in Fig. 4.

Now Boolean algebra is not necessary in order to solve these particular problems, but it is sufficient to solve these problems and many more problems

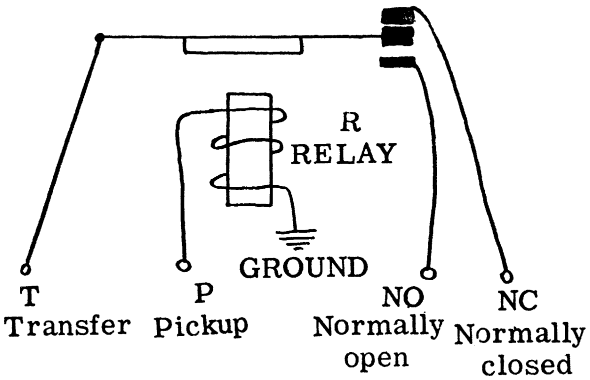


FIG. 2. Relay.

that are more difficult. And it is a straightforward and widely applicable method, more so than most other methods.

It is not our purpose here to explain and state all the rules of Boolean algebra so that it may be understood and calculated with. That information is available elsewhere*.

The Algebra of States and Events

However, there is a large class of problems either in circuits or in contracts involving conditions which change over time. Instead of classes with permanent members, these conditions give rise to classes with membership that fluctuates. Instead of circuits that have only one state, circuits arise that have a succession of states. In these cases Boolean algebra considered as a model for dealing with the situa-

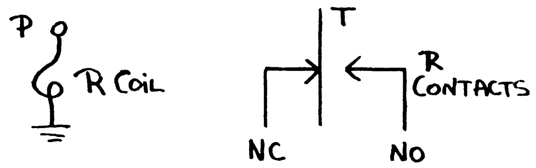


FIG. 3. Relay.

tion is deficient, for it contains no variable representing time.

Our purpose here is to present briefly and fairly simply an "algebra of states and events," an algebra which takes into account classes or statements, and a numerical variable representing time, and which is adapted to the study of conditions which change from time to time in some sequence, "sequential conditions." It is equivalent to Boolean algebra extended to include a numerical variable representing time.

Among the concepts contained in this algebra of states and events are BEFORE, AFTER, WHILE, DURING, HAPPEN, CHANGE, BEGIN, FINISH, IN THE EVENT THAT, DELAY, and similar concepts. We can represent these concepts with symbols that we can calculate with, and we can convert concepts symbolically from one form to another form to construct statements and circuits that satisfy sequential conditions.

Binary Variables

A *statement* is an expression which may be true or false. If P stands for a statement, and T(. . .) stands for the "truth value of . . . ," then we may say that $T(P)$ equals "truth" or "yes" if P is true, and "falsehood" or "no" if P is false. Now these words are not too convenient for compact handling; so let us use 1 for "truth" and 0 for "falsehood." In other words, the *truth value* of a statement P is 1 if P is true and 0 if P is false. If we do not know the truth value of a statement P (capital

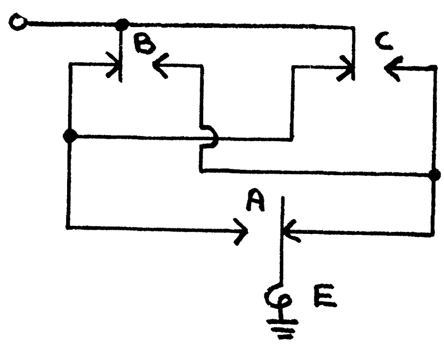


FIG. 4. Simplified circuit: the Boolean equation is $E = A(B' \vee C') \vee A'(B \vee C)$.

* One source is *Circuit Algebra—Introduction*, published by Edmund C. Berkeley and Associates, 36 West 11 St., New York, which also contains a fuller exposition than here given of the algebra of states and events. George Boole's original epoch-making book, *The Laws of Thought*, published in 1854, has been again reprinted, currently by Dover Publications, New York, 1951; but the form of the algebra given in that book has been vastly improved by later mathematicians, especially Ernst Schröder. Another source for Boolean algebra is Chapter XI, "Algebra of Classes," in *Survey of Modern Algebra*, 2nd ed., by Garrett Birkhoff and Saunders MacLane, Macmillan, New York, 1953.

letter), let us use p (lower-case letter) to stand for its unknown or undetermined truth value. Thus p is a binary variable, a variable which can have only 0 or 1 for its values.

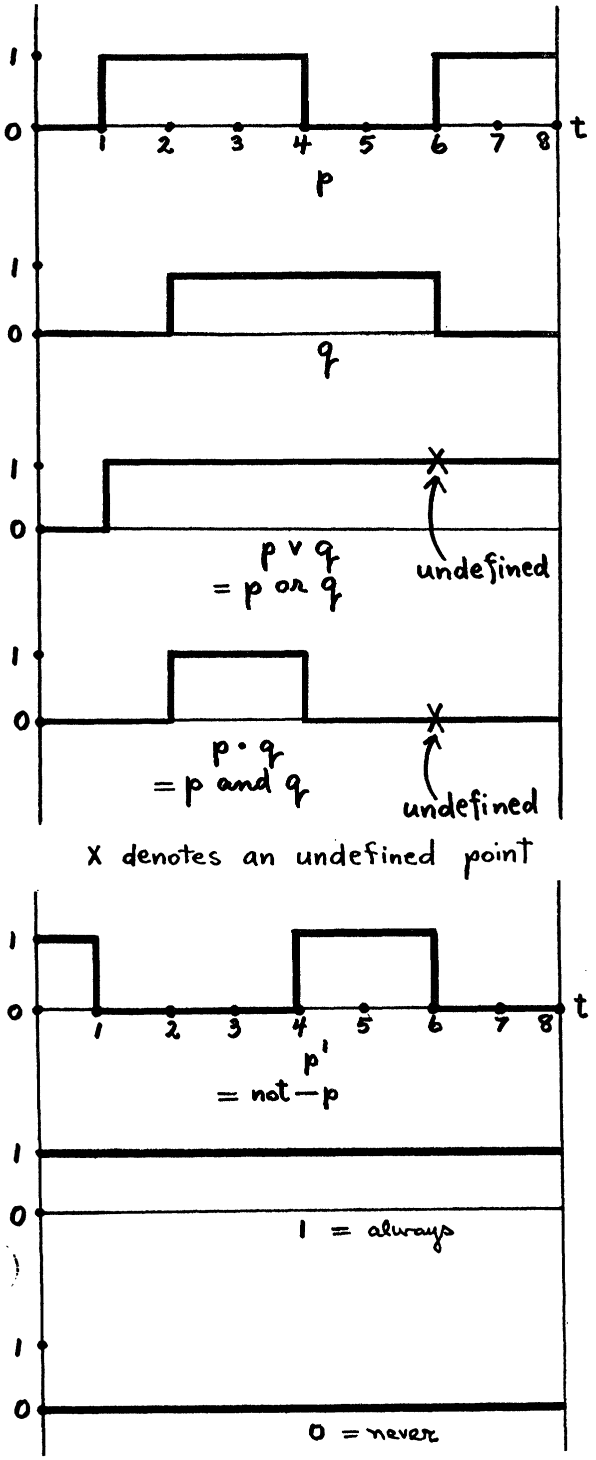


FIG. 5. Binary variables that are functions of time.

From the common meanings of AND, OR (inclusive-OR), and NOT, we can reason out that if P and Q are two statements with truth values p and q, then:

$$\mathrm{T}\ (P \text{ AND } Q) = p \cdot q$$
$$\mathrm{T}\ (P \text{ OR } Q) = p + q - pq$$
$$\mathrm{T}\ (\text{NOT}-P) = 1 - p$$

Here the plus sign, multiplication sign, and minus sign are the signs of elementary algebra, and may be used in just that way. It is convenient however to define two new operators. The operator $\vee$ (read "vee" or "or") is defined as: $p \vee q = p + q - pq$. The operator p' (read "not-p" or "p prime") is defined as $1 - p$. It is convenient to read $p \cdot q$ not only p TIMES q but also p AND q.

This much is all Boolean algebra.

Binary Variables Which Are a Function of Time t

Now let us consider a statement which mentions a time t. Such a statement may be "The circuit element E is ON at time t" or "John Q. Smith is alive at time t." The binary variable p which stood for the truth value of a statement P which did not mention time now becomes a binary variable $p(t)$ or p_t or simply p, which is equal to the truth value of a statement P which does mention time.

We now obtain a territory larger than Boolean algebra; we obtain a territory that may be called "the algebra of states and events," or "statal algebra," or "Boolean calculus." This territory is equal to Boolean algebra extended to provide for changes in time.

At any one time p has to be either 0 or 1 or undefined, but over an interval of time the value of p may change from 0 to 1 or from 1 to 0, and if the change is not instantaneous, during the change the value of p is undefined. We consider that we know the binary variable p completely if for each time in the interval we know whether it has the value 1 or the value 0 or is undefined.

The algebra of states and events is of course a natural extension of Boolean algebra, bringing it into closer relation with the real world. For in the real world, the members of a class of objects enter the class or leave the class as time changes. For example, a horse becomes a current member of the class of horses when it is born and ceases to be a current member of the class of horses when it dies. Also, a statement true at one time may not be true at another time. For example, a circuit element on at one time may be off at another time. It is customary in ordinary language to talk about one of these changes of state as an event, and so it is reasonable to speak of this extension of the algebra of classes as the algebra of states and events.

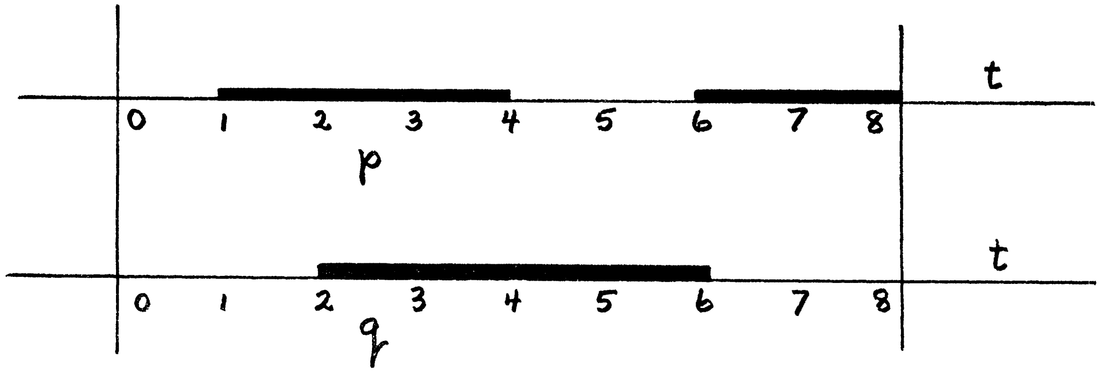


FIG. 6. Second way of graphing binary variables that are functions of time.

Time Units

When we start dealing with p, q, r as functions of time t, the first question we need to consider is: How finely shall we subdivide time? Ordinary differential calculus suggests that we treat time as a "continuous variable," allowing subdivision of time down to infinitely fine subdivisions, much finer than microseconds. This may be theoretically desirable, but it does not correspond very well with at least some aspects of the real world, and we are here primarily interested in the practical handling of conditions and the practical operation of circuits. Practically, it is useful to treat time as consisting of a succession of small finite units of time. For some circuits, seconds are good units, for others milliseconds, and for others microseconds. We shall suppose that we can measure time in units, and we shall often use as a unit either a "little unit," the time that it takes a condition or a circuit element of the type we may be considering to change state from 0 to 1 or from 1 to 0, or a "big unit," which is a large multiple of a little unit, such as 1000 times a little unit.

When we use "little" units of time, p will take all (or most) of a whole unit of time to go from 0 at the start of the unit of time to 1 at the end of the unit of time, and vice versa. When we use "big" units of time, p will ordinarily have just one of the values 0 or 1 for a considerable stretch of time, and every now and then changes will take place as jumps at the beginning or end or some other point in the "big" unit of time. Of course, there will be situations in the real world where neither of these simplifying assumptions works very well, and in such cases we may need to make different assumptions.

Step-Functions

What will our variables p, q, r as functions of t look like when graphed? They will be entirely familiar to us as "step-functions," and will look like the graphs shown in Fig. 5. Here we have provided for 8 "big" units of time along the horizontal axis; we have assumed that it takes the circuit element we are imagining a thousandth of one of these graphed units of time to change from 0 to 1 or from 1 to 0. Also, we have assumed that changes may occur only at the junctions of the intervals.

The first graph shows the function p, which is defined as:

Continuous t	p	Counted t	p
0 to 1	0	1	0
1 to 4	1	2, 3, 4	1
4 to 6	0	5, 6	0
6 to 8	1	7, 8	1

Close to the time t equal to 4 (continuous time t), for example, we shall not know in practice whether p equals 0 or 1; and so here p is undefined. And in practice we contrive the circuit to avoid reliance on such a circuit element at such a time.

The same situation occurs when we go to conditions in real life instead of circuit elements. Suppose p is the truth value of "John Q. Smith is dead at time t." Now there is a day when John Q. Smith dies, and during that day there are moments when he is not yet dead, and moments when he is dead, and a group of moments in between when no one, friend, doctor, or scientist, can actually determine whether he is alive or dead. In those situations, p is in actual fact not defined, and there is little sense in being arbitrary about it and insisting on one definition or another; in actual fact we avoid relying on the information p at a time when it cannot actually be determined whether p is 1 or 0.

In Fig. 5, we have also shown the graph of a function q, which is defined as:

Continuous t	q	Counted t	q
0 to 2	0	1, 2	0
2 to 6	1	3, 4, 5, 6	1
6 to 8	0	7, 8	0

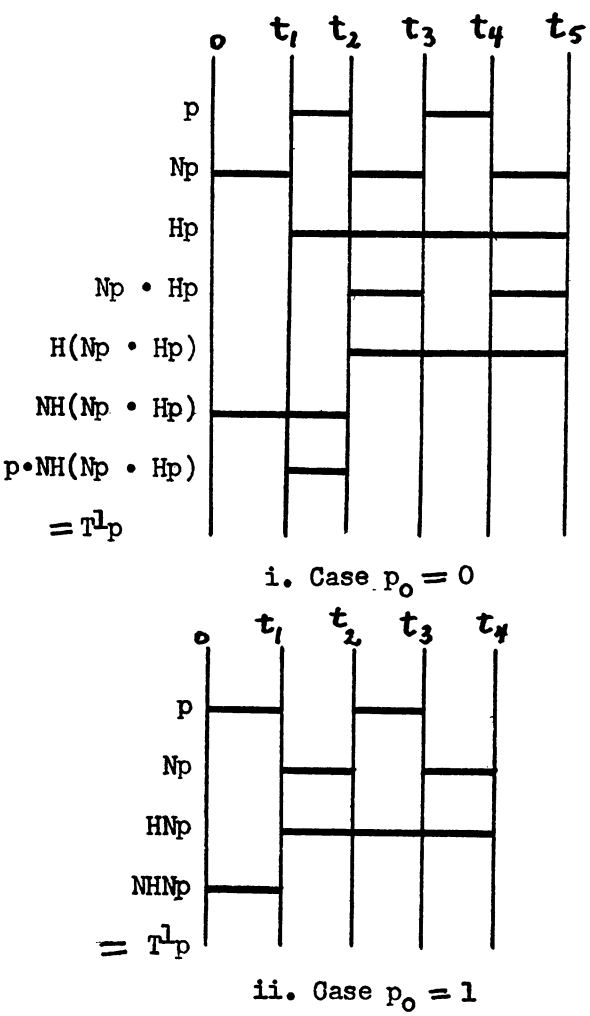


FIG. 7. Calculations of T^1p.

Also, the functions p OR $q = p \vee q$, p AND $q = p \cdot q$, and NOT-$p = p'$, and the functions "always 1" (OR "always") and "always 0" (OR "never"), are also graphed and shown in Fig. 5. Note that $p \vee q$ and $p \cdot q$ have undefined points at $t = 6$.

There is a second way of graphing p, q, r as functions of time, and that is to draw the graph only when it has the value 1. Often this is much more convenient, especially for calculation. See Fig. 6.

In a case where a finite length of time is counted in "little" units, a table of the values of a binary variable p may look like the following:

t	p	t	p
1	0	6	1 to 0
2	0	7	0
3	0 to 1	8	0 to 1
4	1	9	1 to 0
5	1	10	0

This might express the sequential history of a patient with a health record as follows:

t	Statement P	t	Statement P
1	Well	6	Penicillin shot—temperature went to normal
2	Well	7	Normal temperature
3	Fell sick	8	Temperature started up again
4	Fever	9	Another penicillin shot, and down to normal
5	Fever	10	Normal temperature

New Operators

Up to this point, we still have what is almost Boolean algebra; but now we introduce new operators which express such ideas as "before, during, after, happening, starting, finishing, delay, change," etc. See Table 1; it should be examined closely.

The definitions of the new operators given in Table 1 are in many respects obvious. It is easy to express them in terms of each other, and to state theorems. However, in every practical situation the points or intervals where the variables are undefined must be kept in mind and attended to carefully. Following are some comments on a number of the operators.

Delay. The "delay" operator is symbolized D. $D^k(p)$, also written D^kp and p^{+k}, is equal to the binary variable p delayed k units of time, which equals p_{t+k} or $p(t+k)$. In fact, if we think of p as a function f of time t, equal to $f(t)$, then p^{+k} is equal to $f(t+k)$. Note that D^kp is not defined in the interval $t = 0$ to $t = k$, and $D^{-k}p$ is not defined in the interval $t = n - k$ to $t = n$, where n is the last time being considered. An abbreviation for p^{+1} is p' or $\dot{p}$.

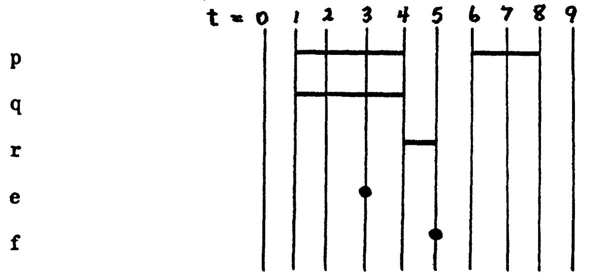
Happen. The operator "happen" or "has happened" is symbolized H. $H(p)$, also written Hp, is equal to 1 for every time t after the binary variable p takes on the value 1. More exactly, $H(p_t) = 1$ if and only if there exists a time t_2 such that $t_2 < t$ and $p(t_2) = 1$.

There is an implicit equation relating the delay operator and the happen operator. Let m be a small amount of time (a *m*oment), in particular the smallest amount of time sufficient for an on-off circuit element to safely change from 0 to 1 or from 1 to 0. Suppose also that the duration of p is at least as long as m. Then:

$$H(p) = p \vee D^mH(p)$$

In words, "p has happened if and only if p exists currently, or at a moment ago p has happened, or both."

Symbol	Meaning
	I. Independent variables (functions of time)
p	a state
q	another state
r	a third state
e	an event
f	another event



Symbol	Meaning
	II. Dependent variables (functions of the independent variables)
Ip	while p, during p, *i*dentically p
Ie	when e, at e, *i*dentically e
T^1p	p for the 1st *t*ime $(=q)$
T^2p	p for the 2nd *t*ime
Hp	p has *h*appened
H^2p	p has *h*appened for the 2nd time
D^1p, $p^{\cdot}$, p^{+1}	p *d*elayed 1 unit of time
D^2p, p^{+2}	p delayed 2 units of time
D^{-1}, p^{-1}	p advanced 1 unit of time
Np	*n*ot-p, except when p, except during p, p'
NHp	p has *n*ot *h*appened
HNp	*n*ot-p has *h*appened
Bq	*b*efore q
Aq	*a*fter q, q has stopped
Eq	*e*ntering q, starting q, beginning of q, just in q
Fq	*f*inishing q, leaving q, end of q, just out of q
C(p,r)	the event state p *c*hanging to state r
S(e,f)	the state from event e to event f
Zp	$p_0 = 0$; p at time 0 is *z*ero

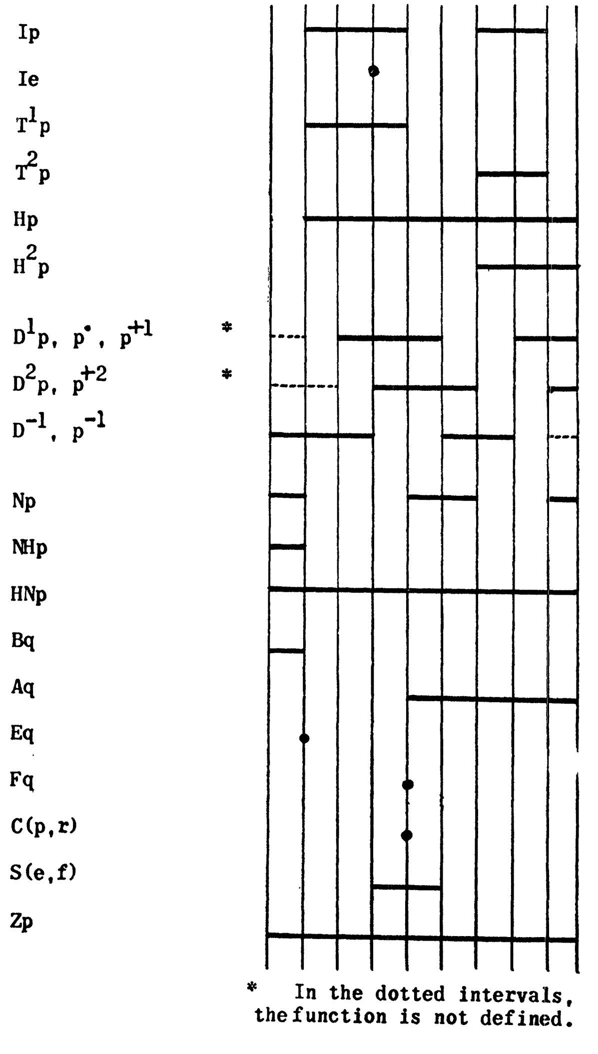


TABLE 1. Operators of the algebra of states and events.

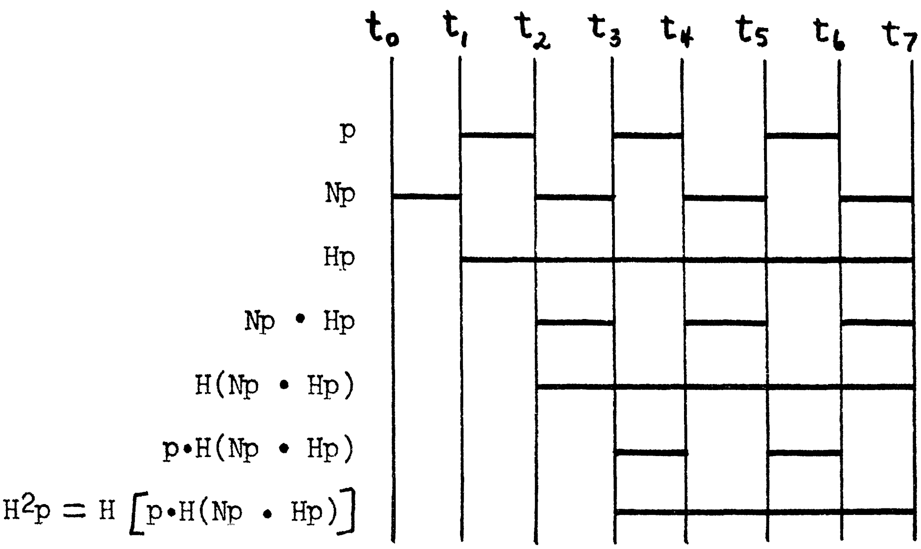


FIG. 8. Calculation of "Has happened for the second time."

Using operators alone, this is:

$$\mathrm{H} = \mathrm{I} \vee \mathrm{D}^m\mathrm{H}$$

where I is the "identity" operator, that is, $\mathrm{I}(p) = p$.

Not. The "not" operator may be symbolized N. $\mathrm{N}(p)$, also written Np, or p', is equal to 0 when p is 1, and 1 when p is 0, and is undefined when p is undefined.

Before. The "before" operator is symbolized B. Let q be a binary variable which has only one *stretch,* that is, only one continuous interval of time during which it has the value 1. Then "before q" is the same as "q has not happened"; in symbols, Bq = NH q. Using operators alone, B = NH.

After. The "after" operator is symbolized A. Then "after q" is the same as "not q, yet q has happened." In symbols, $\mathrm{A}q = \mathrm{N}q \cdot \mathrm{H}q$. Using operators alone, $\mathrm{A} = \mathrm{N} \cdot \mathrm{H}$.

"Before" and "after" can be defined in more than one way for a binary variable with more than one stretch. The definition may take into account such possible meanings as "before the first stretch," "before the last stretch," and "after some stretch." Note that "before any stretch" and "after any stretch" may be ambiguous since "any" may mean "every" or "at least one." The definitions given above for B and A, when applied to a binary variable with more than one stretch, mean "before the first stretch" and "after the first stretch," respectively.

Entering. The "entering" or "starting" operator is symbolized E. Let m be the smallest amount of time sufficient for an on-off circuit element to safely change from 0 to1 or from 1 to 0. It is then a useful definition of "entering" q to say that "entering q" is the same as "q, but a moment ago not-q." In symbols, $\mathrm{E}q = q \cdot \mathrm{ND}^m(q)$. Using operators, $\mathrm{E} = \mathrm{I} \cdot \mathrm{ND}^m$.

Leaving. The "leaving" or "finishing" operator is symbolized F. In words, "finishing q" is the same as "not-q, but a moment ago q." In symbols, $\mathrm{F}q = \mathrm{N}q \cdot \mathrm{D}^m(q)$. Using operators, $\mathrm{F} = \mathrm{N} \cdot \mathrm{D}^m$.

These definitions make Eq part of q and Fq part of Nq, and resolve some ambiguous cases where q would otherwise be undefined.

We can see that as m gets smaller and smaller without reaching 0, the states Eq and Fq can still have a value 1. But in the limit, when m equals 0, we have in both cases $q \cdot q'$, which by Boolean algebra equals 0, and by elementary algebra $q \cdot q' = q(1-q) = q-q^2$, but $q^2 = q$ for both 1 and 0, and so $q-q^2 = q-q = 0$ identically.

The Event of Changing from One State to Another State. The event of changing from one state p to another state r is $\mathrm{C}(p, r) = [\mathrm{F}p \text{ AND } \mathrm{E}r] = \mathrm{F}p \cdot \mathrm{E}r$.

The State from One Event to Another Event. A state S may be defined as beginning with one event e and stopping with another event f. There are two cases. In the first case we think of the state S from the first occurrence of e to the first occurrence of f. This is simple and is equal to $\mathrm{H}e \cdot \mathrm{NH}f$; that is, e has happened and f has not happened. In the second case, we think of the state S from any occurrence of e to the next following occurrence of f. We express this as $S(e, f)$, or S. Technically, e equals 1 at times $t_1, t_3, t_5 \ldots$ for a finite number of values, and 0 elsewhere; f similarly equals 1 at times $t_2, t_4, t_6 \ldots$ for a finite number of values and 0 elsewhere. Consider any of $t_1, t_3, t_5 \ldots$ and find

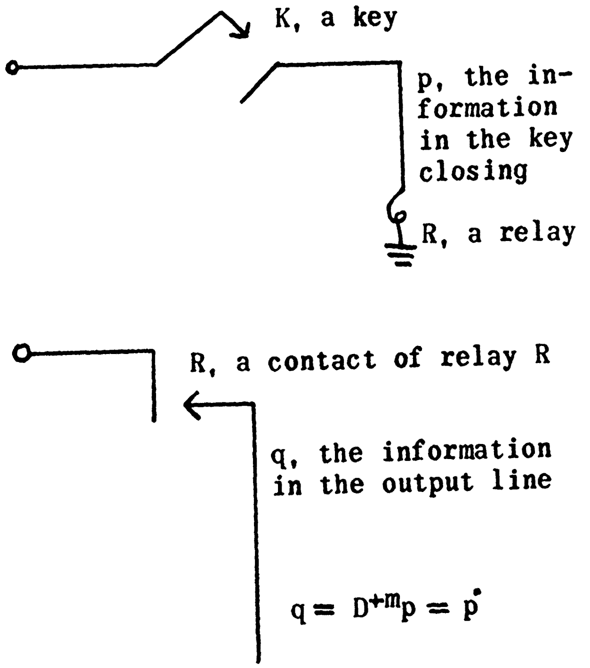


Fig. 9.(a).

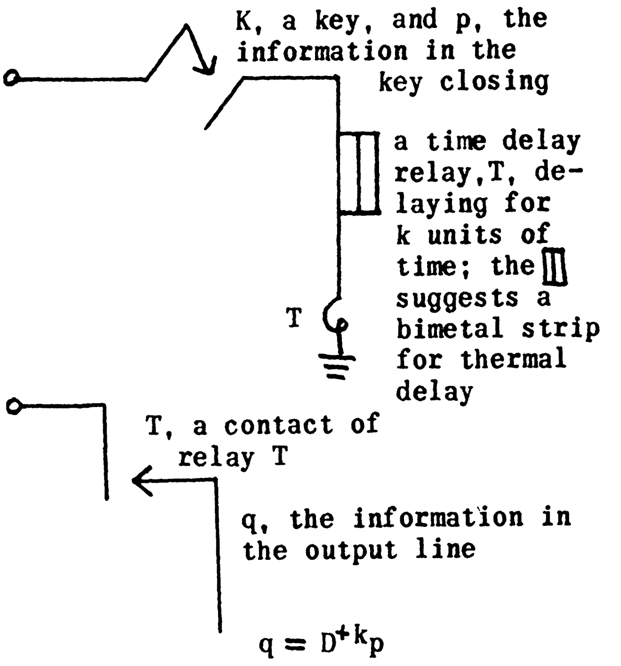


Fig. 9(b).

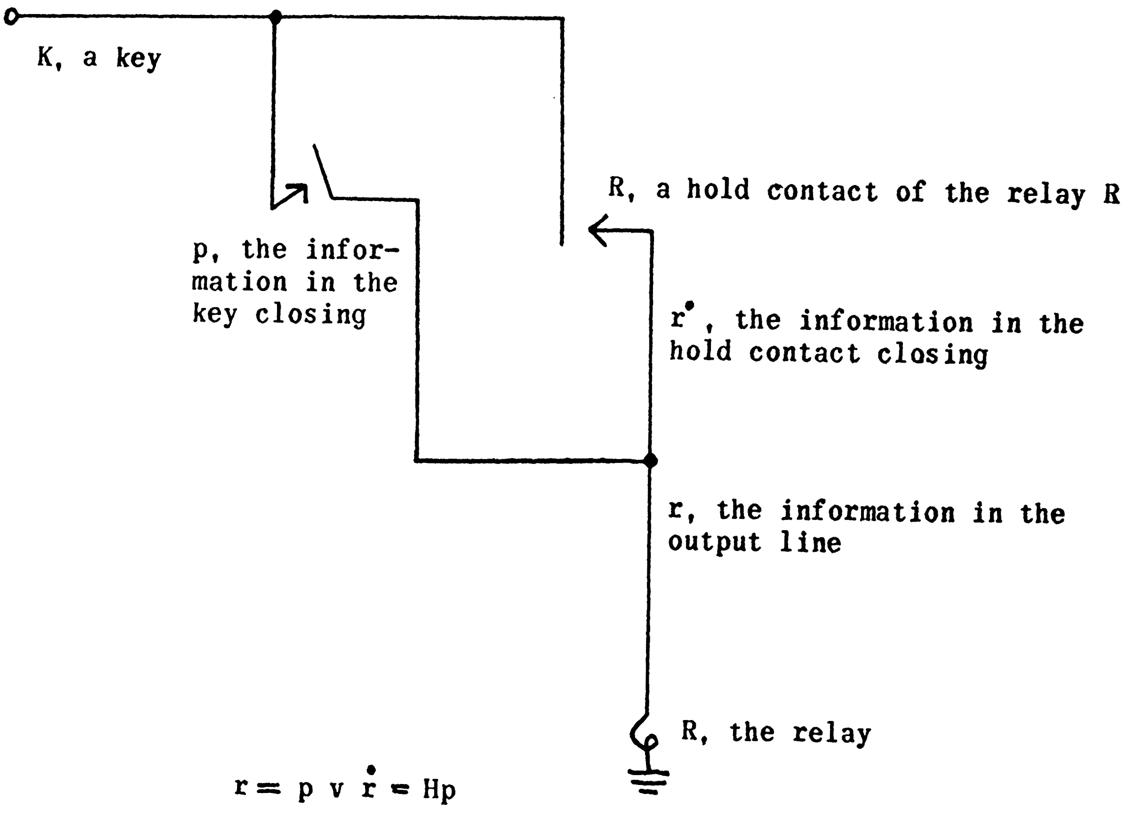


Fig. 9(c).

the value of $t_2, t_4, t_6 \ldots$ which next follows. Suppose they are t_i and t_j respectively. Then the state S equals 1 for every t from t_i to t_j, including t_i but not t_j. This function S has the convenient property:

$$S = \mathrm{N}f \cdot [e \vee \mathrm{D}^m S]$$

In words, the state S is equal to "the event f is not happening, and either the event e is happening, or a moment ago the state S was happening or both." If e and f are both 1 during the same small period of time, S in this period is undefined. If the duration of e is shorter than m, the smallest amount of time sufficient for a circuit element to safely change state, then it is as if e were 0; and the same is true for f.

For the First or Second Time. The operator "is happening for the first time" is symbolized T^1, and

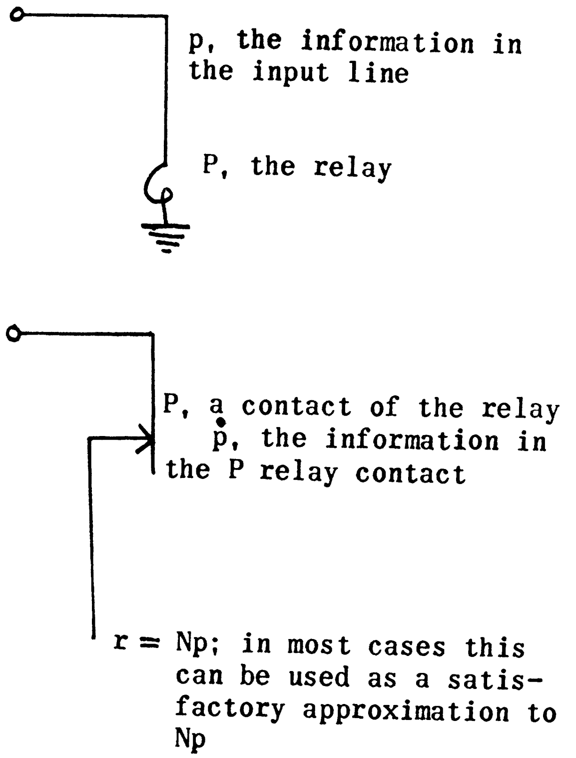


Fig. 9(d).

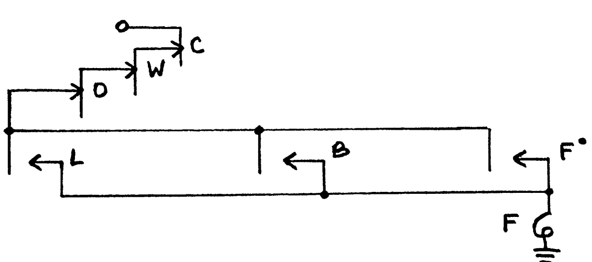


Fig. 10. Circuit satisfying problem 3.

the operator "is happening for the second time" is symbolized T^2. Let p be a binary variable that has more than one stretch. Then we may be interested in distinguishing between the various stretches: $\mathrm{T}^1 p$, the first stretch, "p is happening for the first time"; $\mathrm{T}^2 p$, the second stretch, "p is happening for the second time"; and so on. For example in Table 1, $q = \mathrm{T}^1 p$. There are two cases, corresponding with $p_0 = 0$ and $p_0 = 1$. We can calculate $\mathrm{T}^1 p$ for each case graphically as shown in Fig. 7. We can see that the first case is more general, and includes the second; so we can write:

$$\begin{aligned} \mathrm{T}^1 p &= \mathrm{I} \cdot \mathrm{NH}(\mathrm{N} \cdot \mathrm{H})\,(p) \\ &= [\mathrm{Z} \cdot \mathrm{I} \cdot \mathrm{NH}(\mathrm{N} \cdot \mathrm{H}) \vee \mathrm{NZ}\ \ \mathrm{NHN}]\,(p) \end{aligned}$$

Happened for the Second Time. The operator "has happened for the second time" is symbolized H^2. From the general graphic calculation shown in Fig. 8, we can compute:

$$\mathrm{H}^2 p = \mathrm{H}[\mathrm{I} \cdot \mathrm{H}(\mathrm{N} \cdot \mathrm{H})]\,(p)$$

And we could continue with a large number of the ramifications of the algebra of states and events, but probably enough has been said to show many ways in which it can be further developed.

The main difference between a state and an event is duration. States are long and events are short. But it is a relative difference, not an absolute one; for an event must not be shorter than m, the least time for a circuit element to change state safely, and it is possible for a state to be as short as that. Events, however, are likely to occur without overlapping; that is, their overlapping can ordinarily be neglected. States, on the other hand, are long, and they are likely to occur with overlapping between them, and their overlapping needs to be considered.

Applications of the Algebra of States and Events to On-off Circuit Elements

With the algebra of states and events, we can be both more accurate and more complete in our analysis of the operations of on-off circuit elements. However, we shall consider only one kind of circuit element, relays.

A relay becomes energized and its contacts operate AFTER it is energized by current, instead of—as ordinarily assumed when using Boolean algebra—at the same time. A small amount of time, the "operate-time" of the relay, is needed for a magnetic field to build up in the coil of the relay and to cause its contacts to close. The operate-time of any relay ranges from about 3 milliseconds for a very fast relay, to 20 to 40 milliseconds for ordinary relays, up to a half second or longer for slow-acting relays or time-delay relays.

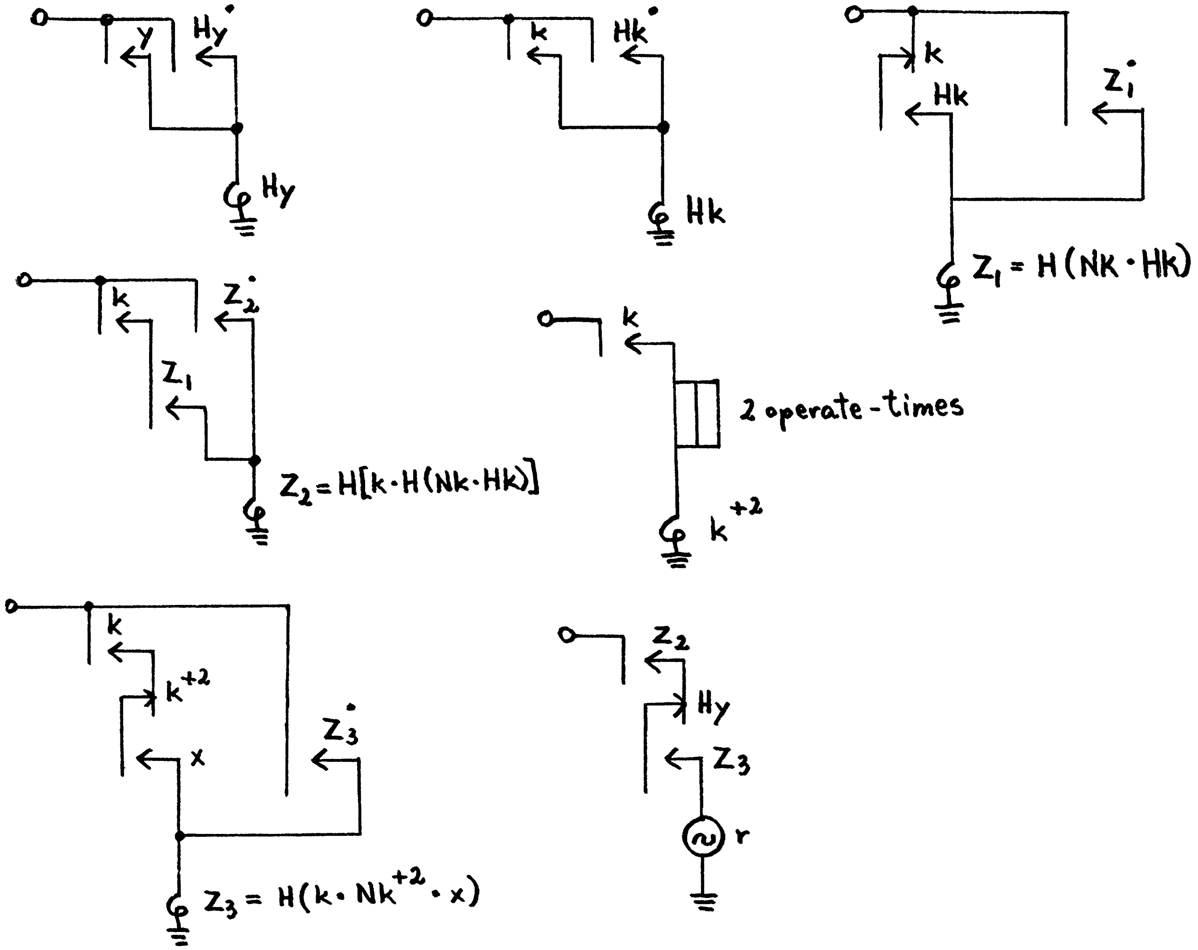


FIG. 11. Circuits satisfying problem 4.

The four chief functions of the new algebra that we desire to represent in the operation of relays are $D^{+m}p$, $D^{+k}p$, Hp and Np. Here m is the safe operate-time of a relay; k is any longer period of time; and the duration of p is not less than m. We show in Fig. 9 circuits which represent these four chief statal algebra functions in relays.

Some Problems and Solutions

Now the chief test of this algebra of states and events is not so much whether it is as yet mathematically entirely rigorous but whether it applies helpfully in solving problems. For mathematical rigor can came more than fifty years later, as it did with ordinary calculus and also with Boole's sketch of his algebra in 1854.

Here then are some samples of problems that this algebra can apply to.

Problem 3. An automatic oil furnace (F) is to run whenever the living room thermostat (L) calls for heat, or when the associated hot water boiler thermostat (B) calls for more hot water. It is to stop whenever the temperature in the chimney (C) is too high, or the water level in the boiler (W) is too low, or when the oil level in the fuel tank (O) is too low. Express F in a relay.

Solution: Let the binary variables of the conditions be as marked by the letters above in the problem. Then, using the formula for the state from one event to another, the statal algebra equation for the circuit is:

$$F = C' \cdot W' \cdot O' \cdot (L \vee B \vee F^{+1})$$

The circuit is shown in Fig. 10.

This problem is typical of the kinds of problems that arise in programming a complex piece of apparatus in order that it may carry out a number of different modes of behavior.

Problem 4. Design a set of relay circuits with a key k and depending on two conditions x and y such that lamp r will light if the key has been pressed twice, y has never occurred, and x was or is on at a time when the key was or is pressed.

Solution:

$$r = H^2(k) \cdot B(y) \cdot H(Ek \cdot x)$$

Now

$$H^2(p) = H[p \cdot H(Np \cdot Hp)],\ B(p) = NHp,\ Ep = p \cdot Np^{+1}$$

therefore

$$r = H[k \cdot H(Nk \cdot Hk)] \cdot NHy \cdot H(k \cdot Nk\cdot \cdot x)$$

We need to discuss $Ek = k \cdot Nk^{+1}$. We desire to pick up a relay while this condition exists. This condition by definition will only last for about the time that a relay can pick up. Hence the operation of the circuit will be uncertain. So let us assume a slower relay for which k^{+2} will apply instead of $k^{+1} = k\cdot$. We will then substitute $Ek = k \cdot Nk^{+2}$. The circuits are shown in Fig. 11.

Much more could be said about this algebra of states and events and its application to other sorts of conditions and on-off circuit elements. But probably enough has been said to show to some extent its usefulness, its power, and its applications.

The discovery of transcendentals, the establishment of the fact that they are far richer in extent and variety than the irrationals of algebra, that they comprise some of the most fundamental magnitudes of modern mathematics—all this showed definitely that the powerful machinery of algebra had failed just where the elementary tools of rational arithmetic had failed two thousand years earlier. Both failures were due to the same source: algebra, like rational arithmetic, dealt with *finite processes* only.

Now as then, infinity was the rock which wrecked the hope to establish number on a firmer foundation. But to legalize infinite processes, to admit these weird irrational creatures on terms of equality with rational numbers, was just as abhorrent to the rigorists of the nineteenth century as it had been to those of classical Greece.

Loud among these rose the voice of Leopold Kronecker, the father of modern *intuitionism*. He rightly traced the trouble to the introduction of irrationals and proposed that they be banished from mathematics. Proclaiming the absolute nature of the integers, he maintained that the natural domain, and the rational domain immediately reducible to it, were the only solid foundation on which mathematics could rest.

"God made the integer, the rest is the work of man," is the famous phrase by which he will be best known to posterity. This phrase reminds me of the story of the pious old dame who was heading a committee for the erection of a new church. The architect who submitted the plans found that the old lady took the business very seriously. Most vehement was her protest against the stained glass for which his specifications called. Finally in despair he asked her on what ground she objected to stained glass. "I want my glass the way the Lord made it!" was her emphatic reply.—Tobias Dantzig, *Number: The Language of Science* (Macmillan, New York, ed. 4, 1954), p. 118.