[1] S. Bochner, "On Compact Complex Manifolds," *J. Indian Math. Soc.*, **11**, 1–21, 1947; "Functions in One Complex Variable as Viewed from the Theory of Functions in Several Variables," in *Lectures on Functions of a Complex Variable* (Ann. Arbor: University of Michigan Press, 1955).

[2] H. Hopf, "Fundamentalgruppe und zweite Bettische Gruppe," *Comm. Math. Helvi.*, **14,** 257–309, 1941.

[3] P. Appell, "Sur les fonctions périodiques de deux variables," *J. math. pures et appl.*, **7,** 157–219 1891.

[4] H. Petersson, "Zur analytischen Theorie der Grenzkreisgruppen," *Math. Ann.*, **115,** 23–67, etc. 1937.

*DIGITAL COMPUTATIONAL METHODS IN SYMBOLIC LOGIC, WITH EXAMPLES IN BIOCHEMISTRY**

By Robert S. Ledley

OPERATIONS RESEARCH OFFICE, JOHNS HOPKINS UNIVERSITY

Communicated by George Gamow, April 9, 1955

INTRODUCTION AND THE DIGITALIZATION

The formal theory of symbolic logic, as developed by Boole,[1] Peirce,[2] Jevons,[3] and Schröder[4] before 1900 and by Whitehead, Russell, Hilbert, Gödel, and others after the turn of the century, is universally recognized for its fundamental importance as a cornerstone of scientific method and thought. Yet it has remained largely a philosophic and esoteric realm of study, and there have existed no methods of actual straightforward computation for direct large-scale applications to specific types of realistic problems. The purpose of this paper is not to present any new results in the formal theory of symbolic logic, but rather to offer a new system of digitalized computational methods in the propositional calculus for application to the great number of practical nonnumerical problems that so frequently occur in science, industry, and government. The object was to formulate a logical "arithmetic" of extreme simplicity, which would provide, for this realm of nonnumerical problems, systematic methods of solution as simple, straightforward, and versatile as those of numerical analysis are for problems of a numerical nature. The theory of Boolean equations[5] becomes a special instance of the more general methods presented in this paper.

It is perhaps not generally realized how wide a range of problems can be attacked and solved by methods of symbolic logic and the related Boolean algebra, which is the calculus of sets and classes. Direct application of logic can always be an aid to deductive reasoning—such as determining consequences of given premises, rules, or axioms and making hypotheses or theorems from which the given premises or factual relationships can be deduced. Besides the use of logical propositional methods in problems concerned with sentences, such as analysis of military information reports and legal and insurance documents, there appear to be even more important applications to fields of operations research, biology, medicine, design of experiments, etc., where the utility of symbolic logic per se is not as immediately evident. Now that the results of the propositional calculus of symbolic logic are made computationally feasible by these new digitalized techniques, it is hoped that they can be directed toward successful solution of many such problems—two examples of problems in biochemistry being given at the end of this paper. The digitalization of the methods in terms of the binary number system provides easy mechanization on existing high-speed electronic computers or on a special electronic digital logic machine.

This paper will contain no proofs, although the foundations of the methods are briefly indicated. Also, it is assumed that the reader is familiar with the methods of symbolic logic[6] and Boolean algebra[7] as given in the elementary reference texts.

REVIEW OF DEFINITIONS AND NOTATION

Propositions are sentences that can be called either *true* or *false* and are symbolized by A_r or X_r. Propositions can be combined by the *propositional operations* "$\cdot$", "$+$", and "$-$", where "$A_r \cdot A_s$" (read "A_r *and* A_s") is true when both A_r and A_s are true; "$A_r + A_s$" (read A_r *or* A_s) is the inclusive *or;* and "$\bar{A}_r$" is the negation of A_r (read "*not* A_r"). (The analogous set or class interpretation of the symbols as *intersection, union,* and *complementation,* respectively, is easily recognized.) An uncombined proposition is called an *elementary element,* while a combination of propositions by the propositional operations is called a *combined element* or *Boolean function,* this latter sometimes being denoted by $f_r(A_1, A_2, \ldots)$.

It is important to distinguish between two different meanings of truth. Some combined elements are true due to the form of the combination, independent of the truth values of the component parts, as, for example, $X_r + \bar{X}_r$ and $(\overline{X_r \cdot X_s}) + (X_r + \bar{X}_s)\cdot(\bar{X}_r + X_s)$. Such elements are called tautologically true, or simply *tautologies,* and are denoted by I. (The intrinsically false element is denoted by O. Hence $\bar{I} = O$.) On the other hand, a certain combined element, not a tautology, may be called true under the circumstances of a problem, this being called *factual* (sometimes *empirical*) truth. (We will not distinguish between the language and the metalanguage but will leave the distinction to be ascertained from the context.)

Two combined elements that are emphasized in all the original as well as in the modern works in logic are $X_r \cdot X_s + \bar{X}_r \cdot \bar{X}_s$ and $\bar{X}_r + X_s$. The former is read "X_r *is equivalent to* X_s" (in truth value), abbreviated by $X_r = X_s$; and when $X_r = X_s$, then $f(X_r, A_1, A_2, \ldots) = f(X_s, A_1, A_2, \ldots)$ for any f. The latter is read "X_r *implies* X_s" or "*if* X_r *then* X_s", abbreviated by $X_r \rightarrow X_s$. Note that this combination is true if X_r and X_s are both false, which might seem contrary to usual usage in language; but the proposition "If you can do that trick, then I'll eat my hat" actually has this meaning. In general, the difficulties in formulating a problem in symbolic logic are no more formidable than the precisely analogous task in mathematics and depend primarily on how well defined the problem is to begin with. However, only the computational aspects of already formulated problems are considered in this paper.

THE BASIS AND THE DIGITALIZATION

To every proposition A_r will be associated a binary number $\#A_r$ called the *designation number* of A_r; in particular, this holds for the elementary elements. The set of designation numbers for the elementary elements is called a *basis,* and one such basis for a system of three elementary elements A_1, A_2, A_3 is:

$$\begin{array}{rcl} & & 87654321 \\ \#A_1 & = & 01010101 \\ \#A_2 & = & 00110011 \\ \#A_3 & = & 00001111 \end{array}$$

where the small numbers above the columns number the positions of the bits. In general, it can be shown that in a system of n elementary elements the designation numbers will have 2^n positions. A basis is distinguished by the fact that the columns represent the 2^n possible combinations of 0, 1 taken n at a time. Thus there are $2^n!$ bases; the basis shown will be used throughout this paper due merely to its visual simplicity of pattern. The relation of a basis to the familiar truth tables is evident. It will be convenient to denote the bit in the ith column and rth row of a basis by $b[A_1, \ldots, A_s]_r^i$. For example, in our basis $b[A_1, A_2, A_3]_2^5 = 1$.

The propositional operations are interpreted in terms of the designation numbers in a manner reminiscent of vector addition as follows, where #A_r^i represents the bit 0 or 1 in the ith position of #A_r:

#$(A_r + A_s)^i$ = #A_r^i + #A_s^i according to the rule $1 + 1 = 1 + 0 = 0 + 1 = 1$, $0 + 0 = 0$;

#$(A_r \cdot A_s)^i$ = #$A_r^i \cdot$ #A_s^i according to the rule $1 \cdot 0 = 0 \cdot 1 = 0 \cdot 0 = 0$, $1 \cdot 1 = 1$;

and #$\bar{A}_r^i = 0, 1$ when #$A_r^i = 1, 0$, respectively.

The operation "+" is called logical addition, "·" logical multiplication, and "−" inversion. Thus the designation number of a combined element with respect to a particular basis is obtained by performing the operations indicated by the combination. For example, with respect to the basis given above, #$(\bar{A}_1 + A_2 \cdot A_3)$ = $10101010 + (00110011) \cdot (00001111) = 10101010 + 00000011 = 10101011$. The set of all the 2^{2^n} possible designation numbers forms a representation of a free Boolean algebra. (Note that for elementary elements, #$A_r^i = b[A_1, \ldots, A_s]_r^i$, of course.)

For the reverse process—namely, given the designation number, to find its propositional representation—note that every product of all the elementary elements or their negations, called an *elementary product*, has a single unit, i.e., #$(A_1 \cdot A_2 \cdot A_3) = 00000001$, #$(\bar{A}_1 \cdot A_2 \cdot A_3) = 00000010$, etc., for each of the 2^3 such products. Hence a propositional representation called the *disjunctive normal form* is obtained by forming the sum of those elementary products corresponding to the units of the designation number. For example, 01100010 = #$(A_1 \cdot \bar{A}_2 \cdot \bar{A}_3 + \bar{A}_1 \cdot A_2 \cdot \bar{A}_3 + \bar{A}_1 \cdot A_2 \cdot A_3)$. Other propositional representations can be systematically generated, such as the simplest-sum-of-products form, simplest-product-of-sums form, and the conjunctive normal form.[8]

Important observations are that #I consists only of units, i.e., #$I^i = 1$ for all i. Hence #$O^i = 0$ for all i. Also, $X_s^i = X_r^i$ if and only if #X_s = #X_r for all i. In addition, $X_r \rightarrow X_s$ if and only if #X_s has units in at least the same positions as does #X_r; i.e., if #$X_r^i = 0$, then #$X_s^i = 0, 1$, but if #$X_r^i = 1$, then #$X_s^i = 1$.

COMPUTATIONAL METHODS

CONSTRAINTS

The concept of constraints is fundamental for the computational methods. Constraints are logical relations between the elementary elements which are to be considered as tautologically true, due to intrinsic or stated circumstances of a problem. In other words, constraints are factually true combined elements which are to remain true throughout a problem and therefore, with respect to this problem, can be considered as tautologies. A systematic computational method will be given that automatically guarantees the maintenance of any desired constraints throughout the solution of a problem. After constraints have been applied, a *constrained propositional calculus* results, which can be interpreted in terms of a nonfree Boolean algebra.

In particular, if H is an automorphism of a Boolean algebra B, such that "+", "•", and "–" are preserved [i.e., for X_r, X_s, ϵB, $H(X_r + X_s) = H(X_r) + H(X_s)$, $H(X_r \cdot X_s) = H(X_r) \cdot H(X_s)$, and $H(\bar{A}_r) = \overline{H(A_r)}$], and if $F(A_1, A_2, \ldots) \epsilon B$ represents a constraint, then we desire that $H(F) = I$. Under this automorphism, equivalence classes are formed by the set $\{F\}$ together with the sets $\{X\}_t$, where $(F + X_r) \epsilon \{F\}$ for all $X_r \epsilon B$, this being the kernel of H, and for $X_s \notin \{F\}$, $X_s \epsilon \{X\}_t$ if $H(X_s \cdot F) = X_t$. The equivalence classes form the elements of the nonfree Boolean *quotient algebra* B/F, when the operations between the classes are defined by $\{X\}_r + \{X\}_s = \{X_r + X_s\}$, $\{X\}_r \cdot \{X\}_s = \{X_r \cdot X_s\}$, and $\overline{\{X_r\}} = \{\bar{X}_r\}$. If $\#F$ has u units, then there will be $2^{(2^n - \mathrm{u})}$ distinct elements in each equivalence class and 2^{u} equivalence classes all together. This quotient algebra automatically satisfies the constraints and represents a constrained propositional calculus.

Given the constraints $F_1, F_2, \ldots, F_r$, the computational method consists in first forming $\prod_{\mu = 1}^{\mu = r} (\#F_\mu)$ and then reducing the reference basis to include only those columns j for which $\prod_{\mu = 1}^{\mu = r} (\#F_\mu^j) = 1$ (i.e., those columns corresponding to the unit positions of the product of the constraints). All computations with respect to this reduced basis will automatically have $\#F_\mu = \#I$ $(\mu = 1, 2, \ldots, r)$, as desired. For example, the constrained basis $b_c[A_1, A_2, A_3]$ that will make A_1, A_2, A_3 mutually exclusive and exhaustive as is often desired in problems concerning classification of subject matter, namely, $A_1 \cdot A_2 = O$, $A_1 \cdot A_3 = O$, $A_2 \cdot A_3 = O$, $A_1 + A_2 + A_3 = I$, becomes

$$\begin{array}{rcl} & & 764 \\ \#A_1 & = & 100 \\ \#A_2 & = & 010 \\ \#A_3 & = & 001 \end{array}$$

For this constraint, A_1, A_2, A_3 are called the components of the three-component proposition A and form an example of an n-component proposition.

ANTECEDENCE AND CONSEQUENCE SOLUTIONS

The majority of logical problems involve given or accepted premises, hypotheses, rules, or other logical relationships which are the *given equations* and essentially comprise the statement of the problem. There are two types of solutions to a set of given equations, the *antecedence solutions* and the *consequence solutions*. Antecedence solutions are hypotheses or theories from which the given equations can be deduced; consequence solutions can be deduced from the given equations. In other words, the truth of the antecedence solutions is sufficient for the truth of the given equations, but the truth of the consequence solutions is necessary for the truth of the given equations. If the given equations are true, then the consequent solutions are true, while the antecedence solutions may or may not be true; but the truth of the given equations can be deduced from the hypotheses embodied in the antecedence solutions, this latter being the method for theory construction.

However, to produce just one antecedence or one consequence solution to given equations is usually trivial; hence logical problems always require solutions of a specified form or solutions involving only certain specified elementary elements or both. Occasionally solutions with the required properties do not exist, and the problem is extended to determine under what conditions such solutions do exist for the given equations. (It is of historical interest to note that the problem of solution to equations as posed by Schröder and others is merely a special case of the general theory presented here, being antecedence solutions of the particular form $X_s = f_s$.)

More specifically, a complete set of antecedence solutions for given equations is any set of constraints F that makes each of the equations true in the constrained system B/F (i.e., that maps each equation into I). All combined elements that become true (are mapped into I) in the constrained system B/E, the constraints being the given set of equations E, are consequence solutions. With these definitions in mind, the problem is summarized in Figure 1. The given equations E_1, $\ldots$, E_r are combined elements or functions of the elementary elements A_1, $\ldots$, A_i, X_1, $\ldots$, X_k. The specified forms F_1, $\ldots$, F_s for the antecedence or consequent solutions are, in general, known functions of X_1, $\ldots$, X_k and of f_1, $\ldots$, f_j, where the f_j are as yet unknown functions of A_1, $\ldots$, A_i. The object is to determine explicitly the functions f_j or, if they do not exist, conditions for their existence. Summarizing, then, computational methods for determining the existence of antecedence or consequence solutions of specified forms and, if they exist, for generating all such solutions, or, if not, for determining conditions of existence, will be presented.

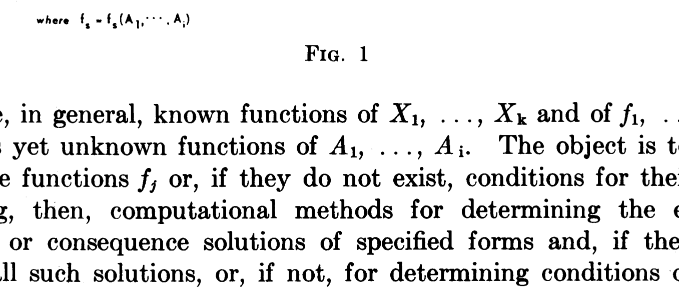


FIG. 1

The fundamental formulas for determining antecedence and consequence solutions are the following:

ANTECEDENCE

$$\boxed{(F_{jk}) \otimes (E_{ki}) = (\bar{R}_{ji})}$$

CONSEQUENCE

$$\boxed{(\bar{F}_{jk}) \otimes (E_{ki}) = (\bar{R}_{ji})}$$

where (F_{jk}), (E_{ki}). (R_{ji}) are Boolean matrices (of elements 0, 1 only), and $\otimes$ represents logical matrix multiplication [i.e., if $(P_{jk}) \otimes (Q_{ki}) = (S_{ji})$, then $\sum_k P_{jk}Q_{ki} = S_{ji}$ with logical multiplication and addition]. The "$-$" represents inversion of each individual element of the matrix. The formation of (E_{ki}) and (F_{jk}), and the interpretation of (R_{ji}) as the desired solutions, proceed as follows.

The Formation of Matrices (E_{ki}) *and* (F_{jk}).—Two bases enter into this calculation namely, $b[A_1, \ldots, A_{\mathrm{i}}, X_1, \ldots, X_{\mathrm{k}}]$ and $b[f_1, \ldots, f_{\mathrm{j}}, X_1, \ldots, X_{\mathrm{k}}]$, where the f_j's are considered as elementary elements and both bases have the usual pattern, the last rows being formed by the X_k's. Then form $\prod_{\mu=1}^{\mu=\mathrm{r}} (\#E_\mu)$ with respect to $b[A_1, \ldots, A_{\mathrm{i}}, X_1, \ldots, X_{\mathrm{k}}]$ and $\prod_{\nu=1}^{\nu=\mathrm{s}} (\#F_\nu)$ with respect to $b[f_1, \ldots, f_{\mathrm{j}}, X_1, \ldots, X_{\mathrm{k}}]$. Now separate the positions of the designation number $\prod_{\mu=1}^{\mu=\mathrm{r}} (\#E_\mu)$ into 2^{k} successive groups of positions, with 2^{i} positions in each group. Index the groups from right to left by k $(k = 1, \ldots, 2^{\mathrm{k}})$ and the positions in each group by i $(i = 1, \ldots, 2^{\mathrm{i}})$. The rows of the matrix (E_{ki}) are simply these respective groups of positions. Similarly, $\prod_{\nu=1}^{\nu=\mathrm{s}} (\#F_\nu)$ is separated into 2^{k} groups, this time of 2^{j} positions, the groups being indexed by k and the positions in each group by j $(j = 1, \ldots, 2^{\mathrm{j}})$. The columns of the matrix (F_{jk}) are simply these respective groups of positions.

In more mathematical terms, the formula for an element E_{ki} of the matrix (E_{ki}) is $E_{ki} = \prod_{\mu=1}^{\mu=\mathrm{r}} (\#E_\mu^p)$ for $(p - 1) = (k - 1)2^{\mathrm{i}} + (i - 1)$. Similarly, $(F_{jk}) = \prod_{\mu=1}^{\mu=\mathrm{r}} (\#F_\mu^q)$ for $(q - 1) = (k - 1)\, 2^{\mathrm{j}} + (j - 1)$. (Note, of course, that $\prod_{\mu=1}^{\mu=\mathrm{r}} (\#E_\mu^p) = \#(\prod_{\mu=1}^{\mu=\mathrm{r}} E_\mu)^p$ and similarly for $\prod_{\nu=1}^{\nu=\mathrm{s}} (\#F_\nu^q)$).

The Interpretation of (R_{ji}).—Two bases are consulted in this process, namely, $b[f_1, \ldots, f_{\mathrm{j}}]$ and $b[A_1, \ldots, A_{\mathrm{i}}]$, both written in the usual pattern. The desired solution is computed by means of the result array, which consists of an as yet empty array with 2^{i} columns and j rows; the columns are indexed by i from right to left, the rows corresponding to $f_1, \ldots, f_{\mathrm{j}}$ assigned from top to bottom. Now, having computed (R_{ji}) by the proper formula, consider those pairs of indices j, i, for which $R_{ji} = 1$: place the jth column of $b[f_1, \ldots, f_{\mathrm{j}}]$ in the ith column of the result array. The rows of the result array thus filled are the designation numbers of the corresponding $f_1, \ldots, f_{\mathrm{j}}$ with respect to the basis $b[A_1, \ldots, A_{\mathrm{i}}]$ and hence give the desired explicit functions $f_1(A_1, \ldots, A_{\mathrm{i}}), \ldots, f_{\mathrm{j}}(A_1, \ldots, A_{\mathrm{i}})$.

In more mathematical terms, $\#f_m(A_1, \ldots, A_{\mathrm{i}})^i = b[f_1, \ldots, f_{\mathrm{j}}]_m^j$ for those j for which $R_{ji} = 1$, and $m = 1, \ldots, \mathrm{j}$, where $\#f_m(A_1, \ldots, A_{\mathrm{i}})$ is interpreted with respect to $b[A_1, \ldots, A_{\mathrm{i}}]$.

Multiple Sets of Solutions.—The occurrence of several columns of $b[f_1, \ldots, f_{\mathrm{j}}]$ in a single result-array column indicates multiple solutions, for each such column determines a different set of solutions. Hence the total number of different sets of solutions is the product of the number of columns of $b[f_1, \ldots, f_{\mathrm{j}}]$ in each array column. If a column of the result array has no basis column in it, then no solutions exist at all.

Conditions for the Existence of Solutions.—Consider the row vector (V_j) comprising 2^{j} units. If no solution exists, compute the row vector (C_i) from the formula $(V_j) \otimes (R_{ji}) = (C_i)$. Then (C_i) is the designation number of the desired condition for the existence of solutions, as referred to $b[A_1, \ldots, A_{\mathrm{i}}]$. In order to impose this condition new bases constrained by the condition must now replace $b[A_1, \ldots,$

$A_i, X_1, \ldots, X_k]$ and $b[A_1, \ldots, A_i]$, and the solution is computed exactly as above, except that for "2^i" now read "u", where u is the number of units in (C_i).

A Simple Example.—Consider the especially simple equation of historical interest discussed at length by Schröder, namely, $A_1X + A_2\bar{X} = O$, to be solved for antecedence solutions of the form $X = f(A_1, A_2)$. The computations proceed as follows:

$$b[f, X]: \qquad \begin{array}{rcl} \#f & = & 0101 \\ \#X & = & 0011 \\ \hline \#(f = X) & = & 1001 \\ \begin{smallmatrix} j \\ k \end{smallmatrix} & & \begin{smallmatrix} 2\,1\,2\,1 \\ 2\;\;\;1 \end{smallmatrix} \end{array} \qquad\qquad b[A_1, A_2, X]: \qquad \begin{array}{rcl} \#A_1 & = & 01010101 \\ \#A_2 & = & 00110011 \\ \#X & = & 00001111 \\ \hline \#(A_1X + A_2\bar{X} = O) & = & 11001010 \\ \begin{smallmatrix} i \\ k \end{smallmatrix} & & \begin{smallmatrix} 4\,3\,2\,1\,4\,3\,2\,1 \\ 2\;\;\;\;\;\;\;\;1 \end{smallmatrix} \end{array}$$

whence $(F_{jk}) = \binom{10}{01}$ and $(E_{ki}) = \binom{1100}{1010}$. Substituting in the antecedence formulas, $(F_{jk}) \otimes (\bar{E}_{ki}) = \binom{10}{01} \otimes \binom{0011}{0101} = \binom{0011}{0101} = (\bar{R}_{ji})$, whence $(R_{ji}) = \binom{1100}{1010}$. To interpret (R_{ji}),

$$b[f]: \quad \begin{array}{rcl} j & & 2\,1 \\ \hline \#f & = & 01 \end{array}$$

Result array:

i	4	3	2	1
$\#f$	0, 1	0	1	none

Thus no solution exists. The condition for the existence of a solution is computed by $(V_j) \otimes (R_{ji}) = (11) \otimes \binom{1100}{1010} = (1110)$ to be interpreted with respect to $b[A_1, A_2]$: $\#A_1 = 0101$, $\#A_2 = 0011$, whence $1110 = \#(\bar{A}_1 + \bar{A}_2)$, i.e., $\bar{A}_1 + \bar{A}_2 = I$ in order that $A_1X + A_2\bar{X} = O$ have a solution.

The constrained basis $b[A_1, A_2, X]$ becomes $\#A_1 = 010\ 010$, $\#A_2 = 001\ 001$, $\#X = 000\ 111$, whence $\#(A_1X + A_2\bar{X} = O) = 110101$ and $(E_{ki}) = \binom{110}{101}$. Then $\#(R_{ji}) = \left(\frac{110}{101}\right)$, and the result array is

i	3	2	1
$\#f$	0, 1	0	1

which, interpreted in terms of the constrained bases $b_c[A_1, A_2]$: $\#A_1 = 010$, $\#A_2 = 001$, is $\#f = 001 = \#A_2$ and $\#f = 101 = \#\bar{A}_1$. Hence, under the proper conditions, the given equation has two solutions, $X = A_2$ and $X = \bar{A}_1$.

Finding Antecedence and Consequence Solutions Involving Only Specified Elementary Elements.—Let $A_1, \ldots, A_i$ be the specified elementary elements, all the rest being included in the $X_1, \ldots, X_k$. Then all the computations proceed as above. For example, consider the problem of determining consequence solutions of $A_1X + A_2\bar{X} = O$ involving only A_1 and A_2. As above, $(E_{ji}) = \binom{1100}{1010}$, but the desired form is simply f, and so, since $\#f = 0101$ as referred to bases $b[f, X]$, $(F_{jk}) = \binom{00}{11}$. Thus $(\bar{F}_{jk}) \otimes (E_{ki}) = \binom{11}{00} \otimes \binom{1100}{1010} = \binom{1110}{0000} = (\bar{R}_{ji})$. Therefore, $(R_{ji}) = \binom{0001}{1111}$, whence

$$\begin{array}{rcl} j & & 2\,1 \\ \hline \#f & = & 01 \end{array}$$

i	4	3	2	1
$\#f$	1	1	1	0, 1

or $f = \bar{A}_1 + \bar{A}_2$ and $f = I$. In other words, $\bar{A}_1 + \bar{A}_2 = I$ or, equivalently, $A_1 \cdot A_2 = O$ is the desired nontrivial solution.

Special Theorems.—Solutions of the form $f_m = X_m$, $m = 1, \ldots, k$ are particularly important and occur frequently. For antecedence solutions of this form, (F_{jk})

is the identity matrix, and hence $(E_{kt}) = (R_{jt})$ (as was seen in the above example), enabling (R_{jt}) to be written down directly. In addition, it can be shown that if, to a set of given equations, both antecedence solutions and consequence solutions of this form exist, then there is only one such set of solutions (i.e., no multiple solutions), and it is the same for both cases. Also, if only one consequence (antecedence) solution of this form exists, then it is also an antecedence (consequence) solution of this form.

LOGICAL DEPENDENCE (CONDITIONS) AND LOGICAL INDEPENDENCE

Conditions on the Solutions.—Often it is desired to find only those solutions $f_1, \ldots, f_j$ which satisfy given conditions or constraints $G(f_1, \ldots, f_j)$ or, in other words, only those $f_1, \ldots, f_j$ for which the given combined element or function $G(f_1, \ldots, f_j) = I$. The computational procedure is similar to calculating solutions with conditions on the $A_1, \ldots, A_i$, except that now bases constrained by $G(f_1, \ldots, f_j)$ replace $b[f_1, \ldots, f_j, X_1, \ldots, X_k]$, and $b[f_1, \ldots, f_j]$ (where $\#G(f_1, \ldots, f_j)$ is determined with respect to $b[f_1, \ldots, f_j]$); and for "2^j" now read "u", where u is the number of units in $\#G(f_1, \ldots, f_j)$. All sets of solutions thus calculated will automatically satisfy the given conditions.

Logically Independent Solutions.—Since $f_1 = f_1(A_1, \ldots, A_i), \ldots, f_j = f_j(A_1, \ldots, A_i)$, it can happen that these equations imply a logical relation between $f_1, \ldots, f_j$. For example, if $f_1 = A_1 \cdot A_2$ and $f_2 = A_1 + A_2$, then $f_1 \rightarrow f_2$. However, often a prime requirement of sets of solutions for a problem is that they be logically independent, that is, no relations of the form $G(f_1, \ldots, f_j) = I$ exist other than tautologies. Then $f_1, \ldots, f_j$ will be logically independent when the result array has at least 2^j different columns. Three cases arise: $j = i$, $j < i$, $j > i$. If $j = i$, then there can be no repeated columns in the result array for independent solutions. For $j < i$, 2^j different columns must appear in the array for independence. For $j > i$ there can be no independent set of solutions.

Determining the Logical Dependence between a Given Set of Solutions.—Given a set of solutions, it is often desired to find the logical dependence between them, if any exists at all. The computational method is to consider the result array for this given set of solutions $f_1, \ldots, f_j$. Rewrite the array, omitting any duplicate columns, and adjoin to this (on the right) those columns necessary to make the completed array a basis, say $b'[f_1, \ldots, f_j]$, not necessarily of the usual pattern. Then the designation number with zeros in the positions corresponding to the adjoined columns, units elsewhere, produces the desired conditions between $f_1, \ldots, f_j$ when referred to $b'[f_1, \ldots, f_j]$.

EXAMPLES IN BIOCHEMISTRY

ENZYME CHEMISTRY

In biochemistry an enzyme often cannot be isolated easily, and therefore the experiments performed often will involve several other enzymes. Consequently a combination of reactions must be observed. In such a complex situation the ordinary simple logical analysis usually used in experimental science is found inadequate. In these cases the logical computational methods presented above can be extremely useful for evaluating experimental results as well as for planning future experiments to yield the maximum information.

Suppose that a chemist were studying enzymes, A_1, A_2, A_3 in relation to reactions X_1, X_2, X_3. Suppose that he has done the following four experiments: (E_1)—A solution containing none of A_1, A_2, A_3 produced reaction X_2 but not X_1, and not X_3. (E_2)—The solution contained A_1 and either A_2 or A_3, or both; he could not be sure. The reaction was neither X_2 nor X_1 and X_3 together. (E_3)—The solution had A_2 but not A_1, or did not have A_2 but had A_3. Reactions X_1 and X_2 occurred, or reaction X_3 occurred but X_1 did not. (E_4)—The solution was obtained from a source that had A_3 and A_1 or A_2 or both, or else had neither A_1 nor A_3; the solution turns color, which means X_1 does not take place or both X_2 and X_3 do.

The antecedence problem is: What theories about the enzymes associated with each reaction will explain the experimental results? The consequence problem is: What combinations of enzymes are necessary for each reaction to take place?

The experiments and their designation numbers with respect to the usual basis are as follows:

$$E_1: \ \bar{A}_1 \cdot \bar{A}_2 \cdot \bar{A}_3 = \bar{X}_1 \cdot X_2 \cdot \bar{X}_3$$

0111 1111 0111 1111 1000 0000 0111 1111 0111 1111 0111 1111 0111 1111 0111 1111

$$E_2: \ A_1 \cdot (A_2 + A_3) = \bar{X}_2 \cdot (\overline{X_1 \cdot X_3})$$

0001 0101 0001 0101 1110 1010 1110 1010 0001 0101 1110 1010 1110 1010 1110 1010

$$E_3: \ \bar{A}_1 \cdot A_2 + \bar{A}_2 \cdot A_3 = X_1 \cdot X_2 + \bar{X}_1 \cdot X_3$$

1101 0001 1101 0001 1101 0001 0010 1110 0010 1110 1101 0001 0010 1110 0010 1110

$$E_4: \ A_3 \cdot (A_1 + A_2) + \bar{A}_1 \cdot \bar{A}_3 = \bar{X}_1 + X_2 \cdot X_3$$

1010 0111 0101 1000 1010 0111 0101 1000 1010 0111 0101 1000 1010 0111 1010 0111

$$\text{Thus} \prod_{\mu=1}^{\mu=4} (\#E_\mu) = \text{0000 0001 0001 0000 1000 0000 0000 1000 0000 0100 0100 0000 0010 0010 0010 0010}$$

For the antecedence problem the form of the solution is $X_1 = f_1$, $X_2 = f_2$, $X_3 = f_3$. Recall for antecedence solutions of this special form that $(E_{ki}) = (R_{ji})$.

Hence

$$(E_{ki}) = (R_{ji}) = \begin{array}{c} \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array} \begin{array}{cc} 8765 & 4321 \\ \left(\begin{array}{cc} 0000 & 0001 \\ 0001 & 0000 \\ 1000 & 0000 \\ 0000 & 1000 \\ 0000 & 0100 \\ 0100 & 0000 \\ 0010 & 0010 \\ 0010 & 0010 \end{array}\right) \end{array}$$

from which the solutions are read:

$b[f_1, f_2, f_3]$:

j	87654321
$\#f_1 =$	01010101
$\#f_2 =$	00110011
$\#f_3 =$	00001111

Result array:

i	8	7	6	5	4	3	2	1
$\#f_1$	0	1	1, 0	1	1	0	1, 0	0
$\#f_2$	1	0	1, 1	0	1	0	1, 1	0
$\#f_3$	0	1	1, 1	0	0	1	1, 1	0

Hence there are four sets of antecedence solutions of the form $X_s = f_s$, namely,

$$\begin{array}{llll}
X_1 = f_1, \text{ where } f_1 = A_1\cdot\bar{A}_3 + \bar{A}_1\cdot A_3, & \bar{A}_3\cdot(A_1 + A_2) + \bar{A}_1\cdot\bar{A}_2\cdot A_3, & & \\
X_2 = f_2, \text{ where } f_2 = \bar{A}_1 \qquad\qquad\quad , & \bar{A}_1 & , & \\
X_3 = f_3, \text{ where } f_3 = A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2, & A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2 & , & \\
 & \bar{A}_3\cdot(A_1 + A_2) + \bar{A}_1\cdot A_3, & A_1\cdot\bar{A}_3 + \bar{A}_1\cdot\bar{A}_2\cdot A_3; & \\
 & \bar{A}_1 \qquad\qquad\qquad\quad , & \bar{A}_1; & \\
 & A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2 \qquad , & A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2. &
\end{array}$$

The first two sets are logically independent. The last two sets of solutions imply that the reactions are not independent but that $X_2\cdot X_3 \rightarrow X_1$, and $X_1\cdot X_2\cdot X_3 = O$, respectively. For example, we write the designation numbers of the last solution, cross off the repeating column, adjoin the missing column and obtain:

$$\begin{array}{rcll}
\#(A_1\cdot\bar{A}_3 + \bar{A}_1\cdot\bar{A}_2\cdot A_3) & = & 0101 & 10\cancel{0}0\,\vdots\,1, \\
\#\bar{A}_1 & = & 1010 & 10\cancel{1}0\,\vdots\,1, \\
\#(A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2) & = & 0110 & 01\cancel{1}0\,\vdots\,1, \\
 & & 1111 & 11\ \ 1\ 0 = \#(f_1\cdot f_2\cdot f_3 = O), \text{ or } X_1\cdot X_2\cdot X_3 = O.
\end{array}$$

For the consequence problem, the form of the solution and corresponding designation numbers with respect to $b[f_1, f_2, f_3, X_1, X_2, X_3]$ are as follows:

$$\begin{array}{ll}
\#(X_1 \rightarrow f_1) & 1111\ 1111\ 0101\ 0101\ 1111\ 1111\ 0101\ 0101\ 1111\ 1111\ 0101\ 0101\ 1111\ 1111\ 0101\ 0101 \\
\#(X_2 \rightarrow f_2) & 1111\ 1111\ 1111\ 1111\ 0011\ 0011\ 0011\ 0011\ 1111\ 1111\ 1111\ 1111\ 0011\ 0011\ 0011\ 0011 \\
\#(X_3 \rightarrow f_3) & 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 0000\ 1111\ 0000\ 1111\ 0000\ 1111\ 0000\ 1111 \\
\prod_{\nu = 1}^{\nu = 4} (\#F_\nu) = & 1111\ 1111\ 0101\ 0101\ 0011\ 0011\ 0001\ 0001\ 0000\ 1111\ 0000\ 0101\ 0000\ 0011\ 0000\ 0001
\end{array}$$

hence

$$(F_{jk}) = \begin{pmatrix} 1000 & 0000 \\ 1100 & 0000 \\ 1010 & 0000 \\ 1111 & 0000 \\ 1000 & 1000 \\ 1100 & 1100 \\ 1010 & 1010 \\ 1111 & 1111 \end{pmatrix} \text{ and } (\bar{F}_{jk}) \otimes (E_{ki}) = (\bar{R}_{ji}), \text{ whence } (R_{ji}) = \begin{pmatrix} 0000 & 0001 \\ 0001 & 0001 \\ 1000 & 0001 \\ 1001 & 1001 \\ 0000 & 0101 \\ 0101 & 0101 \\ 1000 & 0101 \\ 1111 & 1111 \end{pmatrix},$$

whence the result array:

i	8	7	6	5	4	3	2	1
$\#f_1$	1010	11	1	1111	11	1010	1	ϕ
$\#f_2$	1111	10	1	1010	11	1100	1	ϕ
$\#f_3$	1100	11	1	1100	10	1111	1	ϕ

Since the most significant set of solutions for this form is the minimum set, i.e., the solutions with the most zeros, we choose

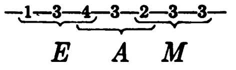
$$\begin{array}{lll}
\#f_1 = 0111\ 1010 = \#[\bar{A}_3\cdot(A_1 + A_2) + \bar{A}_1\cdot A_3], & & X_1 \rightarrow \bar{A}_3\cdot(A_1 + A_2) + \bar{A}_1\cdot A_3; \\
\#f_2 = 1010\ 1010 = \#\bar{A}_1, & \text{or} & X_2 \rightarrow \bar{A}_1; \\
\#f_3 = 0110\ 0110 = \#[A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2], & & X_3 \rightarrow A_1\cdot\bar{A}_2 + \bar{A}_1\cdot A_2.
\end{array}$$

THE PROTEIN DECODING PROBLEM

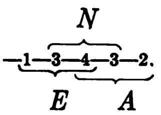
According to present views,[9] hereditary informations are stored in chromosomes in the form of long sequences of four different units (known as bases) constituting the molecules of nucleic acid. In the process of protein synthesis, these polynucleotide sequences must be translated into the corresponding polypeptide sequences of twenty different amino acids. Thus there must exist a mechanism which carries through such translations in a unique way, producing a long word written in a twenty-letter alphabet for each long number written in the four-digital system. Since twenty different triplets can be formed (with disregard to order) out of four different elements, it is inviting to associate each amino acid in the resulting polypeptide chain with a group of three bases in the original polynucleotide sequence. One can make different assumptions as to the type of correlation involved. For example, we can assume that while one amino acid is defined by three neighboring bases, its neighbor is given by the next three bases:

$$-\underbrace{1-3-4}_{E}-\underbrace{3-2-3}_{M}-.$$

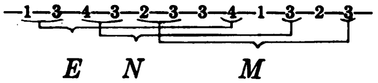
Or one can make the assumption of overlapping triplets, in which case each base is participating in the choice of more than one amino acid. For example, we may have the scheme



or the more restrictive scheme



The possibility is also not excluded that because of twisting of the chain, nonneighboring bases participate in the choice, as in the following:



In all overlapping translation codes, one should expect a stronger or weaker intersymbol correlation, i.e., some restriction in the choice of neighboring amino acids. Thus, comparing these restrictions with actually observed sequences of amino acids in protein chains, one should be able to find the correct coding procedure and the one-to-one correspondence between the twenty amino acids and the twenty hypothetical triplets of bases. Since there are $20! = 2.3 \times 10^{17}$ such one-to-one correspondences (which equals the number of seconds in the age of the universe), the straightforward test of all possibilities is out of the question, and it is desirable to find a systematic and feasible procedure for solving the problem (with the aid of high-speed electronic computing machines).

Use of the Logical Computational Methods Developed Above.—To illustrate the method of solution, it is best to consider a much simpler case which makes more transparent the computational methods involved and can easily be extended to the complex problem given above. Suppose that there were only three kinds of amino acids and only three types of base triplets allowed. Let the rules of combination involve groups of three consecutive triplets, and let the first position of the group be denoted by A, the second by B, and the third by C. In addition, the subscript 1, 2, or 3 will denote which one of the three allowed triplets is in the position, i.e., A_2 stands for "The first position has triplet 2". Hence A_s, B_s, and C_s, for $s = 1$, 2, 3, are each three component propositions. Similarly, let X_r, Y_r, Z_r, where $r = 1$, 2, 3, represent the positions of one of the three amino acids, considered in consecutive groups of three. Let the hypothetical conditions on the sequences of triplets be:

$$\begin{array}{ll}
A_1 \cdot B_2 \rightarrow C_3 + C_2 & \qquad A_1 \cdot B_2 \rightarrow C_1 + C_2 \\
A_1 \cdot B_3 \rightarrow C_2 + C_3 & \qquad A_1 \cdot B_3 \rightarrow C_1 + C_3 \\
A_2 \cdot B_1 \rightarrow C_1 + C_3 & \qquad A_2 \cdot B_1 \rightarrow C_1 + C_2 \\
A_2 \cdot B_3 \rightarrow C_1 + C_3 & \qquad A_2 \cdot B_2 \rightarrow C_1 + C_2 \\
A_3 \cdot B_1 \rightarrow C_1 + C_2 & \qquad A_2 \cdot B_3 \rightarrow C_2 + C_3 \\
A_3 \cdot B_2 \rightarrow C_1 + C_2 & \qquad A_3 \cdot B_2 \rightarrow C_2 + C_3 \\
A_1 \cdot B_1 \rightarrow C_1 + C_3 & \qquad A_3 \cdot B_3 \rightarrow C_1 + C_3
\end{array}$$

Suppose that the experimental results gave the following conditions on the amino acids:

$$X_3 \cdot Y_3 \rightarrow Z_2 + Z_3, \qquad X_1 \cdot Y_1 \rightarrow Z_1 + Z_2, \qquad X_2 \cdot Y_2 \rightarrow Z_1 + Z_2.$$

The problem is to find antecedence solutions of the form $X_r = f_r$, $Y_r = g_r$, and $Z_r = h_r$ for $r = 1, 2, 3$. Introducing the following shorthand: 1 represents the column $\begin{smallmatrix}1\\0\\0\end{smallmatrix}$, 2 the column $\begin{smallmatrix}0\\1\\0\end{smallmatrix}$, and 3 the column $\begin{smallmatrix}0\\0\\1\end{smallmatrix}$, since A_s, B_s, C_s (and X_r, Y_r, Z_r) are three component propositions, find for $b[A_s, B_s, C_s]$ (or $b[f_r, g_r, h_r]$):

i (or j):	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
$\#A_s$ (or $\#f_r$) =	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
$\#B_s$ (or $\#g_r$) =	1	1	1	2	2	2	3	3	3	1	1	1	2	2	2	3	3	3	1	1	1	2	2	2	3	3	3
$\#C_s$ (or $\#h_r$) =	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3

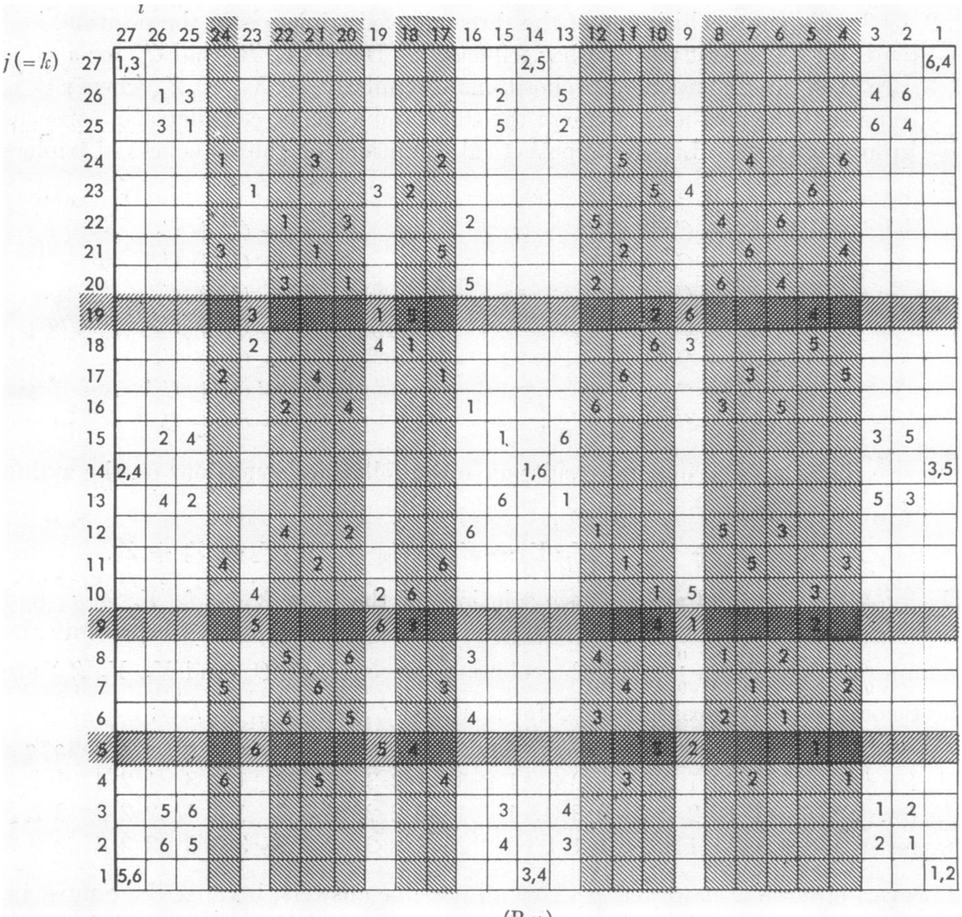
However, observe that we really know all possible answers, because there must be a one-to-one correspondence between the components of X_r and those of A_s; and, in addition, the same correspondence must hold between Y_r and B_s, and Z_r and C_s. Hence the 3! possible solutions are as follows:

Solution number:		1	2	3	4	5	6
X_1	=	A_1,	A_2,	A_1,	A_3,	A_2,	A_3
X_2	=	A_2,	A_1,	A_3,	A_1,	A_3,	A_2
X_3	=	A_3,	A_3,	A_2,	A_2,	A_1,	A_1

Letting $P = 121233$, $Q = 213312$, $R = 332121$, the result array for the solutions with respect to $b[A_s, B_s, C_s]$ can be written as follows:

i	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
$\#f_r$	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R
$\#g_r$	P	P	P	Q	Q	Q	R	R	R	P	P	P	Q	Q	Q	R	R	R	P	P	P	Q	Q	Q	R	R	R
$\#h_r$	P	P	P	P	P	P	P	P	P	Q	Q	Q	Q	Q	Q	Q	Q	Q	R	R	R	R	R	R	R	R	R

where this is to be interpreted as 3! = 6 solutions (*not* 6^{27} solutions according to the notation used above), where each solution corresponds to the same respective column for $i = 1, \ldots, 27$. Hence (R_{ji}) can be constructed, where we put the number of each solution in the proper R_{ji} element. Thus we obtain Figure 2.



j(=k) \ i	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
27	1,3													2,5													6,4
26		1	3										2		5										4	6	
25		3	1										5		2										6	4	
24				1			3				2						5				4			6			
23					1				3	2								5	4				6				
22						1		3				2				5				4		6					
21				3			1				5						2				6			4			
20						3		1				5				2				6		4					
19					3				1	5								2	6				4				
18					2				4	1								6	3				5				
17				2			4				1						6				3			5			
16						2		4				1				6				3		5					
15		2	4										1.		6										3	5	
14	2,4													1,6													3,5
13		4	2										6		1										5	3	
12						4		2				6				1				5		3					
11				4			2				6						1				5			3			
10					4				2	6								1	5				3				
9					5				6	[illegible]								4	1				2				
8						5		6				3				4				1		2					
7				5			6				3						4				1			2			
6						6		5				4				3				2		1					
5					6				5	4								3	2				1				
4				6			5				4						3				2			1			
3		5	6										3		4										1	2	
2		6	5										4		3										2	1	
1	5,6													3,4													1,2

(R_{ji})

FIG. 2

However, this matrix (R_{ji}) would be the result with no conditions on A_s, B_s, C_s, X_r, Y_r, Z_r. Adding the conditions to $b[A_s, B_s, C_s, X_r, Y_r, Z_r]$ amounts to eliminating some columns (corresponding to the zeros of the conditions on A_s, B_s, C_s) and some rows (corresponding to zeros of those on X_r, Y_r, Z_r) as indicated in Figure 2. Now only those solutions remain that are contributed to by *every* column. Hence, solution 1 is eliminated, for it does not appear in column 19 or column 9, and so forth. Only solution 4 is not eliminated and hence, under the given and experimental conditions, it is the only valid hypothesis.

Now we are able to sketch briefly the feasible method of solving this problem by means of computers.[10] First, there is a method for putting the $n!$ permutations of $1, \ldots, n$ into one-to-one correspondence with $1, \ldots, n!$, so that the correspondence can be easily accomplished in either direction. Thus these integers will number the solutions uniquely. Second, there is also a systematic procedure for determining the solution number, given i and j ($= k$). So choose some allowed column i_0, and find the solution numbers for all j that have not been eliminated due to constraints. Solution numbers not found by this step cannot be valid solutions, for they do not appear in this column i_0. Third, there is a procedure for determining j, given i and the solution number. So for all other allowed columns i, and for each solution number just obtained, try to determine j. Those solutions for which no allowed j can be found for every allowed i cannot be valid solutions. The remaining solutions are valid.

Of course, for $3! = 6$ solutions it is easiest to try them all. However, the feasible method just presented is intended for choosing one out of the 20! possible solutions of the original problem. The (R_{ji}) matrix in this case is $8{,}000 \times 8{,}000$, but, when a few hundred conditions are applied to A_s, B_s, C_s, X_r, Y_r, Z_r, it will reduce to about $5{,}000 \times 5{,}000$. It is estimated that the complete calculation should take a computer no more than a hundred hours. On the other hand, to try 20! solutions, a computer put to work in the days of the Roman Empire, at a rate of one million solutions per second, 24 hours a day, all year round, would not yet be close to finishing the job.

The author gratefully acknowledges the assistance and guidance of Dr. George Gamow, particularly with respect to the general description of the Protein Decoding Problem. The kind encouragement of Dr. Nicholas M. Smith and the Operations Research Office is especially appreciated.

[*] This paper is an excerpt of a technical memorandum which is being published by the Operations Research Office and is part of an investigation of methods for applying symbolic logic to operations research problems.

[1] George Boole, *George Boole's Collected Logical Works* [1859], ed. P. E. B. Jourdain (Chicago and London, 1916), Vols. 1 and 2.

[2] C. S. Peirce, "On the Algebra of Logic," *Am. J. Math.*, **3**, 15–57, 1880.

[3] W. S. Jevons, *Pure Logic* [1864] *and Other Minor Works*, ed. R. Adamson and H. A. Jevons (London and New York, 1890), pp. 1–77.

[4] E. Schröder, *Algebra der Logik* (Leipzig, 1890), Vol. 1.

[5] G. Birkhoff, "Lattice Theory," *Am. Math. Soc. Coll.* (New York, 1948).

[6] H. Reichenbach, *Elements of Symbolic Logic* (New York: Macmillan Co., 1947).

[7] G. Birkhoff and S. Mac Lane, *A Survey of Modern Algebra* (New York: Macmillan Co., 1948).

[8] Robert S. Ledley, "Mathematical Foundations and Computational Methods for a Digital Logic Machine," *J. Operations Research Soc. Amer.*, Vol. **2**, No. 3 (August, 1954); Robert S. Ledley, *Bull. Am. Math. Soc.*, **60** (No. 4), 378, 1954; **61** (No. 1), 50, 52, 1955; Robert S. Ledley, *New Computational Methods in Symbolic Logic for Operations Analysts: An Introductory Manual* (Operations Research Office Technical Memorandum), Vols. 1–3 (in publication).

[9] J. D. Watson and F. H. Crick, *Nature*, **171**, 964, 1953; G. Gamow, *Nature*, **173**, 318, 1954; *Kgl. Danske Videnskab Selskab Biol. Medd.*, Vol. **22**, No. 3, 1954; G. Gamow, A. Rich, and M. Yčas, chap. 00 in *Advances of Biological and Medical Physics* (New York: Academic Press, 1955) (in press); F. Lipmann, *Currents in Biochemistry* (in press).

[10] The details can be found in Vol. 2 of Ledley, *New Computational Methods.*