PROCEEDINGS OF THE
Cambridge Philosophical Society

VOL. 54 JULY, 1958 PART 3

MANY-VALUED LOGICAL MACHINES

BY ALAN ROSE

Communicated by M. H. A. NEWMAN

Received 29 October 1957

McCallum and Smith (1) have developed a logical computer which solves problems which are essentially of the following kind.

Given a formula $\Phi(X_1, X_2, \ldots, X_n)$ constructed from the propositional variables $X_1, X_2, \ldots, X_n$ and certain logical connectives of the 2-valued propositional calculus, find an assignment of truth-values to $X_1, X_2, \ldots, X_n$ under which the formula $\Phi(X_1, X_2, \ldots, X_n)$ takes the truth-value T.

They showed that certain practical problems could be reduced to that above. The problem is then set up on a machine whose main components are 'decision elements' and a 'scanning mechanism'. A decision element corresponding to the logical functor F of i argument places ($i = 1, 2, \ldots$) is a device with i input wires and one output wire. Each wire is capable of two states, which we shall call T and F. If the formula $F(X_1, X_2, \ldots, X_i)$ takes the truth-value $f(x_1, x_2, \ldots, x_i)$ when $X_1, X_2, \ldots, X_i$ take the truth-values $x_1, x_2, \ldots, x_i$, respectively $(x_1, x_2, \ldots, x_i = T, F)$, then, when for all j $(1 \leqslant j \leqslant i)$ the jth input wire is in state x_j, the output wire will be in state $f(x_1, x_2, \ldots, x_i)$. Thus, for example, the output wire of the decision element corresponding to the logical connective $\supset$ will be in state F when the first input wire is in state T and the second is in state F. In the other three cases the output wire will be in state T. For the relay and rectifier decision elements of (1) the states T and F are connexion to positive potential and connexion to earth, respectively. When decision elements are made from thermionic valves the states T, F correspond normally to the presence of a pulse and to a pause respectively. (See, for example (2), p. 43).

Thus, a circuit can be constructed to give a decision mechanism for any formula $\Phi(X_1, X_2, \ldots, X_n)$. If this formula is of the form

$$F(\Psi_1(X_1, X_2, \ldots, X_n),\ \Psi_2(X_1, X_2, \ldots, X_n),\ \ldots,\ \Psi_i(X_1, X_2, \ldots, X_n))$$

the required circuit is obtained by first constructing decision mechanisms for the formulae $\Psi_1, \Psi_2, \ldots, \Psi_i$ and then, for all j $(1 \leqslant j \leqslant i)$, feeding the output of the decision mechanism for Ψ_j into the jth input of a decision element for F. Thus, for example, we can construct a decision mechanism for the formula

$$(X \equiv Z) \supset (X \vee Y \supset Y \,\&\, \sim Z) \,\&\, (X \equiv Z)$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

from 2 decision elements for &, 1 for ∨, 2 for ⊃, 1 for ∼ and 1 for ≡, as shown in Fig. 1. We make the convention that when $i < j$, the ith input wire of a decision element lies to the left of the jth. Two wires are not considered to be in contact unless their point of intersection in the diagram is marked with a small black circle. We note that although 8 logical functors occur in the formula only 7 decision elements are used. This is because the subformula $X \equiv Z$ occurs twice, thus enabling us to feed the output of one ≡ element into two other elements.

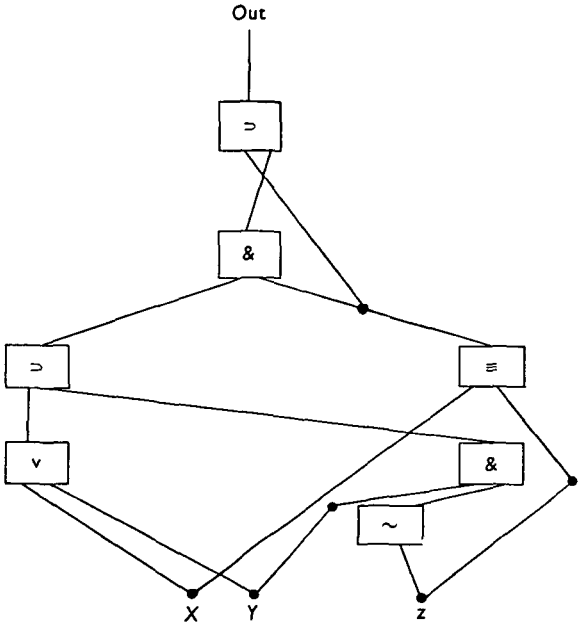


Fig. 1

In order to find a suitable assignment of truth-values to the variables of $\Phi(X_1, X_2, \ldots, X_n)$ we must be able to supply, in rapid succession, inputs corresponding to each of the 2^n possible sets of assignments of truth-values to $X_1, X_2, \ldots, X_n$. This can be done in various ways, one of which is described in (1). The machine is made in such a way that it stops when it reaches the first of the 2^n sets of inputs which gives an output of T, and the particular set of inputs can then be read off the machine.

It is possible to obtain decision elements for all functors of one or two arguments from a single decision element. Such a decision element is known as a universal decision element ((3), (4), (11)) and corresponds to a functor such that we may define all functors of one or two arguments in terms of the given functor and the logical constants t, f, the given functor occurring in the definition only once. Thus, to obtain a decision element for any functor of one or two arguments we require only one universal decision element. For example, if the truth-table of the fifth degree functor Φ is defined by

$$\Phi(X, Y, Z, U, V) =_T [X \Rightarrow Y, Z, U \Rightarrow V],$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

where the square brackets denote conditioned disjunction (5), we can make the definitions

$$\mathfrak{A} \vee \mathfrak{B} = {}_{\mathrm{df}} \Phi(t,f,\mathfrak{B},\mathfrak{A},f), \qquad \mathfrak{A} \downarrow \mathfrak{B} = {}_{\mathrm{df}} \Phi(f,f,\mathfrak{B},t,\mathfrak{A}),$$

$$\mathfrak{A} \,\&\, \mathfrak{B} = {}_{\mathrm{df}} \Phi(\mathfrak{A},f,\mathfrak{B},f,f), \qquad \mathfrak{A}/\mathfrak{B} = {}_{\mathrm{df}} \Phi(t,\mathfrak{A},\mathfrak{B},t,f),$$

$$\mathfrak{A} \supset \mathfrak{B} = {}_{\mathrm{df}} \Phi(t,f,\mathfrak{B},t,\mathfrak{A}), \qquad \mathfrak{A} \supsetneq \mathfrak{B} = {}_{\mathrm{df}} \Phi(f,f,\mathfrak{B},\mathfrak{A},f),$$

$$\mathfrak{A} \equiv \mathfrak{B} = {}_{\mathrm{df}} \Phi(\mathfrak{A},f,\mathfrak{B},t,\mathfrak{A}), \qquad \mathfrak{A} \not\equiv \mathfrak{B} = {}_{\mathrm{df}} \Phi(t,\mathfrak{A},\mathfrak{B},\mathfrak{A},f),$$

$$\mathfrak{A}^{A} = {}_{\mathrm{df}} \Phi(t,f,f,\mathfrak{A},f), \qquad \sim \mathfrak{A} = {}_{\mathrm{df}} \Phi(f,f,f,t,\mathfrak{A}),$$

$$[\mathfrak{A},\mathfrak{B},\mathfrak{C}] = {}_{\mathrm{df}} \Phi(\mathfrak{A},f,\mathfrak{B},\mathfrak{C},f).$$

Thus, the circuit for a v decision element would be as shown in Fig. 2. In order to obtain the logical constants t, f we have only to keep the corresponding input wires in states T, F, respectively.

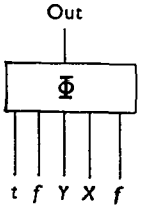


Fig. 2

The problems set to the machine by McCallum and Smith may seem to be of no very great importance and, as far as the author is aware, such machines have never been used on important problems. It would seem, however, that a large machine which could deal with the corresponding problems for many-valued propositional calculi could be used to solve problems of greater interest. In this paper we shall show that a 2-valued logical machine can be used as an m-valued machine ($m = 3, 4, \ldots$) and we shall also show how such a machine could deal with two actual problems, one concerning academic timetables and the other concerning the most efficient use of machines in a factory.

We first show how many-valued decision mechanisms can be synthesized from 2-valued decision elements. We shall consider the m-valued propositional calculus, where

$$m = \sum_{i=1}^{n-1} a_i 2^{i-1} + 2^{n-1} + 1$$

and, for all integers i ($1 \leqslant i \leqslant n-1$), $a_i = 0$ or $a_i = 1$. The truth-values are the integers $1, 2, \ldots, m$. For the present we shall rewrite the truth-values T, F as 0, 1, respectively. We define a proposition X of the m-valued propositional calculus to be an ordered set of n 2-valued propositions $(X_1, X_2, \ldots, X_n)$. We shall say that when $X_1, X_2, \ldots, X_n$ take the truth-values $x_1, x_2, \ldots, x_n$, respectively, then X takes the truth-value

$$\min\left(m, \sum_{i=1}^{n} x_i 2^{i-1} + 1\right).$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

Thus in the case $m = 11$ we should have $n = 4$ and the truth-value of X would be determined by the truth-values of X_1, X_2, X_3, X_4 as follows:

X_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
X_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
X_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
X_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X	1	2	3	4	5	6	7	8	9	10	11	11	11	11	11	11

If $F(X_1, X_2, \ldots, X_i)$ is a formula of the m-valued propositional calculus formed by combining the m-valued propositional variables $X_1, X_2, \ldots, X_i$ by means of the m-valued functor F, we may then represent each m-valued proposition X_j $(1 \leqslant j \leqslant i)$ by n 2-valued propositions $X_{j1}, X_{j2}, \ldots, X_{jn}$ and we may also represent the m-valued formula $F(X_1, X_2, \ldots, X_i)$ by n 2-valued formulae $(F(X_1, X_2, \ldots, X_i))_k$ $(k = 1, 2, \ldots, n)$. Thus $(F(X_1, X_2, \ldots, X_i))_k$ $(1 \leqslant k \leqslant n)$ is of the form

$$\Phi_k(X_{11}, X_{12}, \ldots, X_{1n}, X_{21}, X_{22}, \ldots, X_{2n}, \ldots, X_{i1}, X_{i2}, \ldots, X_{in}).$$

As a simple illustration of the above ideas we shall consider the formula $X \vee Y$ of the 4-valued propositional calculus, where $\vee$ denotes the disjunction functor of Post (6) and Łukasiewicz ((7); for an English translation see (8), pp. 38–59). We now have $n = 2$ and our 4-valued truth-table is

$X \vee Y$	1	2	3	4	Y
1	1	1	1	1	
2	1	2	2	2	
3	1	2	3	3	
4	1	2	3	4	
X					

Thus the truth-values of $(X \vee Y)_1$, $(X \vee Y)_2$ are determined by those of X_1, X_2, Y_1, Y_2 as follows:

		T	F	T	F	Y_1
$(X \vee Y)_1$		T	T	F	F	Y_2
T	T	T	T	T	T	
F	T	T	F	F	F	
T	F	T	F	T	T	
F	F	T	F	T	F	
X_1	X_2					

		T	F	T	F	Y_1
$(X \vee Y)_2$		T	T	F	F	Y_2
T	T	T	T	T	T	
F	T	T	T	T	T	
T	F	T	T	F	F	
F	F	T	T	F	F	
X_1	X_2					

Hence we may take $(X \vee Y)_1$, $(X \vee Y)_2$ to be

$$[[X_1 \vee Y_1, Y_2, X_1], X_2, [Y_1, Y_2, X_1 \vee Y_1]], \quad X_2 \vee Y_2,$$

respectively. Thus a decision mechanism for the 4-valued disjunction functor can be constructed as shown in Fig. 3. The two outputs can then be used as the two inputs corresponding to a 4-valued proposition occupying an argument place of another functor.

As a second example we shall consider the 12-valued cyclic negation functor of Post (6). The truth-table of this functor is

X	1	2	3	4	5	6	7	8	9	10	11	12
$\sim X$	2	3	4	5	6	7	8	9	10	11	12	1

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

Since the entry '12' in the lower row of the table corresponds to 5 different sets of assignments of truth-values to the quadruple (X_1, X_2, X_3, X_4) the entries in the truth-tables for $(\sim X)_1$, $(\sim X)_2$, $(\sim X)_3$ will not be uniquely determined and there will be

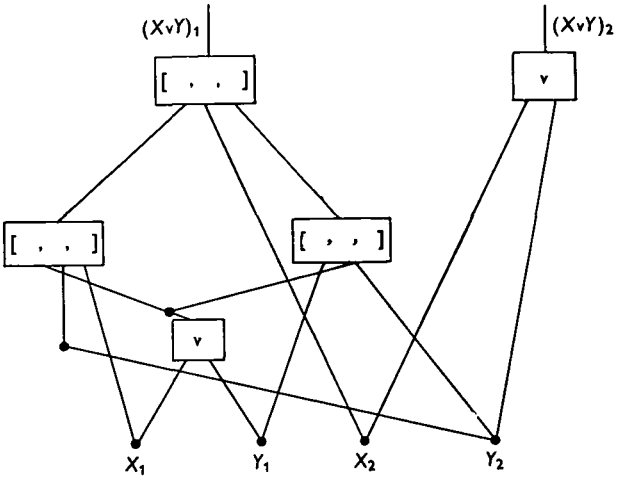


Fig. 3

5 ways of choosing the combination of these three tables.* If we had chosen a functor of one variable with α entries of '12' in the lower row the number of ways† would, of course, have been 5^{α}. Our choice is immaterial and we shall use the following tables:

X_1	X_2	X_3	X_4	$(\sim X)_1$	$(\sim X)_2$	$(\sim X)_3$	$(\sim X)_4$
T	T	T	T	F	T	T	T
F	T	T	T	T	F	T	T
T	F	T	T	F	F	T	T
F	F	T	T	T	T	F	T
T	T	F	T	F	T	F	T
F	T	F	T	T	F	F	T
T	F	F	T	F	F	F	T
F	F	F	T	T	T	T	F
T	T	T	F	F	T	T	F
F	T	T	F	T	F	T	F
T	F	T	F	F	F	T	F
F	F	T	F	T	T	T	T
T	T	F	F	T	T	T	T
F	T	F	F	T	T	T	T
T	F	F	F	T	T	T	T
F	F	F	F	T	T	T	T

Thus we may take

$$(\sim X)_1 = X_1 \supset (X_3 \downarrow X_4), \quad (\sim X)_2 = (X_1 \equiv X_2) \vee (X_3 \downarrow X_4),$$
$$(\sim X)_3 = X_4 \supset (X_1 \vee X_2 \equiv X_3), \quad (\sim X)_4 = (X_3 \equiv X_4) \vee (X_3 \not\equiv X_1 \vee X_2).$$

* Since for all five choices of truth-values of X_1, X_2, X_3, X_4 which give the truth-value 12 to X, the truth-value of X_4 is F, the truth-table of $(\sim X)_4$ is uniquely determined.

† If these α entries include an entry of '12' at the extreme right of the lower row the number of ways is $5^{\alpha+4}$.

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

Fig. 4 illustrates the use of 12-valued decision mechanisms to provide a decision mechanism for the formula $\sim (X \vee \sim Y)$.

The functors in the diagram are 12-valued and each rectangle will, of course, represent several decision elements.

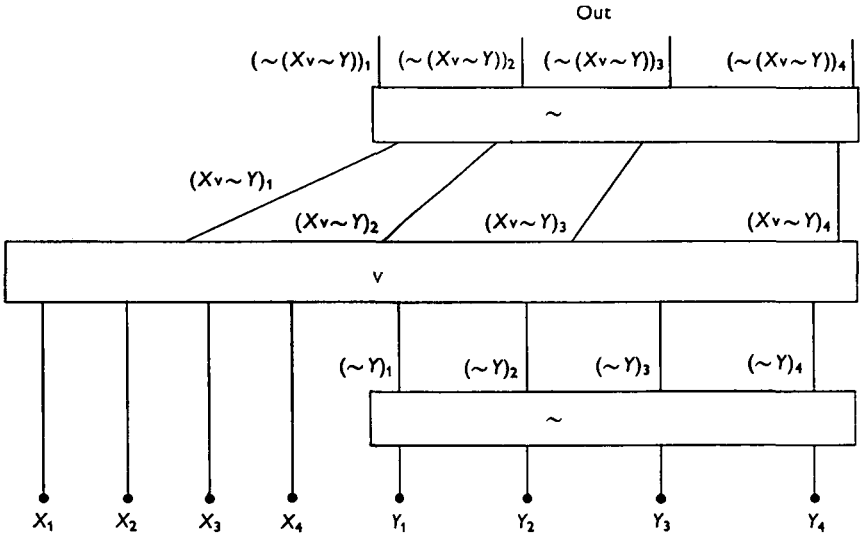


Fig. 4

We can carry out the scanning process as in the 2-valued case, since, for each m-valued propositional variable X, we can, when the machine stops, read off the truth-values of $X_1, X_2, \ldots, X_n$. From these the truth-value of X can be determined at once. If m is not an integral power of 2 the number of assignments of truth-values of 2-valued variables will then be increased by a factor of M^N/m^N, where N is the number of m-valued propositional variables considered and M is the least integral power of 2 such that $M > m$. Thus the scanning process will take longer than necessary and it may be desirable, before using the machine for such a value of m, to alter its scanning mechanism in such a way that, for each variable X, no set of truth-values for $X_1, X_2, \ldots, X_n$ for which

$$\sum_{i=1}^{n} x_i 2^{i-1} + 1 > m$$

is tested.

In the 2-valued case we wished the machine to stop when it found an assignment of truth-values to the propositional variables under which the formula being tested took the truth-value T. For the m-valued case ($m = 3, 4, \ldots$) various types of condition may correspond to this. We might stipulate that, for some one value of s ($1 \leqslant s \leqslant m-1$) the formula should take the truth-value s. Alternatively, we might stipulate that it should take a truth-value satisfying $1 \leqslant x \leqslant s$. In order to determine the procedure for stopping the machine in the first case we shall, following Rosser and Turquette (9), consider the first degree functors $J_1, J_2, \ldots, J_m$. If X, $J_i(X)$ take the truth-values $x, j_i(x)$, respectively, then

$$j_i(i) = 1,$$
$$j_i(x) = m \quad (x \neq i).$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

Let $$s = \sum_{k=1}^{n} b_k 2^{k-1} + 1 \quad (b_1, b_2, \ldots, b_n = 0, 1).$$

Then, if the summation operator Π denotes summation by '&', we may take all the formulae $(J_s(X))_1$, $(J_s(X))_2$, ..., $(J_s(X))_n$ to be

$$\prod_{k=1}^{n} F_k(X_k),$$

where $F_k(X_k)$ is X_k when $b_k = 0$ and $F_k(X_k)$ is $\sim X_k$, when $b_k = 1$. Thus, for example, if $m = 13$ the formulae $(J_7(X))_\alpha$ $(\alpha = 1, 2, 3, 4)$ may all be taken to be

$$X_1 \,\&\, \sim X_2 \,\&\, \sim X_3 \,\&\, X_4.$$

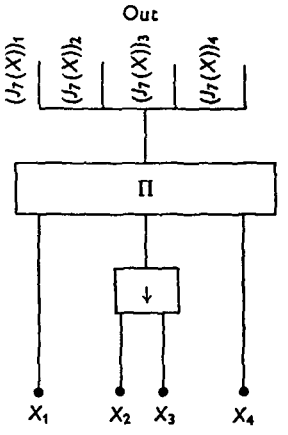


Fig. 5

This may of course be simplified to

$$X_1 \,\&\, (X_2 \downarrow X_3) \,\&\, X_4.$$

We note in this connexion that, if the summation operator Σ denotes summation by 'v', it is possible* for fairly small integers α to construct a single decision element corresponding to either of the formulae $\prod_{i=1}^{\alpha} X_i$, $\sum_{i=1}^{\alpha} X_i$. The decision mechanism for $J_7(X)$ is shown in Fig. 5. If the m-valued formula to be tested is $\Phi(X, Y, \ldots, W)$ the formula $(J_s(\Phi(X, Y, \ldots, W)))_1$ is then a 2-valued formula of the form

$$\Psi(X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n, \ldots, W_1, W_2, \ldots, W_n).$$

This can be tested in the usual way by feeding the outputs corresponding to $\Phi_1, \Phi_2, \ldots, \Phi_n$ into a $(J_s(\ ))_1$ decision mechanism. The formula Ψ takes the truth-value T if and only if $\Phi(X, Y, \ldots, W)$ takes the truth-value s.

In the second case we proceed as before except that instead of the decision mechanism for J_s we must now construct a decision mechanism for the functor V_{sm} whose truth-table is defined by

$$V_{sm}(X) = {}_T \sum_{i=1}^{s} J_i(X),$$

* See for example **(1)**, **(10)**.

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

the summation operator Σ corresponding to summation by the disjunction functor of Post and Łukasiewicz. The formulae $(V_{sm}(X))_\alpha$ $(\alpha = 1, 2, \ldots, n)$, all of which are the same, may be constructed as follows.* Let

$$s = \sum_{k=1}^{n} b_k 2^{k-1} + 1$$

as above and let x $(= x(s))$ be the greatest integer such that $b_{x+1} = 1$. If there is no such integer x is undefined. Then we may take

$$(V_{2^\alpha, m}(X)) = \prod_{i=\alpha+1}^{n} X_i \quad (\alpha = 0, 1, \ldots, n-1).$$

If s is not an integral power of 2 we can take $(V_{sm}(X))_1$ (in future abbreviated to $V_{sm}(X)$) to be

$$V_{2^x, m}(X) \vee V_{2^{x+1}, m}(X) \,\&\, V_{s-2^x, 2^x}(X) \quad (2^{x+1} < m),$$
$$V_{2^x, m}(X) \vee V_{s-2^x, 2^x}(X) \quad (2^{x+1} \geqslant m).$$

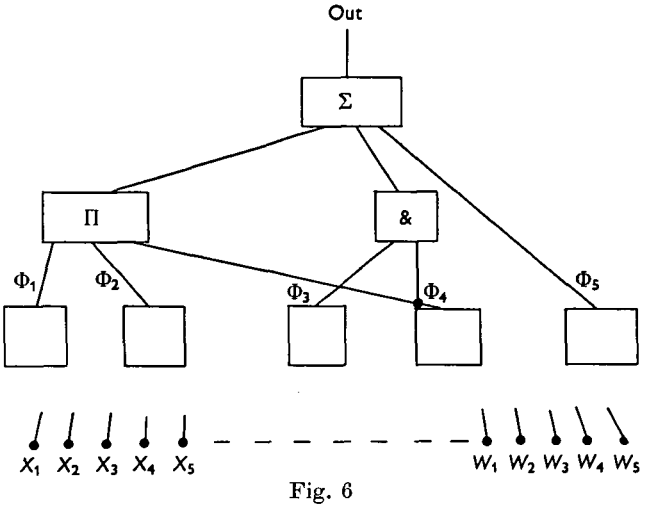


Fig. 6

Thus, for example, we can take the formula $V_{21, 29}(X)$ to be

$$V_{16, 29}(X) \vee V_{5, 16}(X),$$

i.e.
$$X_5 \vee V_{4, 16}(X) \vee V_{8, 16}(X) \,\&\, V_{1, 4}(X),$$

i.e.
$$X_5 \vee X_4 \,\&\, X_3 \vee X_4 \,\&\, X_2 \,\&\, X_1.$$

Thus if we wished to find an assignment of truth-values to $X, Y, \ldots, W$ under which the formula $\Phi(X, Y, \ldots, W)$ of the 29-valued propositional calculus took a truth-value satisfying $1 \leqslant x \leqslant 21$ we would first set up decision mechanisms for $\Phi_1, \Phi_2, \ldots, \Phi_5$ and then complete the circuit as shown in Fig. 6. The formula

$$\Phi_5 \vee \Phi_4 \,\&\, \Phi_3 \vee \Phi_4 \,\&\, \Phi_2 \,\&\, \Phi_1$$

is a 2-valued formula constructed from the 2-valued variables $X_1, X_2, \ldots, X_5$, $Y_1, Y_2, \ldots, Y_5, \ldots, W_1, W_2, \ldots, W_5$ which takes the truth-value T if and only if $\Phi(X, Y, \ldots, W)$ takes a truth-value satisfying $1 \leqslant x \leqslant 21$.

* In certain particular cases, simpler constructions may be possible.

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

A further alternative is that we might wish to determine the least integer s such that there exists an assignment of truth-values to $X, Y, \ldots, W$ under which $\Phi(X, Y, \ldots, W)$ takes the truth-value s and to determine one such assignment. This can be accomplished by carrying out the scanning process twice and making use of flip-flops.

A flip-flop is a device with one input and one output. Initially the input wire is in state F but, if the input wire ever becomes in state T, the output wire becomes in state T and remains in that state, even if the state of the input wire later reverts to F. Let the least and greatest values of s considered to be possible answers to the first part of the problem be α, β, respectively. We then set up decision mechanisms corresponding to each of the functors $J_\alpha, J_{\alpha+1}, \ldots, J_\beta$ and feed into these the formulae $\Phi_1, \Phi_2, \ldots, \Phi_n$, thus obtaining outputs corresponding to $(J_\alpha(\Phi))_1, (J_{\alpha+1}(\Phi))_1, \ldots, (J_\beta(\Phi))_1$. We then take $\beta - \alpha + 1$ flip-flops and feed the outputs of the J-mechanisms into these, the output (if any) of each flip-flop lighting a lamp. The scanning process is allowed to terminate and we then determine, by observing the lamps, the smallest integer γ such that $(J_\gamma(\Phi))_1$ takes the truth-value T for at least one assignment of truth-values to $X, Y, \ldots, W$. We then take $s = \gamma$, remove the flip-flops and all but one of the J-mechanisms and determine, by scanning in the manner described above, an assignment of truth-values to $X, Y, \ldots, W$ under which Φ takes the truth-value s.

We shall now consider the application of the above methods to the construction of academic timetables. Let us suppose that there are classes $C_1, C_2, \ldots, C_\alpha$ of students and lecturers $L_1, L_2, \ldots, L_\beta$. Let the number of lectures per week delivered to C_i by L_j be $f(i,j)$. Our propositional variables will be those X_{ijk} for which $f(i,j) \neq 0$ and $1 \leqslant k \leqslant f(i,j)$. Let us suppose that lectures may begin at any of λ times of day on any of μ days of the week and let the ith lecture time on the jth day of the week be numbered $(i-1)\mu + j$. Let $\lambda\mu = m$ and let us consider an m-valued logic. We shall denote by X_{ijk} ($f(i,j) \neq 0$, $1 \leqslant k \leqslant f(i,j)$) the proposition '$L_j$ takes C_i and L_j has previously taken C_i on $k-1$ occasions during the week'. If L_j takes C_i for the kth time during the period whose number is ν we shall say that X_{ijk} has the truth-value ν.

In order to construct a suitable timetable we must impose certain conditions on the times at which lectures are held, i.e. we must impose conditions on the truth-values of the X_{ijk}. Let us suppose that we wish to impose a condition on the truth-values of $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_\omega$ where $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_\omega$ are the names of ω of the X_{ijk}. The total number of sets of assignments to these propositional variables will be m^ω. Let us suppose that these m^ω assignments are divided into ξ classes $\mathscr{E}_1, \mathscr{E}_2, \ldots, \mathscr{E}_\xi$ and that, for an assignment belonging to $\mathscr{E}_1$ the condition is completely satisfied, the corresponding feature of the timetable causing no dissatisfaction to anyone. Let us further suppose that, for an assignment belonging to $\mathscr{E}_\eta$ ($2 \leqslant \eta \leqslant \xi$) the condition is not satisfied, an undesirable feature of the timetable thereby resulting and that, for $\zeta > \eta$, an assignment of $\mathscr{E}_\zeta$ causes a greater degree of dissatisfaction than an assignment of $\mathscr{E}_\eta$. In order to express this condition we must then construct a formula

$$\Phi(\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_\omega)$$

which takes the truth-value 1 when the assignment of truth-values belongs to $\mathscr{E}_1$ and which, for some selection of truth-values $v_2, v_3, \ldots, v_\xi$ such that

$$1 < v_2 < v_3 < \ldots < v_\xi \leqslant m,$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

takes the truth-value v_δ when the assignment of truth-values belongs to $\mathscr{E}_\delta$ $(\delta = 2, 3, \ldots, \xi)$. The choice of $v_2, v_3, \ldots, v_\xi$ will, of course, depend on the degree of dissatisfaction caused in each case. If some assignments cause an intolerable state of affairs we must take $v_\xi = m$.

Thus, for example, if L_1 lectures to C_1 once per week and to C_2 once per week, the condition that he cannot deliver both lectures simultaneously is expressed by the formula

$$X_{111} \not\equiv X_{211}.$$

The m-valued functor $\not\equiv$ is such that if X, Y, $X \not\equiv Y$ take the truth-values x, y, $r(x, y)$, respectively, then

$$r(x, x) = m,$$
$$r(x, y) = 1 \quad (x \neq y).$$

The following method will enable us to construct the 2-valued formulae $\Phi_1, \Phi_2, \ldots, \Phi_n$ corresponding to the m-valued formula Φ. Let $v_1 = 1$ and let the set of 2-valued truth-values corresponding to v_δ be $(w_{1\delta}, w_{2\delta}, \ldots, w_{n\delta})$ $(\delta = 1, 2, \ldots, \xi)$. If, for some integer $\tau\,(1 \leqslant \tau \leqslant n)$ the truth-values $w_{\tau\delta}$ $(\delta = 1, 2, \ldots, \xi)$ are all T (all F) we take the formula Φ_τ to be the logical constant $t(f)$. Let us now consider those values of τ for which the truth-values $w_{\tau 1}, w_{\tau 2}, \ldots, w_{\tau\xi}$ are not all the same. We then construct a 2-valued formula

$$\Phi_\tau(\mathfrak{A}_{11}, \mathfrak{A}_{12}, \ldots, \mathfrak{A}_{1n}, \mathfrak{A}_{21}, \mathfrak{A}_{22}, \ldots, \mathfrak{A}_{2n}, \ldots, \mathfrak{A}_{\omega 1}, \mathfrak{A}_{\omega 2}, \ldots, \mathfrak{A}_{\omega n})$$

which, for all δ $(\delta = 1, 2, \ldots, \xi)$, takes the truth-value $w_{\tau\delta}$ when the truth-values of $\mathfrak{A}_{11}, \mathfrak{A}_{12}, \ldots, \mathfrak{A}_{1n}, \mathfrak{A}_{21}, \mathfrak{A}_{22}, \ldots, \mathfrak{A}_{2n}, \ldots, \mathfrak{A}_{\omega 1}, \mathfrak{A}_{\omega 2}, \ldots, \mathfrak{A}_{\omega n}$ correspond to an assignment of truth-values to $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_\omega$ belonging to $\mathscr{E}_\delta$. This is the required formula Φ_τ.

For the case $m = 2^n$ it can easily be seen that, corresponding to the example given above, the formulae $(X \not\equiv Y)_1$, $(X \not\equiv Y)_2$, ..., $(X \not\equiv Y)_n$ are all

$$\sum_{i=1}^{n} (X_i \not\equiv Y_i).$$

In order to avoid timetable clashes we need all the conditions expressed by the formulae

$$X_{i_1 j_1 k_1} \not\equiv X_{i_1 j_2 k_2}, \quad X_{i_1 j_1 k_1} \not\equiv X_{i_2 j_1 k_2}, \quad X_{i_1 j_1 k_1} \not\equiv X_{i_1 j_1 k_3}$$
$$(i_1 \neq i_2,\ j_1 \neq j_2,\ k_1 \neq k_3,\ f(i_1, j_1) \neq 0,\ f(i_1, j_2) \neq 0,\ f(i_2, j_1) \neq 0,$$
$$1 \leqslant k_1 \leqslant f(i_1, j_1),\ 1 \leqslant k_2 \leqslant f(i_1, j_2),\ k_2 \leqslant f(i_2, j_1),\ 1 \leqslant k_3 \leqslant f(i_1, j_1)).$$

We do not need to express the fact that the truth-value of $X_{i_1 j_1 k_1}$ is less than that of $X_{i_1 j_1 k_2}$ for $k_1 < k_2$, since we can always reverse the roles of $X_{i_1 j_1 k_1}$, $X_{i_1 j_1 k_2}$.

The numbers of m-valued decision mechanisms required to express the conditions of the above three types are, respectively

$$\sum_{i=1}^{\alpha} \sum_{\lambda=1}^{\beta} \sum_{\mu=\lambda+1}^{\beta} (f(i, \lambda) f(i, \mu)),$$
$$\sum_{j=1}^{\beta} \sum_{\lambda=1}^{\alpha} \sum_{\mu=\lambda+1}^{\alpha} (f(\lambda, j) f(\mu, j)),$$
$$\sum_{(i,j) \in E} {}^{f(i,j)}C_2,$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

where E is the set of pairs of integers (i, j) such that $f(i, j) \geqslant 2$. Since each m-valued $\equiv$ mechanism requires $n+1$ decision elements (including one for Σ), the total number of decision elements required for these conditions is

$$(n+1)\,(\Sigma(f(i,\lambda)\,f(i,\mu))+\Sigma(f(\lambda,j)\,f(\mu,j))+\sum_{(i,j)\in E}{}^{f(i,j)}C_2),$$

where the first two Σ's denote triple sums with limits of summation as above.

Thus for example, if $\alpha = 6$, $\beta = 8$, $m = 32$ and the lectures are distributed as shown below, each lecturer taking each class three times per week (or not at all), we have $n = 5$ and the number of decision elements required is

$$6(6.3.9+(1+3+3+1+1+1+1+1).9+18.{}^3C_2) = 6(162+108+54) = 1944.$$

Class ...	1	2	3	4	5	6
Lecturers	1	1	2	4	3	2
	4	2	3	7	5	3
	5	7	8	8	6	6

Similar methods could be used to express other conditions, e.g. that every lecturer must have a free day, that no lecturer must meet the same class twice on one day or on two consecutive days or that not more than a given number of lectures are held simultaneously. In order to determine whether a timetable is acceptable as a whole we must combine the Φ's by some form of conjunction functor. The most appropriate functor might perhaps be that whose truth-value is such that if x, y, $k(x, y)$ are the truth-values of X, Y, $X \,\&\, Y$, respectively, then

$$k(x, y) = \min(m, x+y-1).$$

Decision mechanisms for this functor can be constructed by a method similar to that given below for the functor '$\circ$'. The best timetable can then be determined by using flip-flops as described above.

At the present state of electrical development the scanning time might be rather long if many variables are used. This might be reduced by the feed-back method described by McCallum and Smith (1) or by presetting the truth-values of some of the propositional variables. This should be possible since most of the practical difficulties in timetable construction usually occur towards the end of the work. Also the machine could first be programmed to decide merely on which day each lecture should be given. the instructions given to it for this being such that the subsequent problems of constructing detailed timetables for each day are feasible. The logic for the first stage would then be μ-valued, corresponding to the μ days of the week, and the machine would be instructed, among other things, not to select more than a certain number of lectures for any one day. The logic of the remaining μ stages would then be λ-valued, corresponding to the λ periods of the day.

Finally we shall consider a problem* relating to the economic use of factory machines. Let us suppose that a factory contains N machines and that each job to be done during a particular week has to be passed through machines $M_1, M_2, \ldots, M_N$,

* I am indebted to Mr M. Castledine for drawing my attention to this problem.

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

the job being transferred to M_i from M_{i-1} immediately M_{i-1} has finished with it or as soon thereafter as M_i is available ($i = 2, 3, \ldots, N$). Let the jobs be $J_1, J_2, \ldots, J_n$, the time spent on J_i by M_j being t_{ij} hours* ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant N$). We shall assume that the t_{ij} are all integers. We have to determine the permutation $i_1, i_2, \ldots, i_n$ of the integers $1, 2, \ldots, n$ such that, if M_1 is supplied with J_{i_k} immediately it has finished with $J_{i_{k-1}}$ ($k = 2, 3, \ldots, n$), the length of time elapsing between the beginning of J_{i_1} by M_1 and the ending of J_{i_n} by M_N is as short as possible.

We shall use propositional variables $X_1, X_2, \ldots, X_{n-1}$, the variable X_i being capable of taking the truth-values $1, 2, \ldots, n-i+1$ ($i = 1, 2, \ldots, n-1$). If $x-1$ of $J_{i+1}, J_{i+2}, \ldots, J_n$ are begun by M_1 before it begins J_i we shall say that X_i takes the truth-value x. Let the 2-valued propositional variables corresponding to X_i be $X_{i1}, X_{i2}, \ldots, X_{i\alpha_i}$, where

$$n-i+1 = 2^{\alpha_i-1}+\lambda \quad (1 \leqslant \lambda \leqslant 2^{\alpha_i-1}).$$

Our first major objective is to construct the formulae

$$\Phi_{xy}(X_{11}, X_{12}, \ldots, X_{1\alpha_1}, X_{21}, X_{22}, \ldots, X_{2\alpha_2}, \ldots, X_{n-1,1})$$

which take the truth-value T if and only if the truth-values of $X_1, X_2, \ldots, X_{n-1}$ are such that $x = i_y$. In order to construct the Φ_{xy} we first construct the formulae $E_{i,\alpha}(X_1, X_2, \ldots, X_\alpha)$ which take the truth-value T if and only if exactly i of $X_1, X_2, \ldots, X_\alpha$ take the truth-value T ($i = -1, 0, 1, 2, \ldots, \alpha+1$). We make the definitions

$$E_{02}(X_1, X_2) = {}_{\mathrm{df}}\, X_1 \downarrow X_2, \quad E_{12}(X_1, X_2) = {}_{\mathrm{df}}\, X_1 \equiv X_2, \quad E_{22}(X_1, X_2) = {}_{\mathrm{df}}\, X_1 \,\&\, X_2,$$

$$E_{-1,\alpha}(X_1, X_2, \ldots, X_\alpha) = {}_{\mathrm{df}} f, \quad E_{\alpha+1,\alpha}(X_1, X_2, \ldots, X_\alpha) = {}_{\mathrm{df}} f,$$

$$E_{i,\alpha+1}(X_1, X_2, \ldots, X_{\alpha+1}) = {}_{\mathrm{df}}\, [E_{i-1,\alpha}(X_1, X_2, \ldots, X_\alpha), X_{\alpha+1}, E_{i,\alpha}(X_1, X_2, \ldots, X_\alpha)] \quad (0 \leqslant i \leqslant \alpha+1).$$

Thus, for example, we have

$$\begin{aligned}
&E_{35}(X_1, X_2, X_3, X_4, X_5)\\
&\quad = {}_{\mathrm{df}}\, [E_{24}(X_1, X_2, X_3, X_4), X_5, E_{34}(X_1, X_2, X_3, X_4)]\\
&\quad = {}_{\mathrm{df}}\, [[E_{13}(X_1, X_2, X_3), X_4, E_{23}(X_1, X_2, X_3)], X_5,\\
&\qquad\qquad [E_{23}(X_1, X_2, X_3), X_4, E_{33}(X_1, X_2, X_3)]]\\
&\quad = {}_{\mathrm{df}}\, [[[E_{02}(X_1, X_2), X_3, E_{12}(X_1, X_2)], X_4, [E_{12}(X_1, X_2), X_3, E_{22}(X_1, X_2)]],\\
&\qquad X_5, [[E_{12}(X_1, X_2), X_3, E_{22}(X_1, X_2)], X_4, [E_{22}(X_1, X_2), X_3, E_{32}(X_1, X_2)]]]\\
&\quad = {}_{\mathrm{df}}\, [[[X_1 \downarrow X_2, X_3, X_1 \equiv X_2], X_4, [X_1 \equiv X_2, X_3, X_1 \,\&\, X_2]], X_5,\\
&\qquad\qquad [[X_1 \equiv X_2, X_3, X_1 \,\&\, X_2], X_4, [X_1 \,\&\, X_2, X_3, f]]].
\end{aligned}$$

Thus, the circuit would be as shown in Fig. 7. We note that, since a single output can be used to provide inputs for several other elements, this circuit also provides us with decision mechanisms for the functors $E_{02}, E_{12}, E_{22}, E_{13}, E_{23}, E_{33}, E_{24}, E_{34}$.

More generally, having obtained decision mechanisms for all the functors $E_{02}, E_{12}, E_{22}, \ldots, E_{0i}, E_{1i}, \ldots, E_{ii}$ we can obtain decision mechanisms for $E_{0,i+1}$,

* Usually an hour will be the most convenient unit of time, but a different unit may, of course, be used if desired.

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

$E_{1,i+1}, \ldots, E_{i+1,i+1}$ by using in addition $i+2$ conditioned disjunction elements, provided that for all j the formulae occupying the argument places of the E_{ij} are independent of the value of i and form a subset of those occupying the argument places of the $E_{i,j+1}$.

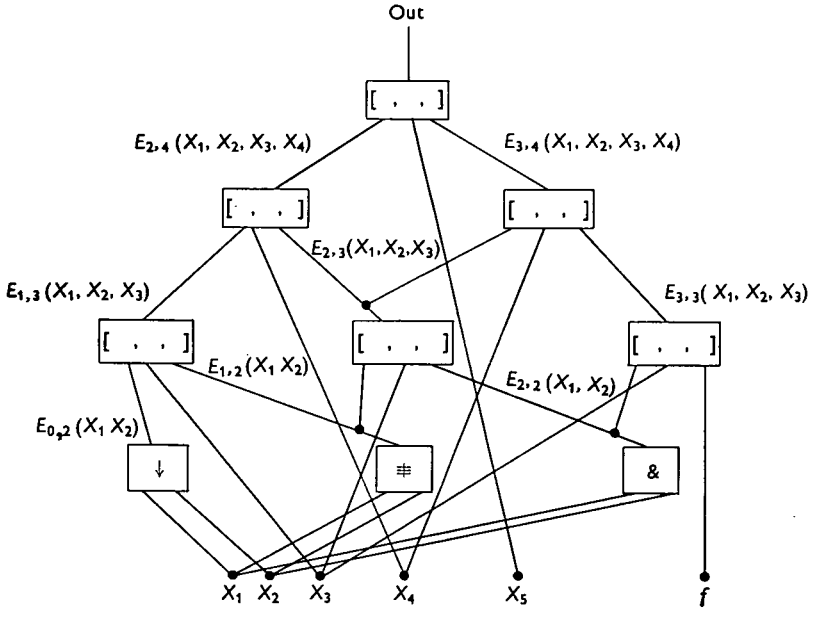


Fig. 7

We now take the Φ_{xy} to be

$$\sum_{i=-x+y-1}^{y} (J_i(X_x) \,\&\, E_{y-i,(x-1)(y-1)}(\Phi_{11}, \Phi_{12}, \ldots, \Phi_{1,y-1}, \ldots, \Phi_{x-1,1}, \Phi_{x-1,2}, \ldots, \Phi_{x-1,y-1})).$$

Since for each integer α $(1 \leqslant \alpha \leqslant x-1)$ at most one of the formulae $\Phi_{\alpha 1}, \Phi_{\alpha 2}, \ldots, \Phi_{\alpha,y-1}$ can take the truth-value T, we may replace the formula

$$E_{y-i,(x-1)(y-1)}(\Phi_{11}, \ldots, \Phi_{x-1,y-1})$$

by

$$E_{y-i,x-1}\left(\sum_{k=1}^{y-1} \Phi_{1k}, \sum_{k=1}^{y-1} \Phi_{2k}, \ldots, \sum_{k=1}^{y-1} \Phi_{x-1,k}\right).$$

We note that, in view of the property of E_{ij} mechanisms mentioned in the previous paragraph, the decision mechanisms for

$$E_{y-i,x-1}\left(\sum_{k=1}^{y-1} \Phi_{1k}, \sum_{k=1}^{y-1} \Phi_{2k}, \ldots, \sum_{k=1}^{y-1} \Phi_{x-1,k}\right)$$

can be constructed from those for

$$E_{y-i,x-2}\left(\sum_{k=1}^{y-1} \Phi_{1k}, \ldots, \sum_{k=1}^{y-1} \Phi_{x-2,k}\right).$$

Now that we have obtained the Φ_{xy} we are in a position to use an m-valued logic, the value of m being the greatest number of hours which we could possibly allow for the completion of the jobs in any week. We shall construct formulae Ψ_{xy}, Λ_{xy} whose

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

truth-values are the number of hours which elapse from the beginning of the week's work to the respective times of the starting and finishing of J_{i_y} by M_x.

We first construct the formulae Ω_{xy} whose truth-values are $t_{i_x y}$. These are given by

$$(\Omega_{xy})_\alpha = \prod_{k=1}^{n} (\Phi_{kx} \supset (t_{ky})_\alpha),$$

where $(t_{ky})_\alpha$ is the αth 2-valued logical constant corresponding to the truth-value t_{ky}. We can then construct the Ψ's and Λ's as follows, the operation & being the conjunction functor of Post and Łukasiewicz (taking the truth-value $\max(x, y)$) and the operation 'o' being such that if the truth-values of X, Y, $X \circ Y$ are $x, y, f(x, y)$, respectively, then $f(x, y) = \min(m, x + y)$.

$$\Lambda_{11} = {}_{\mathrm{df}}\, \Omega_{11}, \quad \Lambda_{1,y+1} = {}_{\mathrm{df}}\, \Lambda_{1y} \circ \Omega_{y+1,1} \quad (y = 1, 2, \ldots, n-1),$$
$$\Psi_{1,y+1} = {}_{\mathrm{df}}\, \Lambda_{1y} \quad (y = 1, 2, \ldots, n-1),$$
$$\Psi_{x+1,1} = {}_{\mathrm{df}}\, \Lambda_{x,1} \quad (x = 1, 2, \ldots, N-1),$$
$$\Psi_{x+1,y} = {}_{\mathrm{df}}\, \Lambda_{xy} \,\&\, \Lambda_{x+1,y-1} \quad (x = 1, 2, \ldots, N-1;\ y = 1, 2, \ldots, n-1),$$
$$\Lambda_{x+1,y} = {}_{\mathrm{df}}\, \Psi_{x+1,y} \circ \Omega_{y,x+1} \quad (x = 1, 2, \ldots, N-1;\ y = 1, 2, \ldots, n).$$

We have then to find the assignment of truth-values to the 2-valued propositional variables which corresponds to the permutation $i_1, i_2, \ldots, i_n$ which makes the truth-value of $\Lambda_{N,n}$ as high (i.e. numerically low) as possible. This can be accomplished by scanning twice and using flip-flops as described above. It will facilitate the reading off of the answer from the computer if the truth-values of the Φ_{xy} are able to be read off direct.

It now only remains for us to determine the 2-valued formulae corresponding to the m-valued functors &, o. It will be convenient to us to take m to be a power of 2 ($m = 256$ should usually be sufficient).

Let $m = 2^n$ and let us consider X & Y, where X, Y are represented by the 2-valued variables $X_1, X_2, \ldots, X_n$, $Y_1, Y_2, \ldots, Y_n$. We first construct the 2-valued formula $G(X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n)$ which takes the truth-value T if and only if $x \geqslant y$, where x, y are the truth-values of X, Y, respectively. We make the definitions

$$G(X_1, Y_1) = {}_{\mathrm{df}}\, X_1 \supset Y_1,$$
$$G(X_1, X_2, \ldots, X_{n+1}, Y_1, Y_2, \ldots, Y_{n+1}) = {}_{\mathrm{df}}\, [G(X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n),\ X_{n+1} \equiv Y_{n+1}, Y_{n+1}].$$

We can then take

$$(X \,\&\, Y)_i = [X_i, G(X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n), Y_i].$$

In order to obtain the formulae $(X \circ Y)_i$ $(i = 1, 2, \ldots, n)$ we use a variation of the 'sum and carry' method of the serial adder ((2), p. 53). We make the definitions given below, using the fact that the formula $[X, Y \not\equiv Z, Z]$ takes the truth-value T if and only if at least 2 of X, Y, Z take the truth-value T.

$$\Xi_1(X_1, Y_1) = {}_{\mathrm{df}}\, X_1 \not\equiv Y_1, \quad \Theta_1(X_1, Y_1) = {}_{\mathrm{df}}\, X_1 \,\&\, Y_1,$$
$$\Xi_{i+1}(X_1, \ldots, X_{i+1}, Y_1, \ldots, Y_{i+1}) = {}_{\mathrm{df}}\, (X_{i+1} \not\equiv Y_{i+1}) \not\equiv \Theta_i(X_1, \ldots, X_i, Y_1, \ldots, Y_i),$$
$$\Theta_{i+1}(X_1, \ldots, X_{i+1}, Y_1, \ldots, Y_{i+1}) = {}_{\mathrm{df}}\, [\Theta_i, X_{i+1} \not\equiv Y_{i+1}, X_{i+1}].$$
$$(X \circ Y)_i = {}_{\mathrm{df}}\, \Xi_i(X_1, \ldots, X_i, Y_1, \ldots, Y_i) \,\&\, \Theta_n(X_1, \ldots, X_n, Y_1, \ldots, Y_n).$$

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press

REFERENCES

(1) McCallum, D. B. and Smith, J. B. Mechanized reasoning—logical computors and their design. *Electronic Engng*, 23 (1951), 126–33.
(2) Bowden, B. V. *Faster than thought* (London, 1953).
(3) Sobociński, B. On a universal decision element. *J. Comput. Systems*, 1 (1953), 71–80.
(4) Rose, A. Sur les éléments universels de décision. *C. R. Acad. Sci., Paris*, 244 (1957), 2343–5.
(5) Church, A. Conditioned disjunction as a primitive connective for the propositional calculus. *Portug. Math.* 7 (1948), 87–90.
(6) Post, E. L. Introduction to a general theory of elementary propositions. *Amer. J. Math.* 43 (1921), 163–85.
(7) Łukasiewicz, J. and Tarski, A. Untersuchungen über den Aussagenkalkül. *C. R. Acad. Sci., Warsaw* (3), 23 (1930), 30–50.
(8) Tarski, A. *Logic, semantics, metamathematics* (Oxford, 1956).
(9) Rosser, J. B. and Turquette, A. R. *Many-valued logics* (Amsterdam, 1952).
(10) Greniewski, H., Bochenek, K. and Marczyński, R. Applications of bi-elemental Boolean algebra to electronic circuits. *Studia logica*, 2 (1955), 7–76.
(11) Rose, A. and Parton, J. E. Improvements in or relating to decision element circuits. British Patent Application No. 28685/56.

The University
Nottingham

https://doi.org/10.1017/S0305004100033508 Published online by Cambridge University Press