



Soviet-era science, translated into English

MATHEMATICS

Yu. I. YANOV

1957

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-195701.32046>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

Abstract

Full Text

MATHEMATICS

Yu. I. YANOV

ON THE EQUIVALENCE AND TRANSFORMATIONS OF PROGRAM SCHEMES

(Presented by Academician M. V. Keldysh on 6 X 1956)

In programming for universal automatic computing machines, (logical) **program schemes** (p.s.) are used ⁽¹⁾. Since a p.s. is not uniquely determined by an algorithm, questions arise concerning equivalence, as well as identical transformations, of p.s. In the present article, p.s. are considered as certain records of the order in which operators and logical conditions are executed, depending on the values of logical (two-valued) variables*. In this connection, operators are regarded as elementary objects to which a definite ability to change the values of logical variables is ascribed.

Definition 1. Symbols $_i, _j$ with natural indices i, j will be called, respectively, left and right primes. An ordered collection of an algebraic-logical function α and a left prime $_i$ will be called a **logical condition** α_{i} . Operators A_1, A_2, \dots and logical conditions will be called **elementary expressions** (e.e.).

Definition 2. A **program scheme** is a finite string composed of symbols of operators A_1, A_2, \dots , logical conditions $\alpha_{i}, \beta_{j}, \dots$, and right primes $_i, _j, \dots$, such that for each left prime $_i$ with index i occurring in this string, there is in it one and only one right prime $_i$ with the same index i , and conversely, for each right prime $_i$ there is one and only one left prime with the same index.

We shall assume that each occurrence of an operator in a p.s. is individualized.

Definition 3. Let there be a p.s. $\mathfrak{A}(p_1, \dots, p_k, A_1, \dots, A_n)$, where p_1, \dots, p_k are independent logical variables. Denote by $\Delta_1, \dots, \Delta_{2^k}$ all possible sets of values of the variables p_1, \dots, p_k . We shall call the **value of the scheme** $\mathfrak{A}(p_1, \dots, p_k, A_1, \dots, A_n)$ for the **sequence of sets**

$$\Delta_{s_1}, \Delta_{s_2}, \dots, \Delta_{s_m}, \Delta_{s_{m+1}}, \dots \quad (1 \leq s_j \leq 2^k, j = 1, 2, \dots) \quad (1)$$

the string of operators obtained by the following process.

1st step: mark the leftmost symbol of the scheme \mathfrak{A} .

Suppose that l steps have been performed, as a result of which the string of operators

$$A_{i_1}, A_{i_2}, \dots, A_{i_{m-1}} \quad (2)$$

has been written down.

* Taking the values 0 and 1, and let 0 correspond to “false,” and 1 to “true.” (in this case the empty line should be denoted by A_{i_0}). Then at the $(l+1)$ -st step we consider the symbol marked at the l -th step; moreover, if it is an e.v., then we shall say that it is executed in \mathfrak{A} for the sequence (1) under the set Δ_{s_m} .

The following cases are possible:

- 1) The symbol marked at the l -th step is an operator. Then we append it on the right to line (2) and mark in the scheme \mathfrak{A} the symbol immediately following it.
- 2) The symbol marked at the l -th step is the logical condition $\alpha(p_1, \dots, p_k) \perp_i$. Then, in the case where $\alpha(\Delta_{s_m}) = 1$, we mark in the scheme \mathfrak{A} the symbol immediately following the logical condition $\alpha \perp_i$; but if $\alpha(\Delta_{s_m}) = 0$, then we mark the symbol standing in \mathfrak{A} immediately to the right of the right prime \perp_i (if such symbols occur in \mathfrak{A}).
- 3) Finally, if the symbol marked at the l -th step is a right prime, then we mark in \mathfrak{A} the symbol immediately following it (if there is one). The process terminates if at some step no symbol is marked. The line of operators written out as a result of this process will be called the value of the s.p. \mathfrak{A} for the sequence (1). In this connection, in the case where the process described continues infinitely, while the line of operators written out is finite, we append to it on the right parentheses $()$, which we shall call an empty period. Values with an empty period will be regarded as infinite.

Definition 4. We shall say that for the set of operators $\mathfrak{S} = \{A_1, \dots, A_n\}$ and the set of logical variables $\mathfrak{P} = \{p_1, \dots, p_k\}$ a distribution of shifts is given,

$$A_i - \mathfrak{P}_i \quad (i = 1, \dots, n), \quad (3)$$

if to each operator A_i from \mathfrak{S} there is put in correspondence some set of logical variables $\mathfrak{P}_i \subset \mathfrak{P}$.

Definition 5. Let the s.p. $\mathfrak{A}(p_1, \dots, p_k, A_1, \dots, A_n)$, for the sequence (1), have the value

$$A_{i_1}, A_{i_2}, \dots, A_{i_m}, A_{i_{m+1}}, \dots \quad (4)$$

Then the sequence (1) will be called admissible for \mathfrak{A} under the distribution of shifts (3), if for every $m = 1, 2, \dots$ the set $\Delta_{s_{m+1}}$ differs from the set Δ_{s_m} in the values of variables only from \mathfrak{P}_{i_m} . Moreover, if line (4) is finite and A_{i_r} is its last operator, then the sets $\Delta_{s_{r+2}}, \Delta_{s_{r+3}}, \dots$ are arbitrary.

Definition 6. We shall say that the s.p. $\mathfrak{A}(p_1, \dots, p_k)$ and $\mathfrak{B}(p_1, \dots, p_k)$ are equivalent under the distribution of shifts (3) (which we shall denote by $\mathfrak{A} = \mathfrak{B}$), if for any sequence of sets admissible for \mathfrak{A} or \mathfrak{B} under this distribution of shifts, their values coincide.

Definition 7. We shall say that the s.p. \mathfrak{A} and \mathfrak{B} are partially equivalent under the distribution of shifts (3), if for any sequence of sets admissible for \mathfrak{A} or \mathfrak{B} , either their values coincide, or the value of that one of them for which this sequence is admissible is infinite.

The following problems are decidable: to find an algorithm which, for any pair of s.p.'s, would give an answer as to whether they are equivalent (respectively, partially equivalent) or not under the given distribution of shifts.

Definition 8. For each e.v. \mathfrak{B} and program scheme $\mathfrak{A}(p_1, \dots, p_k)$ we define the algebraic-logical function $\mathfrak{B}_{(\mathfrak{A})}^{\otimes}(p_1, \dots, p_k)$ as follows:

$$\mathfrak{B}_{(\mathfrak{A})}^{\otimes}(\Delta_s) = \begin{cases} 1, & \text{if } \mathfrak{B} \text{ is fulfilled in } \mathfrak{A} \text{ for the stationary sequence } \Delta_s, \Delta_s, \dots, \Delta_s, \dots; \\ 0, & \text{otherwise.} \end{cases}$$

Definition 9. For a program scheme $\mathfrak{A}(p_1, \dots, p_k, A_1, \dots, A_n)$, under the distribution of shifts (3), denote

$$\alpha_i^1 = \max_{\mathfrak{P}_i} A_{i(\mathfrak{A})}^{\otimes}(p_1, \dots, p_k) \quad (i = 1, \dots, n),$$

where the maximum is taken over all possible sets of values of the variables from \mathfrak{P}_i , and consider the program scheme

$$\mathfrak{A}^{(1)} \equiv q_1 \vee \overline{\alpha_1^1} \lfloor_{j_1} \dots q_n \vee \overline{\alpha_n^1} \lfloor_{j_n} \mathfrak{A}(p_1, \dots, p_k, A_1 \rfloor_{j_1}, \dots, A_n \rfloor_{j_n})^*,$$

where q_1, \dots, q_n are independent logical variables. Suppose the functions α_i^{ν} ($i = 1, \dots, n$) and the program scheme $\mathfrak{A}^{(\nu)}$, which we shall call the ν -th immersion of the program scheme \mathfrak{A} , have been defined. Put

$$\alpha_i^{\nu+1} = \max_{\mathfrak{P}_i, q_1, \dots, q_n} A_{i(\mathfrak{A}^{(\nu)})}^{\otimes}(p_1, \dots, p_k, q_1, \dots, q_n) \quad (i = 1, \dots, n),$$

where the maximum is taken over all possible sets of values of the variables from \mathfrak{P}_i and q_1, \dots, q_n , and, correspondingly,

$$\mathfrak{A}^{(\nu+1)} \equiv q_1 \vee \overline{\alpha_1^{\nu+1}} \lfloor_{j_1} \dots q_n \vee \overline{\alpha_n^{\nu+1}} \lfloor_{j_n} \mathfrak{A}(p_1, \dots, p_k, A_1 \rfloor_{j_1}, \dots, A_n \rfloor_{j_n}).$$

It is obvious that for every $i = 1, 2, \dots, n$ and $\nu = 1, 2, \dots$, $\alpha_i^\nu \rightarrow \alpha_i^{\nu+1}$; **therefore there exists a natural number μ such that for all $i = 1, 2, \dots, n$, $\alpha_i^{\mu+1} \equiv \alpha_i^\mu$.** We shall call the program scheme $\mathfrak{A}^{(\mu)}$ the stationary immersion** of the program scheme \mathfrak{A} under the distribution of shifts (3).

Definition 10. We shall call program schemes $\mathfrak{A}(p_1, \dots, p_k)$ and $\mathfrak{B}(p_1, \dots, p_k)$ **weakly equivalent** if, for any sequence of the form $\Delta_s, \Delta_s, \dots, \Delta_s, \dots$, the first operators of their values coincide; moreover, if the value of one of them is an empty cycle or is empty, then the value of the other program scheme must be the same.

Theorem. In order that program schemes $\mathfrak{A}(p_1, \dots, p_k)$, $\mathfrak{B}(p_1, \dots, p_k)$ be equivalent under the given distribution of shifts (3), it is necessary and sufficient that their stationary immersions be weakly equivalent.

Similarly, for each program scheme $\mathfrak{A}(p_1, \dots, p_k, A_1, \dots, A_n)$ one can construct such a system of functions $\bar{\beta}_0, \bar{\beta}_1, \dots, \bar{\beta}_n$ that the question of partial equivalence of the program scheme will be reduced to the question of weak equivalence of a program scheme of the form

$$q_0 \vee \bar{\beta}_0 \lfloor_{j_0} \dots q_n \vee \bar{\beta}_n \lfloor_{j_n} \rfloor_t 0 \lfloor_t \rfloor_{j_0} \mathfrak{A}(p_1, \dots, p_k, A_1 \rfloor_{j_1}, \dots, A_n \rfloor_{j_n}).$$

The question of identity transformations of program schemes is reduced to the construction of a complete system of elementary transformation rules. For this it is first necessary to introduce an effective notion of subordination of an e.v. to a logical function, whose substantive meaning is as follows: if an e.v. \mathfrak{B} in the program scheme \mathfrak{A} is subordinated to the function α under the distribution of shifts (3), then for any admissible sequence the equality $\alpha(\Delta_s) = 1$ is a necessary condition for the fulfillability of \mathfrak{B} on the set Δ_s .

* If \mathfrak{A} does not contain some of the operators A_1, \dots, A_n , then the corresponding right primes \rfloor_{j_r} in construction

** I.e. $(\alpha_i^\nu \rightarrow \alpha_i^{\nu+1}) \equiv 1$.

Definition 11. We shall call an **expression** any finite string composed of distinct symbols of operators, logical conditions, and right strokes, in such a way that for every natural number i it contains no more than one left and no more than one right stroke with index i .

Obviously, a p.s. is an expression.

The following system of axiom schemes and rules of inference (where $\mathfrak{M}, \mathfrak{N}$ are arbitrary expressions; $\mathfrak{D}(\mathfrak{M})$ is an expression formally containing the expression \mathfrak{M} ; A_k is an arbitrary operator; $\alpha, \beta, 0, 1$ are formulas of the classical propositional calculus) is complete in the sense of the derivability of every formula of the form $\mathfrak{A} = \mathfrak{B}$, true for a given distribution of shifts, where \mathfrak{A} and \mathfrak{B} are p.s.

$$\text{I. } \begin{array}{l} 1) \ 0 \lfloor_i A_k \rfloor_i = 0 \lfloor_i \rfloor_i; \\ 2) \ 1 \lfloor_i \mathfrak{M} \rfloor_i = \mathfrak{M}; \\ 3) \ \rfloor_i \mathfrak{M} 1 \lfloor_i = \mathfrak{M}. \end{array}$$

$$\text{II. } \begin{array}{l} 1) \ \alpha \&\beta \lfloor_i \mathfrak{M} \rfloor_i = \alpha \lfloor_i \beta \rfloor_j \mathfrak{M} \rfloor_i \rfloor_j; \\ 2) \ \rfloor_i \mathfrak{M} \alpha \&\beta \lfloor_i = \rfloor_i \rfloor_j \mathfrak{M} \alpha \lfloor_i \beta \rfloor_j; \\ 3) \ \alpha \vee \beta \lfloor_i = \alpha \lfloor_i \beta \rfloor_j. \end{array}$$

$$\text{III. } \mathfrak{M} \mathfrak{N} = 0 \lfloor_i \rfloor_j \mathfrak{N} 0 \lfloor_k \rfloor_i \mathfrak{M} 0 \rfloor_j \rfloor_k.$$

$$\text{IV. } \rfloor_i \rfloor_j = \rfloor_j \rfloor_i.$$

$$\text{V. } \alpha \lfloor_i \rfloor_j = \Lambda, \quad \text{where } \Lambda \text{ is the empty scheme.}$$

$$\text{VI. } \begin{array}{l} 1) \ \alpha \lfloor_i \mathfrak{M} \rfloor_j \alpha \lfloor_i \mathfrak{N} \rfloor_j = \alpha \lfloor_i \mathfrak{M} \alpha \lfloor_i \mathfrak{N} \rfloor_j \rfloor_j; \\ 2) \ \alpha \lfloor_i \mathfrak{M} \rfloor_j \mathfrak{N} \rfloor_i \alpha \rfloor_j = \alpha \lfloor_i \mathfrak{M} \rfloor_j \rfloor_i \mathfrak{N} \alpha \rfloor_j; \\ 3) \ \rfloor_j \mathfrak{M} \alpha \lfloor_i \mathfrak{N} \rfloor_j \alpha \lfloor_i = \rfloor_j \rfloor_i \mathfrak{M} \alpha \lfloor_i \mathfrak{N} \alpha \rfloor_j; \\ 4) \ \rfloor_i \alpha \rfloor_j \mathfrak{M} \alpha \lfloor_i \mathfrak{N} \rfloor_j = \alpha \rfloor_j \mathfrak{M} \alpha \lfloor_i \mathfrak{N} \rfloor_j \rfloor_i; \\ 5) \ \rfloor_i \alpha \rfloor_j \mathfrak{M} \rfloor_j \mathfrak{N} \alpha \lfloor_i = \alpha \rfloor_j \mathfrak{M} \rfloor_j \rfloor_i \mathfrak{N} \alpha \lfloor_i; \\ 6) \ \rfloor_j \mathfrak{M} \rfloor_i \alpha \rfloor_j \mathfrak{N} \alpha \lfloor_i = \rfloor_j \rfloor_i \mathfrak{M} \alpha \rfloor_j \mathfrak{N} \alpha \lfloor_i. \end{array}$$

$$\text{VII. } \rfloor_j \alpha \lfloor_i \mathfrak{M} \rfloor_i \alpha \rfloor_j = \alpha \lfloor_i \mathfrak{M} \rfloor_i \rfloor_j \alpha \rfloor_j.$$

$$\text{VIII. } \frac{\alpha = \beta}{\mathfrak{M}(\alpha) = \mathfrak{M}(\beta)}.$$

$$\text{IX. } \frac{\mathfrak{M} = \mathfrak{N}, \ \mathfrak{D}(\mathfrak{M}) = \mathfrak{R}}{\mathfrak{D}(\mathfrak{N}) = \mathfrak{R}}^*.$$

X. A pair of corresponding strokes \lfloor_i, \rfloor_i , entering into one expression, may be replaced by any other pair \lfloor_j, \rfloor_j , but in such a way that the expression remains an expression.

XI. If the logical condition $\alpha|_i$, for the given distribution of shifts, is subordinated to β , then $\alpha|_i$ may be replaced by $\alpha\&\beta|_i$.

Mathematical Institute named after V. A. Steklov
Academy of Sciences of the USSR

Received
29 IX. 1956

CITED LITERATURE

A. I. Kitov, *Electronic Digital Machines*, Moscow, 1956, p. 193.

* Here it is assumed that $\mathfrak{D}(\mathfrak{N})$ is also an expression.

Note: Figure translations are in progress. See original paper for figures.

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.