



Soviet-era science, translated into English

R. N. Tonoyan

1963

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-196301.76824>

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

Abstract

Full Text

R. N. Tonoyan

SOME EQUIVALENT TRANSFORMATIONS OF ALGORITHM SCHEMES

(Presented by Academician S. L. Sobolev on 22 X 1962)

In the present paper the concept of an algorithm scheme is defined and some equivalent transformations of schemes are considered. It is assumed that, in the process of executing a scheme, the operators and logical conditions (see (1)) may change according to certain laws. With the aid of these transformations one can pass from an algorithm scheme to computation schemes. Further, an algorithm is given for constructing the scheme of a program (see (2)) realizing the given computation scheme.

Let $L(\alpha)$ be an arbitrary finite ordered set with order relation \prec_α ; $\Lambda = \{\alpha\}$ the set of all possible order relations on L . On Λ we define an order \leq , putting $\alpha \leq \beta$ ($\alpha, \beta \in \Lambda$) if and only if from $l_1 \prec_\alpha l_2$ it follows that $l_1 \prec_\beta l_2$ for all $l_1, l_2 \in L$ comparable under the order \prec_α . The set $L(\alpha) \times N(\alpha')$ with the order \prec_ν : $(l_1, n_1) \prec_\nu (l_2, n_2)$ if and only if $l_1 \prec_\alpha l_2$ and $n_1 \prec_{\alpha'} n_2$, will be called the **order product** of L and N and denoted by $p\{L \times N\}$, and with $\prec_{\nu'}$: $(l_1, n_1) \prec_{\nu'} (l_2, n_2)$, if $l_1 \prec_\alpha l_2$ or $l_1 = l_2$ and $n_1 \prec_{\alpha'} n_2$, will be called the **alphabetic product** $w\{L \times N\}$. Analogously are defined: the order product of the sets L_t ,

$$p \prod_{t=1}^n L_t,$$

and the alphabetic product

$$w \prod_{t=1}^n L_t.$$

We shall say that on the set $L(\alpha)$ a **parametrized system** $(A) = \{\gamma, \Gamma, A, L\}$ is given if the following are given: a completely ordered set $\gamma = \{\xi\}$ of parameters, sets $\Gamma_\xi(\alpha^\xi)$ of values of the parameters, and $A(\xi_1, \xi_2, \dots, \xi_n)$ — a one-to-one correspondence between

$$\Gamma(\nu) = p \prod_{\xi \in \gamma} \Gamma_\xi$$

and L such that if $l_1 \prec_\alpha l_2$ and $A(s_1) = l_1$, $A(s_2) = l_2$ ($s_1, s_2 \in \Gamma$), then $s_1 \prec_\nu s_2$.

Let there be given sets of parametrized systems $\mathfrak{M}_0 = \{(A)\}$ –operators of rank 0, $P = \{(p \uparrow)\}$ –logical conditions, $I = \{(\downarrow)\}$ of lower arrows, and $\widetilde{\mathfrak{M}}_0 = \mathfrak{M}_0 \cup P \cup I$. We shall call any symbol \mathfrak{A} obtained from $h(\xi_1, \xi_2, \dots, \xi_n)$, where $(h) \in \widetilde{\mathfrak{M}}_0$, by substituting, in place of some parameters, values from the corresponding sets of variation, an **elementary expression**. We shall consider that it depends only on the remaining parameters.

Define an **expression of rank 0 on \mathfrak{M}_0** : every elementary expression is an expression of rank 0; if \mathfrak{A}_1 and \mathfrak{A}_2 are expressions of rank 0, then $\mathfrak{A} = \mathfrak{A}_1 \mathfrak{A}_2$ is an expression of rank 0, with

$$\gamma(\mathfrak{A}_1) \cup \gamma(\mathfrak{A}_2) = \gamma(\mathfrak{A}),$$

where $\gamma(\mathfrak{A})$ is the set of parameters on which \mathfrak{A} depends.

Let sets of symbols $\Gamma_{\gamma[\cdot, \Gamma[\cdot, \gamma], \Gamma]}$ be specified, where $\gamma[$ and $\gamma]$ are completely ordered sets of parameters, and $\Gamma[$ and $\Gamma]$ are sets of their values. By induction we define expressions of rank k :

1. Every expression of rank $(k - 1)$ is an expression of rank k .
2. Any symbols $\Gamma_{\gamma[\cdot, \Gamma[\cdot, \gamma], \Gamma]}$ are expressions of rank k .
3. If \mathfrak{A}_1 and \mathfrak{A}_2 are expressions of rank k , then $\mathfrak{A} = \mathfrak{A}_1 \mathfrak{A}_2$ is an expression of rank k , with $\gamma(\mathfrak{A}) = \gamma(\mathfrak{A}_1) \cup \gamma(\mathfrak{A}_2)$.

An **operator of the k -th rank** shall mean an expression of the k -th rank satisfying the conditions:

- a) for every $p \uparrow$ there exists a finite system of lower arrows

$$C = \{\downarrow, \downarrow, \dots, \downarrow\}$$

and for all $\downarrow \in C$:

$$\gamma_{p\uparrow} = \gamma_{\downarrow}, \quad \Gamma_{\downarrow} \subseteq \Gamma_{p\uparrow}, \quad \Gamma_{p\uparrow} = \bigcap_{\downarrow \in C} \Gamma_{\downarrow},$$

and the Γ_{\downarrow} are pairwise disjoint.

- b) every \downarrow belongs to one and only one system C .
- c) for every $[\gamma_{\uparrow}, \Gamma_{\uparrow}]$ there exists a unique $]\gamma_{\downarrow}, \Gamma_{\downarrow}$, located to its right and such that $\gamma_{\uparrow} = \gamma_{\downarrow}$, $\Gamma_{\uparrow} = \Gamma_{\downarrow}$, and conversely.

We shall call the **scope** of $[\gamma_{\uparrow}, \Gamma_{\uparrow}]$ the expression lying between the corresponding brackets. We denote it by $\mathfrak{b}([\gamma_{\uparrow}])$.

A parameter ξ shall be called **bound** in an expression \mathfrak{A} of the k -th rank if all $h \in \mathfrak{A}$ containing the parameter ξ enter the scope $\mathfrak{b}([\gamma_{\uparrow}])$, where $\xi \in \gamma_{\uparrow}$.

ξ shall be called a **free parameter** if in \mathfrak{A} there does not exist $[\gamma_f, \Gamma_f]$ such that $\xi \in \gamma_f$.

d) Every parameter is either bound or free.

e) there does not exist $[\gamma_f, \Gamma_f \subset \mathfrak{b}(\downarrow, \gamma_f)]$ such that $\gamma_f \cap \gamma_{f'} \neq \emptyset$.

Let $z(h)$ be the set of all $[\gamma_f, \Gamma_f]$ in whose scope the expression h enters.

f)

$$z(\downarrow) = z(\uparrow)$$

for corresponding brackets, and

$$z(\downarrow) \subset z(p \uparrow)$$

for all $\downarrow \in C$.

By $\sigma(\mathfrak{A})$ we denote the set of all bound parameters of \mathfrak{A} , by $\Sigma_{\mathfrak{A}}$ the set of values of these parameters, by $\sigma'(\mathfrak{A})$ the set of free parameters, and by $\Sigma'_{\mathfrak{A}}$ the values of these parameters.

A **scheme of an algorithm** shall mean an operator of arbitrary rank, and an expression of the same rank shall be called an **expression in the scheme of the algorithm**. Let an algorithm scheme $\mathfrak{A}\{\sigma(\mathfrak{A}), \Sigma_{\mathfrak{A}}; \sigma'(\mathfrak{A}), \Sigma'_{\mathfrak{A}}\}$ be given, and let \prec_{α} be an order relation on the set $\Sigma_{\mathfrak{A}}$. By $K_{\mathfrak{A}}$ we denote the set of all algorithm schemes obtained from \mathfrak{A} by replacing the sets $\Sigma_{\mathfrak{A}}(\alpha)$ by the sets $\Sigma_{\mathfrak{A}}(\nu)$, where \prec_{ν} is a complete order generated by \prec_{α} . Algorithm schemes \mathfrak{A}' and $\mathfrak{A}'' \in K_{\mathfrak{A}}$ shall be called **α -equivalent**, and we shall write

$$\mathfrak{A}' \equiv \mathfrak{A}''(\alpha).$$

For every $\mathfrak{A}' \in K_{\mathfrak{A}}$ we define by induction the process of execution for values $\lambda \in \Sigma'_{\mathfrak{A}}$. We assign to all free parameters the corresponding values from λ and mark the leftmost symbol of the scheme. Suppose that l steps have been carried out, as a result of which a string of symbols has been written out

$$\beta \dots \tau \tag{1}$$

(possibly empty). Then at the $(l+1)$ -st step the symbol marked at the l -th step is considered. 1) If this is an operator of the k -th rank ($0 \leq k < N$), then we append its value on the right to string (1) and mark the symbol immediately following it. 2) If $p \uparrow$, then for $p = 1$ we mark the symbol following it, and for $p = 0$ we mark the symbol standing immediately after $\downarrow \in C$, where the value of the parameter $p \uparrow$ enters Γ_{\downarrow} . 3) If $[\gamma_f, \Gamma_f]$, then in all $h \in \mathfrak{A}'$ we replace the parameters belonging to γ_f by the minimal elements of Γ_f , and mark the symbol following $[\gamma_f, \Gamma_f]$. 4) If \downarrow , then we mark the symbol following it. 5) If $]\gamma_f, \Gamma_f$, then we replace the values of the parameters by the next values from the corresponding sets of change and, if they enter Γ_f , then we mark the symbol following the corresponding bracket; if they do not enter, we mark the symbol

following $]_{\gamma_l, \Gamma_l}$. If there are no marked symbols, then the process terminates. The string written out as a result of this process shall be called the **value of the algorithm scheme** \mathfrak{A}' for the element $\lambda \in \Sigma'_{\mathfrak{A}}$, and we denote it by $\mathfrak{A}'(\lambda)$. If an order \prec_α is specified on $\Sigma_{\mathfrak{A}}$, then the value of the scheme \mathfrak{A} for $\lambda \in \Sigma'_{\mathfrak{A}}$ is set equal to the set $\widetilde{K}_{\mathfrak{A}}(\lambda)$: this is the set of all $\mathfrak{A}'(\lambda)$, $\mathfrak{A}' \in K_{\mathfrak{A}}$. We shall say that a scheme

\mathfrak{A}_1 is less than \mathfrak{A}_2 (denote $\mathfrak{A}_1 \prec \mathfrak{A}_2$) if there exists a one-to-one correspondence φ between $\Sigma_{\mathfrak{A}_1} = \{\lambda_1\}$ and $\Sigma_{\mathfrak{A}_2} = \{\lambda_2\}$: $\varphi(\lambda_1) = \lambda_2$ such that $\widetilde{K}_{\mathfrak{A}_1}(\lambda_1) \subseteq \widetilde{K}_{\mathfrak{A}_2}(\varphi(\lambda_1))$. If $\mathfrak{A}_1 \prec \mathfrak{A}_2$ and $\mathfrak{A}_2 \prec \mathfrak{A}_1$, then we shall call \mathfrak{A}_1 and \mathfrak{A}_2 **equivalent** and denote $\mathfrak{A}_1 \equiv \mathfrak{A}_2$. We shall say that the expression \mathfrak{B}_1 is less than the expression \mathfrak{B}_2 if, for all schemes $\mathfrak{A}(\mathfrak{B}_2)$ containing \mathfrak{B}_2 , $\mathfrak{A}(\mathfrak{B}_1)$ —the expression obtained from $\mathfrak{A}(\mathfrak{B}_2)$ by replacing \mathfrak{B}_2 by \mathfrak{B}_1 —is an algorithm scheme and $\mathfrak{A}(\mathfrak{B}_1) \prec \mathfrak{A}(\mathfrak{B}_2)$. We shall call the expressions \mathfrak{B}_1 and \mathfrak{B}_2 **equivalent** and denote $\mathfrak{B}_1 \equiv \mathfrak{B}_2$, if $\mathfrak{B}_1 \prec \mathfrak{B}_2$ and $\mathfrak{B}_2 \prec \mathfrak{B}_1$. Let \mathfrak{B} be an arbitrary expression; then:

A. If $\gamma_l^1 + \gamma_l^2 = \gamma_l$ and $\Gamma_l = \Gamma_l^1 \times \Gamma_l^2$, then

$$\begin{aligned} [\gamma_l, w\{\Gamma_l^1 \times \Gamma_l^2\} \mathfrak{B}]_{\gamma_l, w\{\Gamma_l^1 \times \Gamma_l^2\}} &\equiv [\gamma_l^1, \Gamma_l^1 [\gamma_l^2, \Gamma_l^2 \mathfrak{B}]_{\gamma_l^2, \Gamma_l^2}]_{\gamma_l^1, \Gamma_l^1} \prec \\ &\prec [\gamma_l, p\{\Gamma_l^1 \times \Gamma_l^2\} \mathfrak{B}]_{\gamma_l, p\{\Gamma_l^1 \times \Gamma_l^2\}}. \end{aligned}$$

B. If $T = \{t\}$ is a completely ordered set and $\Sigma_l = \prod_{t \in T} \Sigma_l^t$, where

$$\sum_{t \in T} \gamma_l^t = \gamma_l,$$

then

$$\begin{aligned} [\gamma_l, w \prod_{t \in T} \Sigma_l^t \mathfrak{B}]_{\gamma_l, w \prod_{t \in T} \Sigma_l^t} &\equiv [{}_{t, T} [\gamma_l^t, \Sigma_l^t]_{t, T} \mathfrak{B} [{}_{t, \bar{T}}]_{\gamma_l^t, \Sigma_l^t}]_{t, \bar{T}} \prec \\ &\prec [\gamma_l, p \prod_{t \in T} \Sigma_l^t \mathfrak{B}]_{\gamma_l, p \prod_{t \in T} \Sigma_l^t} \end{aligned}$$

(\bar{T} is the set dual to the set T).

C. If $\Gamma_l \subset \Gamma'_l$ and the characteristic function $p(\gamma_l)$ of the set Γ_l is given, then

$$[\gamma_l, \Gamma_l \mathfrak{B}]_{\gamma_l, \Gamma_l} \equiv [\gamma_l, \Gamma'_l p(\gamma_l) \uparrow^{\gamma_l} \mathfrak{B} \downarrow^{\gamma_l}]_{\gamma_l, \Gamma'_l}.$$

Every expression $\mathfrak{B} \subseteq \mathfrak{A}$ may be regarded as a finite completely ordered set. Suppose that on this set a parameterized system $(B) = \{\gamma_B, \Gamma_B, B, \mathfrak{B}\}$ is given,

with $\gamma_B \cap \gamma(\mathfrak{A}) = 0$. Let \mathfrak{N} be a finite set of pairwise nonintersecting expressions, on each of which a parameterized system is given. Denote by \mathfrak{A}' the expression obtained from the algorithm scheme \mathfrak{A} by replacing every $\mathfrak{B} \in \mathfrak{N}$ by the expression $[\gamma_B, \Gamma_B]_{\gamma_B, \Gamma_B} B(\gamma_B)$.

If, for all brackets and logical conditions entering into \mathfrak{A}' , $z([\] = z([\])$ and $z(\downarrow) \subseteq z(p \uparrow)$, then \mathfrak{A}' is an algorithm scheme and $\mathfrak{A} \equiv \mathfrak{A}'$.

An operator of the first rank in which: a) for every $[\]$, the set $\gamma_{[\]}$ consists of only one element; b) $\Gamma_{[\]}$ is some interval of the natural number sequence, will be called a **counting scheme**.

Theorem 1. *For every algorithm scheme \mathfrak{A} one can construct a counting scheme \mathfrak{A}^* such that $\mathfrak{A}^* \prec \mathfrak{A}$.*

Let \mathfrak{A}^* be an arbitrary counting scheme. To each expression $\mathfrak{A}' \subseteq \mathfrak{A}^*$ we assign an expression $\overline{\mathfrak{A}'}$ according to the following rules: c1) if $(h) \in \mathfrak{M}_0$, then $\overline{h} = h$; c2) $\overline{[\xi, \Gamma_\xi]} = (\xi \rightarrow \xi_0) \downarrow^\xi$, where ξ_0 is the first element of the set Γ_ξ ; c3) $\overline{[\]_{\xi, \Gamma_\xi}} = F(\xi, \Gamma_\xi) p(\xi \in \Gamma_\xi) \uparrow^\xi$; c4) if $\mathfrak{A}' = \mathfrak{A}_1 \mathfrak{A}_2$, then $\overline{\mathfrak{A}'} = \overline{\mathfrak{A}_1} \overline{\mathfrak{A}_2}$, where $(\xi \rightarrow \xi_0)$ passes ξ_0 in place of ξ , and $F(\xi, \Gamma_\xi)$ passes to the next value of ξ in \mathfrak{A} .

Theorem 2. *From any counting scheme \mathfrak{A}^* , by means of c1)-c4), one can obtain a program scheme $\overline{\mathfrak{A}}$ such that $\overline{\mathfrak{A}} \equiv \mathfrak{A}^*$.*

Received
12 X 1962

REFERENCES

1. A. A. Lyapunov, *Problems of Cybernetics*, issue 8, 1962.
2. A. A. Lyapunov, *Problems of Cybernetics*, issue 1, 1958.

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

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