Use of Standard Sub-Programs in Census Processing Programs

(A paper indited by a group of research workers from the Central Statistical Board — Romania)

This paper presents a system of standard sub-routines used in programming statistical works in view of solving the main problems of logical checking and table making. This system is a first step towards an oriented language.

GENERAL PROBLEMS APPEARING IN DATA PROCESSING

Introductory concepts

Let P be a given set which will be called, in general, a population. We shall call indicator a function ρ defined as follows:

$$\rho : P \rightarrow M_\rho$$

where M_ρ (the set of values of indicator ρ) is a set of integer, rational or real numbers (codes) or, in other words, a union of closed intervals.

If $\rho_1, \ldots, \rho_n$ are the n indicators according to which the census is done, and p the generic element of set P, the information unit (record) to be processed, corresponding to individual p, will be expressed by the set $\{\rho_1(p), \ldots, \rho_n(p)\}$

Logical validation of data

During the processing, various errors may occur in the stages of annual processing (data collection, coding, punching, etc.). This is the reason why an analysis of data indicator by indicator is required. Such a control may be used in order to detect a number of inconsistencies such as: the verified indicator does not fall within acceptable limits or the value of the indicator

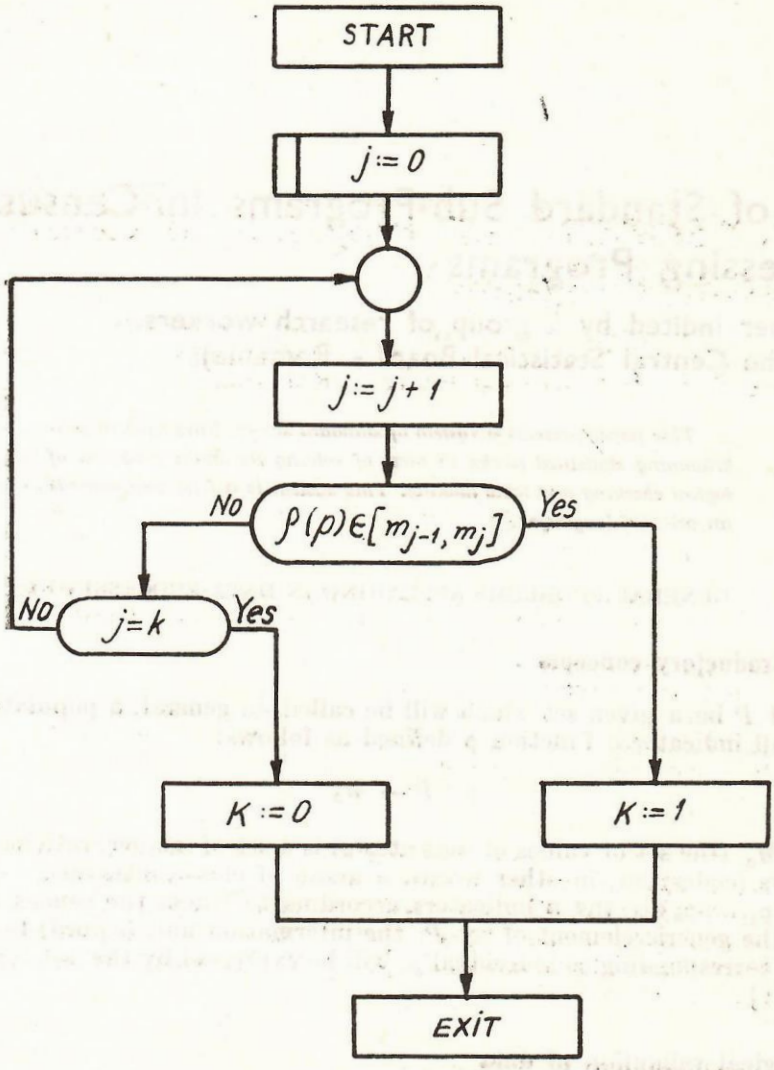

Fig. 1 *Checking indicator limits*

is not consistent with the values of their indicators. Depending on the inconsistency found or the implications it may have, a decision is made: the record is discarded or automatically corrected.

A study of the methodological indications for the logical control of the population and housing census data has shown that many of the controls carried out may be standardized, their indicators and values becoming simple parameters of some standard procedures, the way we adopted in order to solve these general problems will be shown in the next sections.

Control of Limits

The simplest and absolutely necessary control to be effected for the value of each indicator is to check whether it belongs to the set of acceptable values; but since this set is, in the most general case, a union of disjoint intervals of real, rational or integer numbers, the problem amounts to checking whether it belongs to one of the intervals, that is its affiliation to this interval.

Examples from the census are:

— the region code may take values within the interval [1, 18];

— the locality code may take values in the set

$$[1,\ 5] \cup [11,\ 16] \cup [21,\ 39] \cup [41,\ 89] \cup [91,\ 94]$$

In general, let be an indicator whose set of values is M_ρ

Admitting that the indicator may, at a given time, take the value V, the object of our problem is to determine the truth value of the proposition:

$$V \in M_\rho$$

In case the proposition is false a decision is to be made, for instance:

— the information concerning the individual is rejected;

— the indicator is automatically corrected by one of the acceptable values.

In general, designating the decision by D and the truth value of proposition $\mathcal{P}$ by $L(\mathcal{P})$, it can be written:

$$(L((\exists)j: ((j \in [1, k) \wedge (\rho(p) \in [m_{j-1}, m_j])) = 0) \Rightarrow D$$

For the control of limits a standard sub-routine has been written having as parameters: ρ, k, M_ρ, K, where K is a bivalent variable whose value is equal to the truth value of the analyzed proposition. The programmer may take the decision D in accordance with the value of K.

The logical flow-chart of this sub-routine is given in Fig. 1.

Coding of intervals

The stage of logical validation of data implies not only a check on the inclusion of indicators into a certain logical pattern but also the preparation of information for subsequent processing. One way of preparing it is to construct some additional indicators deriving them from some basic indicators. An indicator of this kind is, for example, GV2 which partitions the population by age groups in accordance with the requirements of statistical tables:

GV2 = 1 if age is smaller than 12 years;

GV2 = 2 if age ranges from 13 to 15 years;

GV2 = 3 if age ranges from 16 to 20 years, etc.

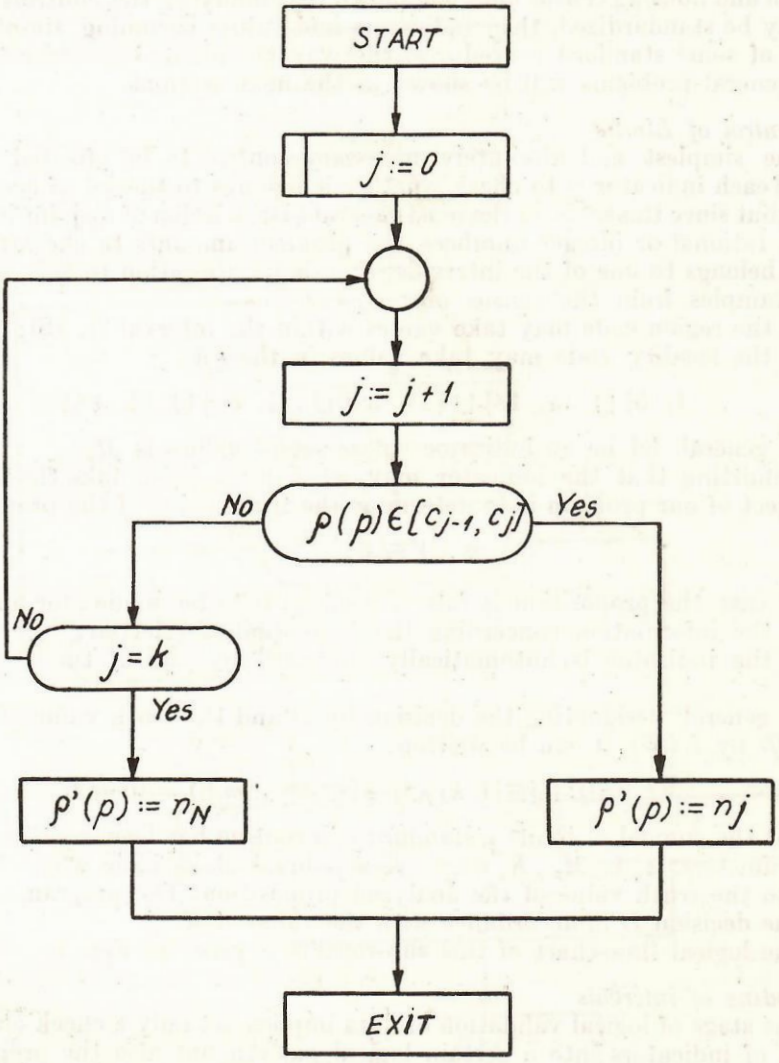

Fig. 2 *Coding of intervals*

The problem posed may be expressed thus:

Let ρ be the basic indicator with the set of values $M_\rho = \bigcup_{j=1,n}[m_{j-1}, m_j] \geqslant C_\rho$ where C_ρ is the set of intervals to be codified:

$$C_\rho = \bigcup_{i=1,k}[C_{i-1}, C_1]$$

Obviously:

$$((\forall)i \in [1, k]) \rightarrow ((\exists)j \in [i, n] : [C_{i-1}, C_i] \subseteq [m_{j-1}, m_j])$$

Let ρ' be the indicator which is produced with the set of values

$$N_{\rho'} = \{n_j\}_{j=1,h} \cup \{n_N\}$$

Clearly, there is a bijective correspondence between the intervals of the set C_ρ and the elements of the set $\{n_j\}$. The coding is made according to the following relations:

$$(L((\exists)j \in [1, k] : (\rho(p) \in [C_{j-1}, C_j])) = 1) \Rightarrow (\rho(p) := n_j)$$

$$L((\forall)\ j \in [1, k]) \rightarrow (\rho(p) \notin [C_{j-1}, C_i])) = 1 \Rightarrow (\rho'(p) := n_N)$$

In order to implement the coding of intervals a sub-routine CODLM (flow-chart in Fig. 2) has been written which admits as parameters: ρ, n, t values n_j, C_{j-1}, δ', n_N.

Searching in the lists

There are cases where an indicator may have a relatively large set of values included in a list; for instance, the code for the national economy branch might take more than 300 values, the set of values for the codes of the villages of a region has more than 1000 elements, etc. It is obvious that in such cases the limit control becomes inefficient. The solution which was chosen was to store the whole list (in ascending order) in the form of a table and to use a searching sub-routine which should give the position of the current value of the indicator within the list (if this value is not included into the list the resulting position would be 0).

The method of dividing the interval into halves (scheme in Fig. 3 for sub-routine CAUT) was used as a method for searching within the list.

Implementation of the Logical AND/OR Functions

So far the logical control of indicators was considered in its simplest aspects, namely the control carried out indicator by indicator each one of them being considered independently of the others. However, relations of a much more complex nature and chain correlations may occur, in reality, which enables a decision to be made in accordance with the values, at a given time, of a set of indicators.

Here is an example of simple correlation:

"If the sex is female AND the marital status single, then the year of marriage will take the value of 'undeclared'". Or:

"If the number of live born children is declared OR the year of marriage is declared, then the sex will be female".

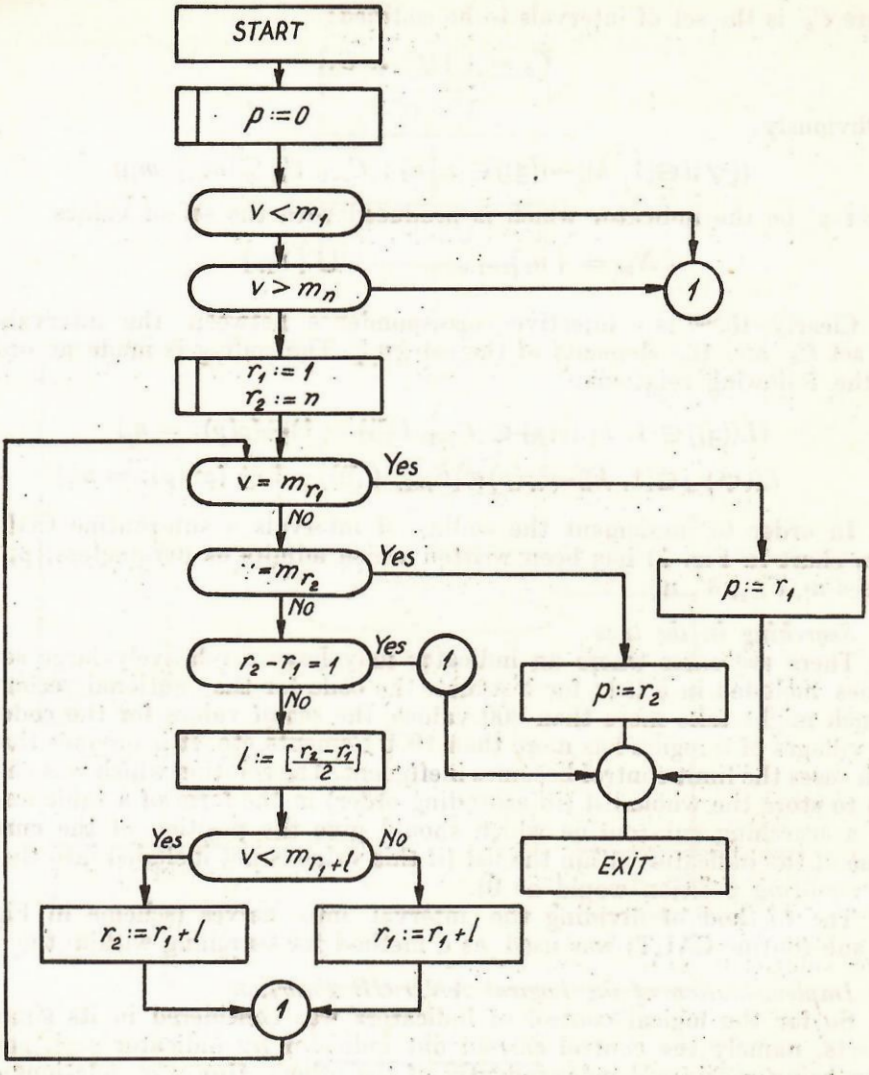

Fig. 3 *Searching in the lists*

Significance of symbols

p resultant position
v value of searched for indicator
m_1 first value in the list
m_n last value in the list
n rumber of values in the list
r_1 lower value of interval in the list under study
r_2 higher value of interval in the list under study
m_j value at position j of list
l length of interval to be studied

In general, let $\mathcal{J} = \{\rho_i\}_{i \in I}$ be a set of indicators, V_i their sets of values and

$$V_I = \{ v_{j1}, ..., v_{ik}, ... \}_{i1}, ... i_k, ..., \in i$$

an element of the Cartesian product $\prod_{i \in I} V_i$; let ρ be another indicator and v its value at a given time. The logical AND function is defined by:

$$(\bigwedge_{i \in I}(L(\rho_i(p) = v_i)) = 1) \Rightarrow (\rho(p) := v)$$

Similarly, the logical OR is:

$$(\bigvee_{i \in I}(L(\rho_i(p) = v_i)) = 1) \Rightarrow (\rho(p) := v)$$

The sub-routines SPSI (logical AND) and SPSAN (logical OR) having as parameters: k, ρ_{i1} ..., ρ_{ik}, v_{i1}, ..., v_{ik}, ρ, v have been used for implementing such controls.

The flow-charts of these sub-routines are given in Figs. 4 and 5.

Multiple correlations and corrections

One can easily imagine the set of indicators in an AND-type correlation, the logical OR respectively, in the form of some chains which are determining another indicator. It can be said, in a more suggestive way, that in the case of the logical AND function an indicator can be determined by a set of other indicators which are linked *in series* while in the case of the logical OR function they are linked *in parallel.*

With the help of these basic intuitive elements, a quick analysis of a number of much more elaborate correlations consisting of chains of series indicators, which are linked in parallel, can be conceived.

Such a correlation may be expressed, in general, as follows:
"if indicator ρ_1 has value v_{11}
AND indicator ρ_2 has value v_{12}
AND .....
AND indicator ρ_m has value v_{1m}
then:
indicator ρ'_1 should have value v'_{11}
AND indicator ρ'_2 should have value v'_{12}
AND .....
AND indicator ρ'_n should have value v'_{1n}
AND indicator ρ''_2 should be equal to indicator ρ''_{11}
AND indicator ρ''_2 should be equal to indicator ρ''_{12}
AND .....
AND indicator ρ''_P should be equal to indicator ρ''_{1P}
OR:
if indicator ρ_1 has value v_{21}
AND indicator ρ_2 has value v_{22}
AND .....
AND indicator ρ_m has value v_{2m}

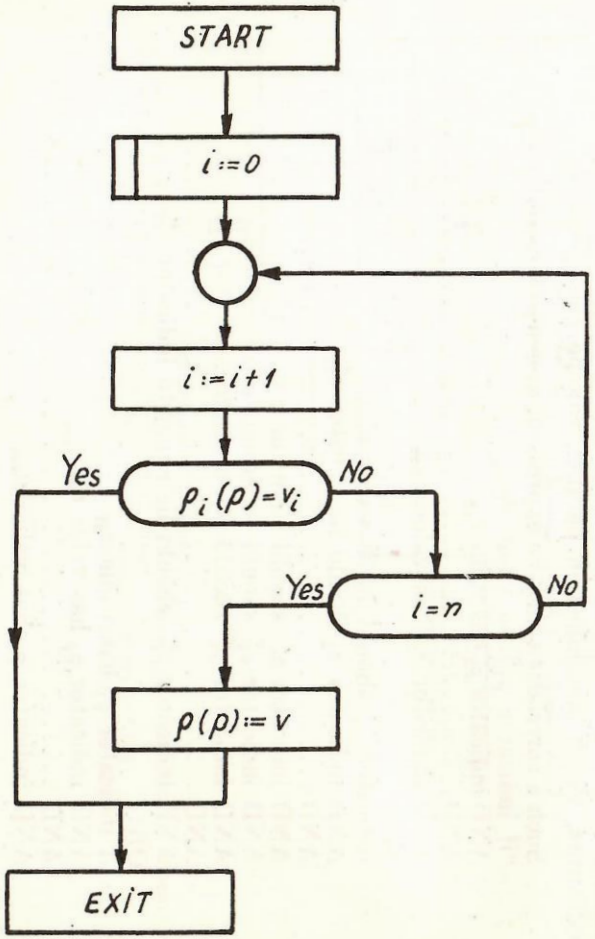

Fig. 4 *Implementation of logical AND function*

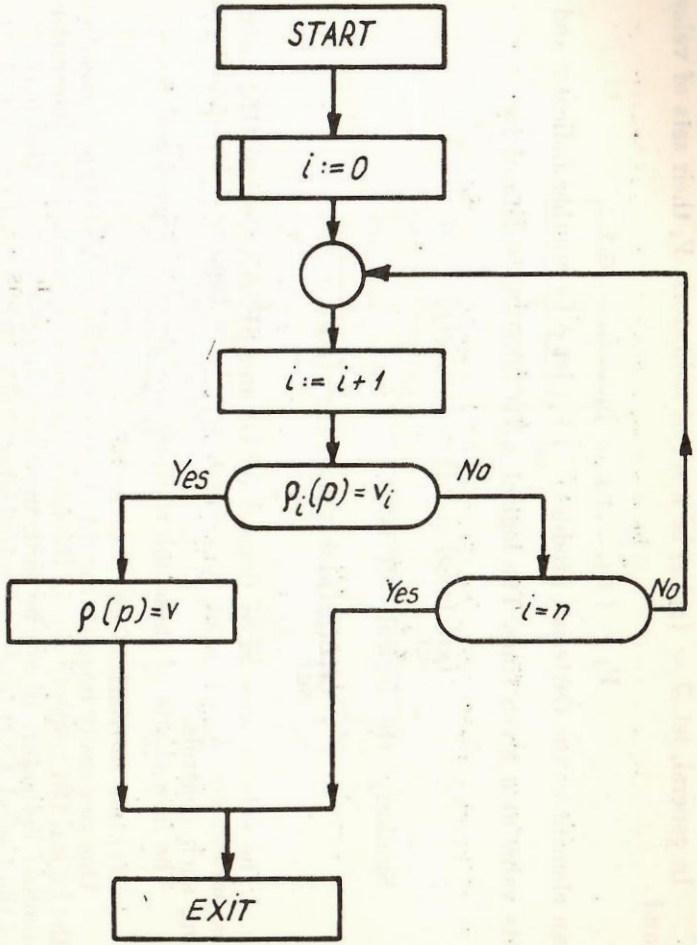

Fig. 5 *Implementation of logical OR function*

then:

indicator ρ'_1 should have value v'_{21}
AND indicator ρ'_2 should have value v'_{22}
AND .....
.....
.....
.....
AND indicator ρ'_n should have value v'_{2n}
AND indicator ρ''_1 should be equal to indicator ρ''_{11}
AND indicator ρ''_2 should be equal to indicator ρ''_{21}
AND .....
.....
AND indicator ρ''_P should be equal to indicator ρ''_{2P}
OR .....

In other words, given the sets of indicators $\{\rho_1, ..., \rho_m\}$, $\{\rho'_1, ..., \rho'_n\}$ and $\{\rho''_1, ..., \rho''_m\}$ to which the following matrices of values are respectively assigned:

$$V = \begin{pmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & & \\ v_{q1} & \cdots & v_{qm} \end{pmatrix}; \quad V' = \begin{pmatrix} v'_{11} & \cdots & v'_{in} \\ \vdots & & \\ v'_{q1} & \cdots & v'_{qn} \end{pmatrix}; \quad V'' = \begin{pmatrix} \rho''_{11} & \cdots & \rho''_{1P} \\ \vdots & & \\ \rho''_{q1} & \cdots & \rho''_{qP} \end{pmatrix}$$

The correlation consists in scanning matrix V and as soon as the values of indicators $\rho_1, ... \rho_m$ entirely fulfill a line, the line of the same order of V' and V'' should be fulfilled by the values of indicators $\rho'_1, ..., \rho'_n$ and $\rho''_1, ..., \rho''_P$ respectively; otherwise a decision will be taken which may consist in:

— rejection:
— automatic correction of the record;
— continuing to scan the next correlation chains which follow.

The set $\{\rho_1, ..., \rho_m\}$ will be called the set of indicators that *control* the correlation, $\{\rho'_1, ..., \rho'_n\}$ will be the set of indicators correlated with constants and $\{\rho'_1, ..., \rho''_P\}$ the set of indicators correlated with other indicators.

We shall try to give a mathematical expression of the above specifications.

Given the set of indicators:

$$\mathcal{J} = \{\rho_i\}_{i \in \mathcal{K}}$$
$$\mathcal{J}' = \{\rho'_i\}_{i \in \mathcal{K}'}$$
$$\mathcal{J}'' = \{\rho''_i\}_{i \in \mathcal{K}''}$$

(one of the $\mathcal{J}' \mathcal{J}''$, may also be the empty set) each indicator ρ_i being associated to a set of values $M\rho_i$.

Let $\mathcal{V}$, $\mathcal{V}'$, $\mathcal{V}''$ be sets which satisfy the following conditions;

$$\mathcal{V} \subseteq \prod_{i \in \mathcal{K}} M\rho_i$$
$$\mathcal{V}' \subseteq \prod_{i \in \mathcal{K}'} M\rho'_i$$
$$\mathcal{V}'' \subseteq \prod_{i \in \mathcal{K}''} M\rho''_i$$

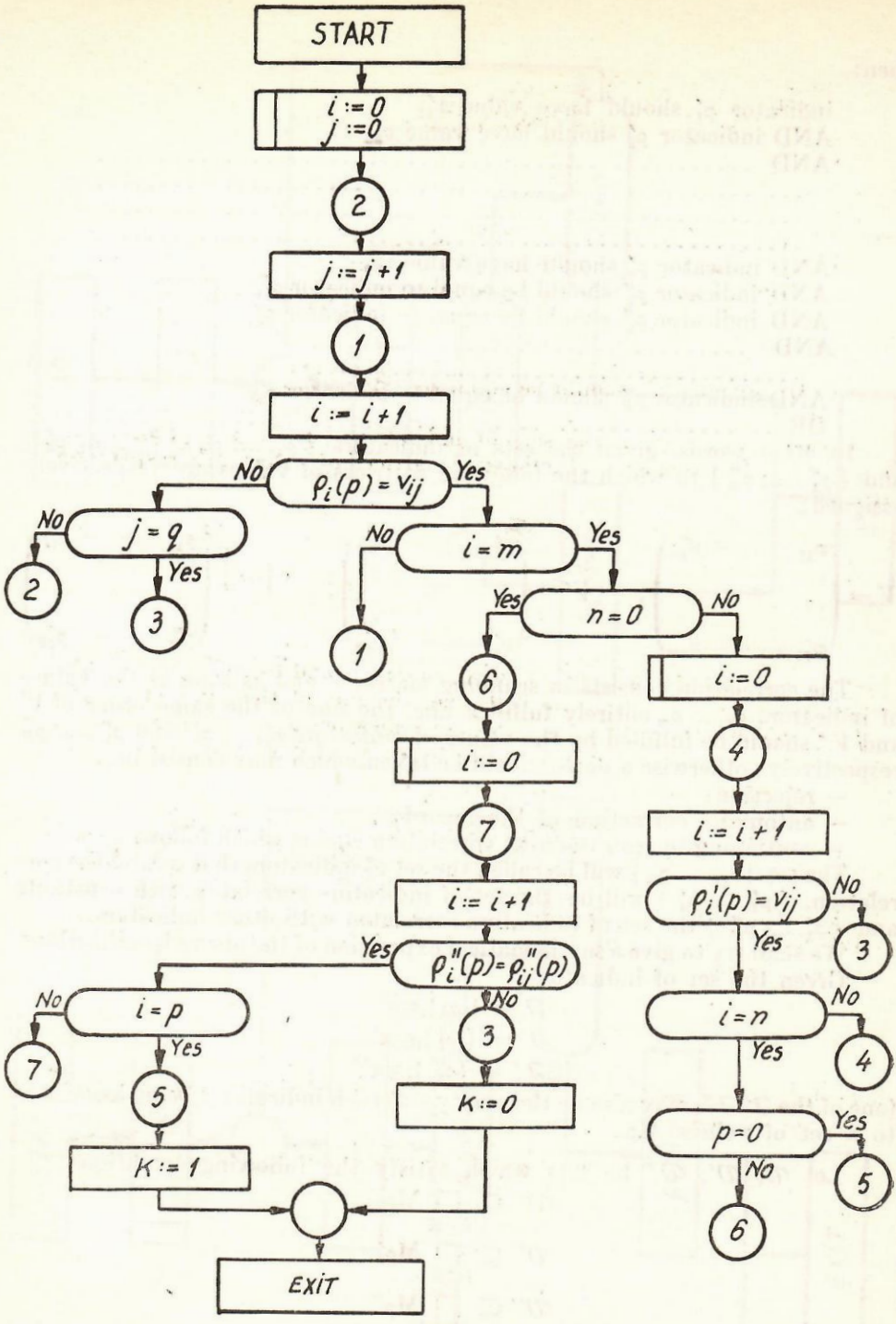

Fig. 6 *Multiple correlation*

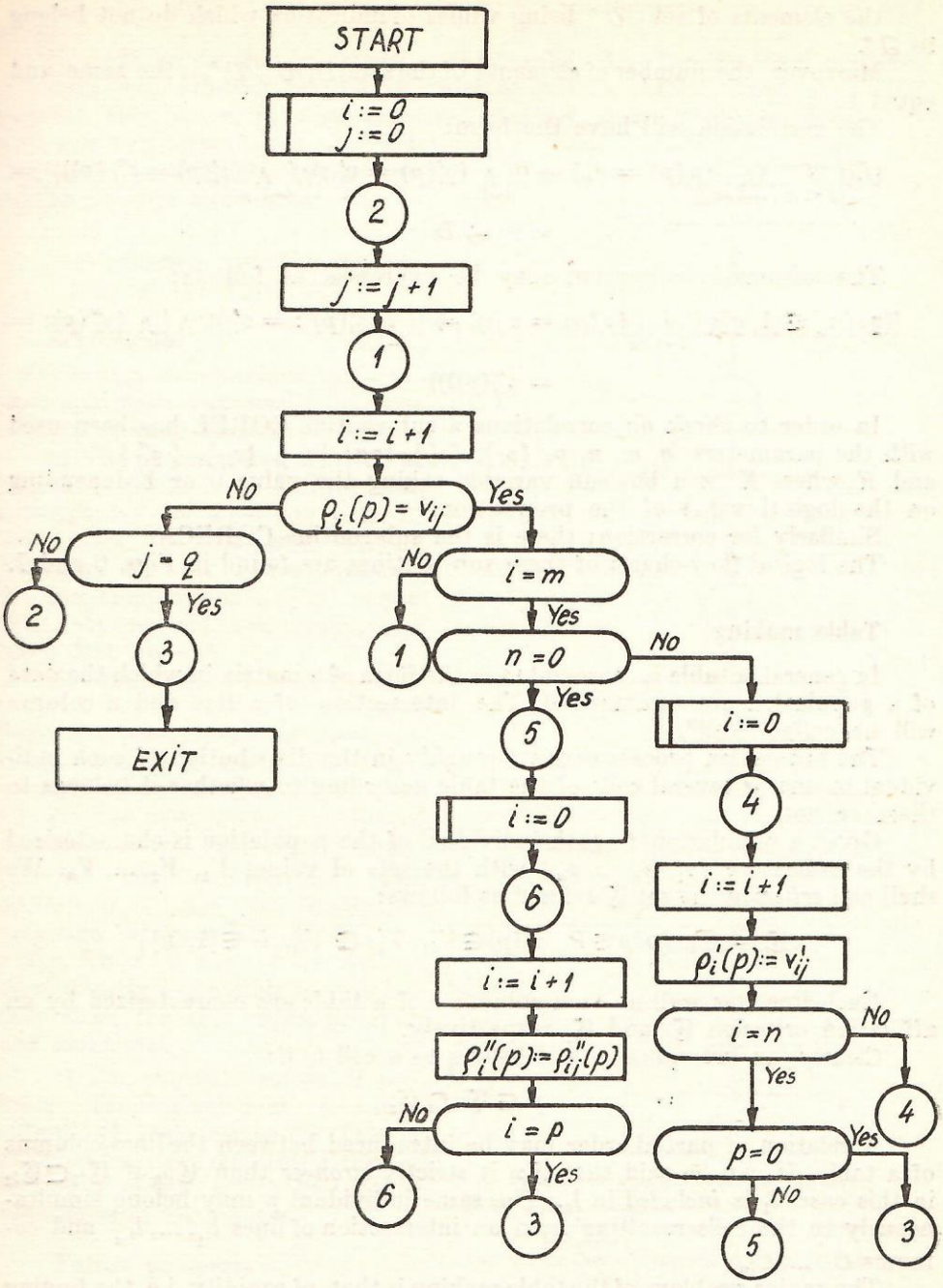

Fig. 7 *Multiple correction*

the elements of set $\mathcal{V}''$ being values of indicators which do not belong to $\mathcal{J}''$.

Moreover, the number of elements of the sets $\mathcal{V}, \mathcal{V}', \mathcal{V}''$ is the same and equal to q.

The correlation will have the form:

$$(L(\bigvee_{j=1,q}(\bigwedge_{i=1,m}(\rho_i(p)=v_{ij})\rightarrow((\bigwedge_{i=1,n}(\rho_i'(p)=v_{ij}'))\wedge(\bigwedge_{i=1,p}(\rho_i'(p)=\rho_{ij}''(p))))=$$

$$=0)\Rightarrow D$$

The automatic correction may be expressed as follows:

$$((\exists)j:j\in[1,q]\wedge(\bigvee_{i=1,m}(\rho_i(p)=v_{ij})))\Rightarrow((\bigvee_{i=1,n}(\rho_i'(p):=v_{ij}'))\wedge(\bigwedge_{i=1,p}(\rho_i''(p):=$$

$$=\rho_{ij}''(p))))$$

In order to check on correlations a sub-routine COREL has been used with the parameters q, m, n, p, $\{\rho_i\}$, $\{\rho_i'\}$, $\{\rho_i''\}$, $\{v_{ij}\}$, $\{v_{ij}\}$, $\{\rho_{ij}''\}$ and K, where K is a boolean variable taking the value 0 or 1 depending on the logical value of the proposition.

Similarly for corrections there is the sub-routine COREC.

The logical flow-charts of these sub-routines are found in Figs. 6 and 7.

Table making

In general, a table is represented in the form of a matrix in which the data of a population are summarized. The intersection of a line and a column will be called "cell".

The tabulation process consists roughly in the distribution of each individual in one or several cells of the table according to whether it belongs to them or not.

Given a population P, each individual of the population is characterized by the indicators $\{\rho_1, \rho_2, ..., \rho_n\}$ with the sets of values $V_1, V_2, ..., V_n$. We shall call *criterion* the set $\mathfrak{C}$ defined as follows:

$$\mathfrak{C}=\bigcap_{j=1,k}\{p/p\in P,\ \rho_{ij}(p)\in V_i',\ V_{ij}'\subset V_{ij},\ i_j\in[1,n]\}$$

Each line l as well as each column c of a table are characterized by an affiliation criterion $\mathfrak{C}_l$ and $\mathfrak{C}_c$ respectively.

Clearly, an individual p will belong to a cell lc if:

$$p\in\mathfrak{C}_l\cap\mathfrak{C}_c$$

A relation of partial order may be introduced between the lines/columns of a table. It can be said that $\mathfrak{C}\rho_i$ is strictly *stronger* than $\mathfrak{C}\rho_2$ if $\mathfrak{C}_{l_1}\subseteq\mathfrak{C}_{l_2}$ in this case, l_1 is *included* in l_2. The same individual p may belong simultaneously to the cells resulting from an intersection of lines $l_{i_1}, ..., l_{i_m}$ and columns $C_j, ..., C_{jn}$

The major problem of the table making is that of rapidity, i.e. the finding of a solution to a simultaneous tabulation of an individual into the cell to which it belongs.

For this purpose the sub-routine SVENS (flow-chart in Fig. 8) was written.

In addition to this sub-routine which solves the fundamental problem of tabulation, sub-routines TABA, SUBA, ADSL, ADDL have also been written. They solve, in a standardized manner, other problems connected with the tabulation such as: writing tables on magnetic tape, summation of core store areas, etc.

CONCLUSIONS ABOUT THE USE OF SUB-ROUTINES

Advantages

The use of standard sub-routines has been extremely efficient in programming because:

a) The program was reduced to sequences of calls of sub-routines accompanied by adequate lists of parameters. In this way writing programs for the census became not too complicated a job, requiring unspecialized knowledge, and being fast enough.

b) Important store economies (approximately 30 per cent) have been obtained. However, these memory economies have sometimes involved a certain loss in run-time e.g. (in the case where the list of parameters was extremely small)

All the sub-routines were written in PLAN, the assembler language of the ICT 1905 computer. They all have been written as efficiently as possible in order to save store for the object-program and run-time.

c) The standardization of programs, the replacement, to a high extent, of instructions by lists of parameters, have facilitated the detecting of errors, which resulted in economies in testing time.

Future prospects

The use of sub-routines has suggested the automation of pro-

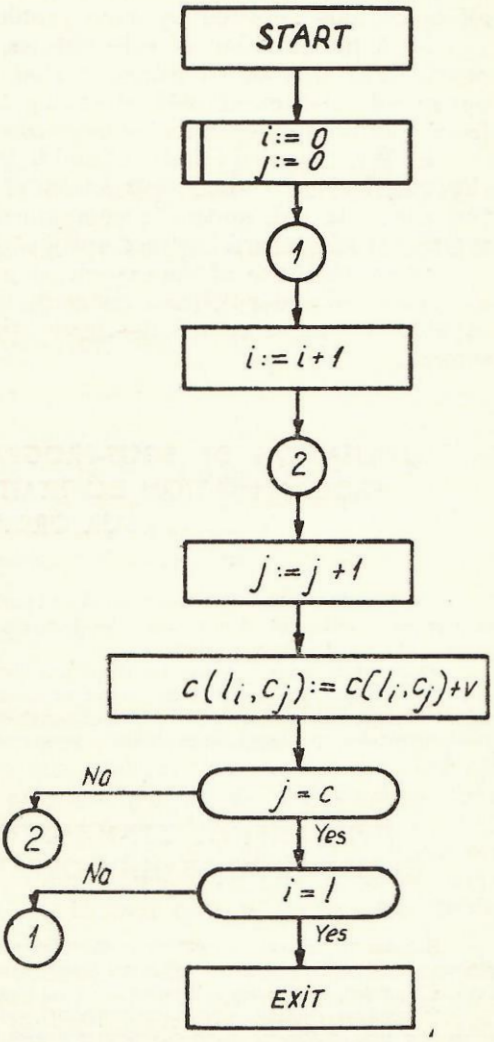

Fig. 8 *Simultaneous tabulation*

Significance of symbols

l =	number of lines in which tabulation is performed
c =	number of columns in which tabulation is performed
$c(\alpha, \beta)$ =	cell resulting from the intersection of line and column
v =	tabulated value
i, j =	current indicators

gramming statistical problems, namely the construction of a language directed towards such problems.

To this end, the following should be implemented:

a) An analysis of statistical problems, and on the basis of conclusions obtained, the vocabulary, syntax and semantic rules of the language should be defined. In general, it can be stated that the sub-routine system fulfills all operations required by such problems.

b) A modification of sub-routines, the selection of their common parts constituting new sub-routines, so that a set of sub-routines linked together, optimized, and minimized according to the memory criterion is obtained from a series of independent sub-routines.

c) What was said under a) and b) leads to a first solution to the translator: the translation of each instruction of the source — language by a call of the adequate sub-routine accompanied by the list of parameters. Such a solution, though certainly not optimal, can be accepted at a first stage.

d) On the basis of the experience gained in implementing a solution such as mentioned under c), the construction of a much more elaborate translator capable of generating object— instructions in an optimal form could be considered.

UTILISATION DE SOUS-PROGRAMMES STANDARD DANS LA PROGRAMMATION DU TRAITEMENT DU RECENSEMENT SUR ORDINATEUR

On présente, dans cet ouvrage, un aspect particulier de grande importance pour la programmation du traitement électronique des travaux statistiques: l'utilisation des sous-routines pour l'écriture des programmes.

On analyse, tout d'abord, les situations concrètes qui ont conduit à l'idée de l'utilisation des sous-routines, on présente le paquet de sousroutines l'une après l'autre, et on mentionne ensuite l'importance de ces sous-routines considérées comme un premier pas vers la réalisation des générateurs de programmes orientés vers les problèmes statistiques.

ПРИМЕНЕНИЕ СТАНДАРТНЫХ ПОДПРОГРАММ ПРИ ПРОГРАММИРОВАНИИ ОБРАБОТКИ ИТОГОВ ПЕРЕПИСИ

В этом труде рассматривается своеобразный аспект, имеющий большое значение для предстоящей деятельности в области программирования электронной обработки статистических данных, а именно — применение подпрограмм при разработке программ.

В первую очередь, проанализированы конкретные обстоятельства, которые привели к мысли использовать подпрограммы; затем, описано значение этих подпрограмм, как первого шага по пути автоматического составления программ, направленных на статистические проблемы.