INSTITUTE OF SONOLOGY - UTRECHT STATE UNIVERSITY

ELECTRONIC MUSIC REPORTS

PROJECT 2

a programme for musical composition

3

E L E C T R O N I C   M U S I C   R E P O R T S

published by the Institute of Sonology at Utrecht State University

No. 3 December 1970

Appears aperiodically once or twice annually.
Single numbers 2.35 guilders (outside Europe $ 1.00),
subscription for five numbers 10.50 guilders ($ 4.60).
If use is not made of an International Post Order the prices will be increased by 1.50 guilders ($0.40).

Gottfried Michael Koenig

P R O J E C T 2

Computer programme for the calculation of musical structure variants

PREFACE

Growing interest in the use of computers for purposes of musical composition has stimulated composers versed in mathematics or matters pertaining to the computer to write programmes for their own use. There ought, apart from this, to be programmes which any composer can use just as he uses manuscript paper, a piano or a tape-recorder, without any knowledge of computers or mathematics. (Experiments in this field have been made with the MUSICOMP programme, written by Robert A. Baker and Lejaren A. Hiller. Max V. Mathews' MUSIC V programme does not belong to this category, being primarily for sound production.)

PROJECT 2 is an attempt to make a composing programme of this sort for general use. Experience gained in the use of PROJECT 1 formed a basis for this (see Electronic Music Reports 2/1970), especially the criticism that the compositional rules for this programme were inflexible. However, the writer of PROJECT 2 is aware of its weaknesses: the manifold combinatorial possibilities of several parameters can never do justice to all the requirements of a composer, not to mention the compositional-theoretical points of departure without which a programme like this could not be written at all but which could be an out-and-out contradiction of the ideas of a composer faced with using the programme.

For these reasons, PROJECT 2 is also intended to be used for research into compositional theory. It is meant to answer the question as to what in music can be programmed, and to what extent rules of serial or aleatoric music, whether already formulated or capable in individual cases of being formulated, are valid. Furthermore, the manner should be observed in which contact between the composer and the computer occurs, or, to put it more precisely: what is the extent to which the composer is ready and able to formulate his structural principles in the framework of the possibilities provided by the programme, to plan the course of a piece of music in advance and to translate this plan into constellations of individual parameter values. For this purpose, copies of all experiments and results will be kept at the institute and in due course subjected to an investigation.

The writer wishes to thank the computer centre at Utrecht University for its support during the exhaustive test runs, Greta Vermeulen, who is responsible for numerous programme details and for revising of the entire programme, and Ruth Koenig for her translation of the manual into English.

CONTENTS

B THE STRUCTURE FORMULA

C INPUT AND OUTPUT

INTRODUCTION

The purpose of PROJECT 2 (PR-2) is to "calculate musical structure variants". By this we mean the forming of musical texts (scores) according to musical data such as instruments, durations, dynamics etc., but also according to given compositional rules (selection and permutation of parameter values, dependency of parameters on one another). The programme is fashioned in such a way that its user (whom we shall call the "composer" from now on) defines the compositional rules and also provides the data. Seen in this light, the computer does not "compose"; it merely carries out the orders which it is given. It performs a mechanical task which the composer would otherwise be forced to perform. Of course, the computer's high calculation speed enables it to do mechanical work which the composer is unable to do because of the time involved. In other words, the computer can help solve problems which previously had to remain unsolved because of the time it took to solve them; for instance, it can combine data to form any number of constellations, thus providing the composer with a clearer survey of the solutions inherent in a problem.

In the following, we shall call compositional rules RULES, musical quantities (durations, dynamics etc.) DATA or ELEMENTS. The characteristics of the musical sound (pitch, timbre, attack etc.) will be called PARAMETERS. The specific combination of rules and elements going to make up a problem will be known as the STRUCTURE FORMULA, and the various combinatorial possibilities resulting from a structure formula will be called VARIANTS. Variants resulting from one single structure formula give us a VARIANT GROUP. To put it in simple terms:

> the composer sets up a structure formula (consisting of rules and elements for several parameters) and has the computer work out one or more variants.

The structure formula is governed by a number of instructions which will be explained in this manual. Among them are:

- the giving of elements and their grouping (section 2),
- the selection and permutation of elements and groups of elements (section 3),
- the available permutation principles (section 4),
- the hierarchy of the parameters and vertical density (section 5),
- attaching groups to a common selection principle, division of the

score into "layers" (section 6),

- in sections 7 and 8 the individual parameters are discussed,
- in section 9 the additional structure data,
- in section 10 the data form and
- in section 11 the layout of all check data and of the composed results.

The appendix contains a survey of all programme entries in the form of a table. In sections 2-6 only the basic principles are discussed. Details as to how the composer must write down his structure formulae are in sections 7-9. A list of illustrations and an alphabetical register complete the manual.

In order to make PR-2 serve as many purposes as possible, all important musical parameters have been taken into account (instrument, harmony, register, entry delay, duration, rest, dynamics and mode of performance). The harmony parameter has been worked out in three alternatives, all of which generate any number of pitches per octave. Clusters and other divergences can be programmed by resorting to the "mode of performance". The parameters mentioned depend on one another in that once a parameter has been composed in a variant, it can not be altered; the following parameters must adapt to those already composed - in the framework of the structure formula set up by the composer. The composer determines the order in which the parameters are inserted in the score.

The present form of the programme is a compromise between great flexibility on the computer's part and clarity for the composer, to whom the input system (data form) appears as the operational panel of a machine. It would doubtlessly be theoretically possible to have more parameters, more rules and especially more opportunities for variation in a variant, but this would affect the programme's "operability"; and those very composers with little experience in planning a structure in advance - who should be encouraged by this programme to gain such experience - would be placed in situations difficult for them to assess. When this programme was being made, the idea was to place a few elements - according to elementary selection and permutation principles - in manifold constellations and to alter these constellations in small stages. This philosophy could make the structure of the programme appear more complicated than necessary, for the composer's data can contradict each other, thus preventing a sensible result. Programme errors can also be made if unsuitable data cause the wrong run instructions for certain indices. This is why it must be stated here that useful results can only be expected if the composer carefully studies the instructions in this manual and acts according to them.

PR-2 is written in ALGOL 60 and consists of three parts with a preliminary programme (see fig. 1-1). In the first part the composer's data are fed in and printed; a number of overall decisions are also made as to the order of the parameters, the duration of the variants, grouping of elements etc. In the second part the score is put together, parameter for parameter. Finally, in the third part the time-values are worked out into metric quantities, and the score and parts are printed in the form of tables. The great length of the programme made it necessary for it to be divided up into three parts in this manner.

The preliminary programme converts the composer's data into code, since the data format for the first part of PR-2 is impractical for the composer, necessitating a great deal of writing and thus providing a greater risk of errors. The output of the preliminary programme then serves as input for the first part of PR-2.

It should also be stated here that - although PR-2 is considered to be complete - a few more alterations are to be expected in the near future. Users of PR-2 will be kept informed of such alterations.

There are also "test programmes" which the composer can use to get acquainted with individual problems of PR-2 and which may be obtained on request (data forms and explanations). These test programmes are called ALEA, SERIES, RATIO, GROUP, TENDENCY, DYNAMICS, HARMONY, TXCL and TIME. For application possibilities of the test programmes see the description of these functions in the manual.

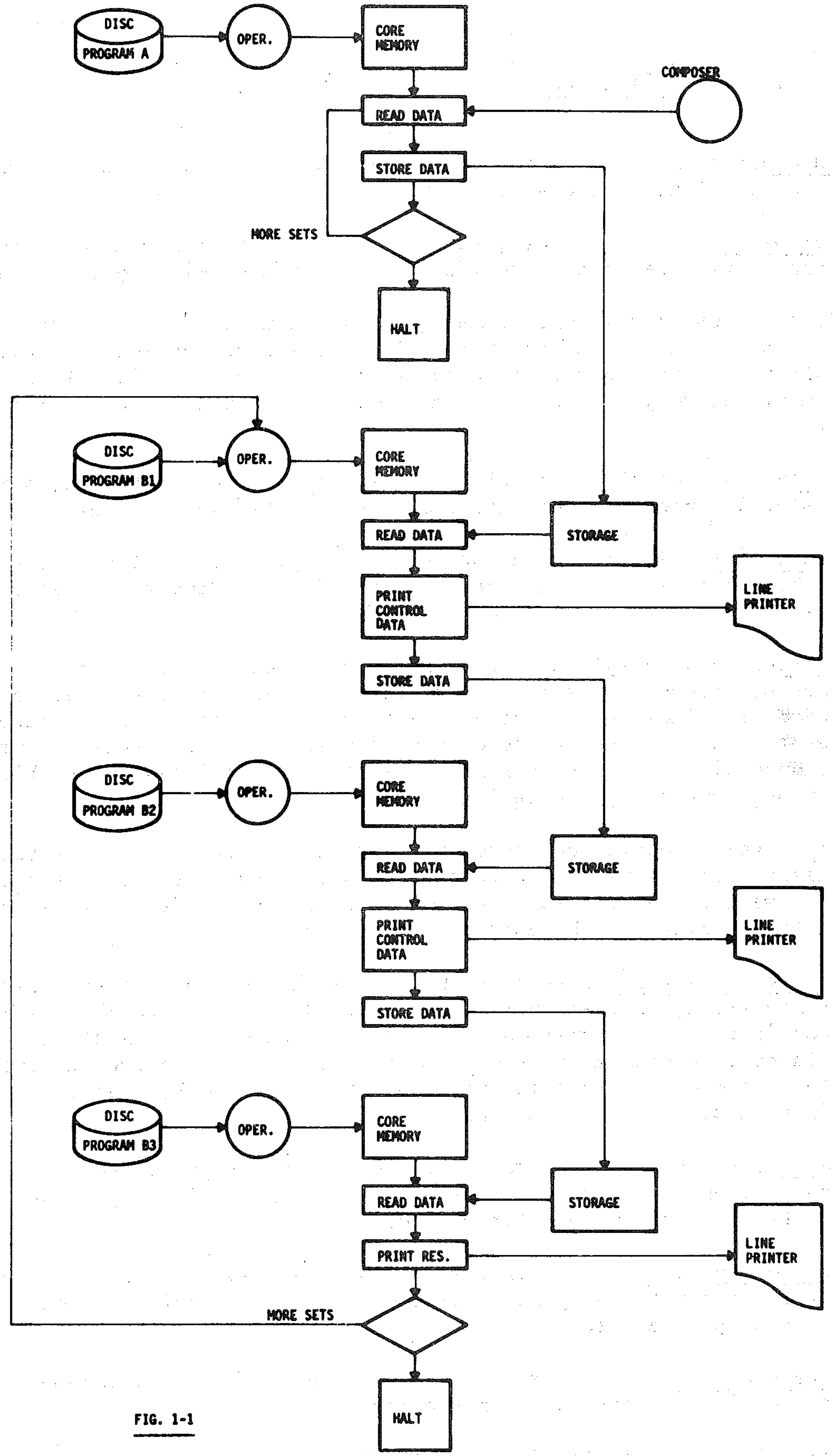

DISC
PROGRAM A
OPER.
CORE MEMORY
COMPOSER
READ DATA
STORE DATA
MORE SETS
HALT
DISC
PROGRAM B1
OPER.
CORE MEMORY
READ DATA
STORAGE
PRINT CONTROL DATA
LINE PRINTER
STORE DATA
DISC
PROGRAM B2
OPER.
CORE MEMORY
READ DATA
STORAGE
PRINT CONTROL DATA
LINE PRINTER
STORE DATA
DISC
PROGRAM B3
OPER.
CORE MEMORY
READ DATA
STORAGE
PRINT RES.
LINE PRINTER
MORE SETS
HALT

FIG. 1-1

1 DEFINITIONS

Absolute pitch. Pitch code for the entire pitch range, consisting of an octave digit (1 to 9) and a →relative pitch.

Autonomous rest. A rest subsequently inserted in the →rhythmic context by means of the rest →parameter; not to be confused with →pseudo rest.

Chord. In general: all tones commencing simultaneously; each →entry point is occupied by a chord. Specially: tones which commence simultaneously in accordance with a chord instruction (→chord size of an instrument, chord in terms of the harmonic →chord principle, →vertical density).

Chord size. Number of tones which can be played by an instrument simultaneously. Defined for each instrument between two limit values.

Chord principle. One of three harmonic principles. Several chords form a stockpile from which selections can be made according to a →selection principle.

Combination. A principle developed for PR-2 according to which the sequence of group-indices in the instrument parameter is automatically adopted by the indicated →parameter; the →groups designated by these indices are put together to form an →ensemble. The parameters excepted from the combination obey their own →selection principles, with which however only one group can be indicated for the ensemble each time.

Comment. Indication in the score printout that certain parameter values are inserted in contradiction to the composer's rules since suitable elements were not available in the →ensemble. A →wrong element is marked by a letter indicating the appropriate →parameter.

Compatible parameters. Parameters whose elements can be fitted together in any manner. Special attention must be paid to the compatibility in the →combination of parameters. Since the selection of elements for the score is made according to an →ensemble, only the respective ensembles have to be compatible; but if group selection is made by aleatoric means, the →lists must also be compatible.

Concealed rest. →Autonomous rest inserted in the →rhythmic context instead of a →sound entry which is not also a →general entry, superposed however by a sustained tone (or →chord).

Definite entry point. Point in time at which an →autonomous rest commences. The definite entry point is found after a →provisional entry point has been established.

Definite rest delay. The time-lag between the end of an →autonomous rest and the →definite entry point of the next rest.

Ensemble. An array (field of numbers) among whose elements a selection can be made by means of a →selection principle for the assembly of the score. The ensemble consists of one or more →groups taken from the →table in accordance with a selection principle.

Entry point. Point in time at which a tone, →chord or →autonomous rest commences.

Entry range. Time range in which the →provisional entry point of an →autonomous rest is found by aleatoric means.

Format. Read instruction for the manner and number of data required for each →programme entry. Infringements of the format prevent the computer from identifying the data.

General entry. →Entry point of a tone or →chord not superposed by a "sustained" tone or chord.

Group. Elements of a →list which are to be put into an →ensemble together. The composer assembles groups to form a →table.

Hierarchy. The order in which the →parameters are used in the score. If the parameters are interdependent the hierarchy can acquire great significance; it is established by the composer or by aleatoric means.

Infinite row. Sequence of pitches generated according to the →interval principle as a consequence of an interval matrix.

Initial set. The first →set of data, which must contain data for all the →programme entries. Unchanged data are automatically adopted for →subsequent sets.

Instrumentation. The allocation of instruments to →entry points, pitches, durations, intensities or modes of performance. In a stricter sense, instrumentation applies to already given tones in →chords; the rules of →vertical density must be obeyed here.

Interval principle. One of three harmonic principles. An interval matrix, either given by the composer or calculated by PR-2 and containing the intervals allowed or forbidden to follow each given interval, results in an →infinite row, the tones of which are used to occupy the time-points in the score in chronological order.

Layer. Musical structure in which the values of each individual →parameter are taken from a uniform stockpile (→ensemble) in chronological order in accordance with a particular →selection principle. A layer is either based on a single →group in the ensemble or on the →union of several groups. Without union several layers occur, and appear one after the other in the printout of the score. For the printout of the parts, the layers are fitted together in such a way that all time values are in chronological order. Several layers always have the same →structure duration and metronome tempo.

List. A stockpile of data provided by the composer for a particular →parameter. List elements are assembled in the →table to form →groups.

Main parameter. The first →parameter in the →hierarchy. The →selection principle for the main parameter can do its work without interference, but creates conditions which in certain circumstances restrict the other parameters.

Parameter. This term is used in PR-2 to indicate the characteristics of the musical structure that can be applied to individual tones. These are: the name of the instrument which is to play the tone, pitch, register, duration, time-lag before the next tone (entry delay), interpolated rests, intensity and mode of performance (or articulation). Interpolated rests and performance are usually used in forming groups of tones. - The term parameter must not be confused with "actual parameter", which is used to denote a variable accompanying a sub-programme, for example a selection programme.

Pitch grid. Division of the octave into single tones (→relative pitches), tone system.

Programme entry. The possibility of conveying the PR-2 data for a particular purpose. The composer's →structure formula can at present be read via 63 programme entries. The data for a programme entry must be written in a particular →format.

Provisional entry point. For →autonomous rests, a provisional entry point is first of all obtained by aleatoric means within a given →entry range. The →definite entry point is then sought, starting from this provisional entry point.

Pseudo chord. Two or more tones commencing simultaneously because of the entry delay 0.

Pseudo rest. Occurs if the duration of a tone is shorter than its entry delay.

Relative pitch. Pitch index within the interval of an octave, expressed in the form of a two-digit number between 1 and 99. Relative pitch 0 (0-pitch) means a percussion instrument.

Rest duration. The duration of an →autonomous rest, not to be confused with →pseudo rest.

Rhythmic context. The totality of all time-data (→entry points and ends of tones) in chronological order.

Row principle. One of three harmonic principles. Several tones form a stockpile which can be transposed according to various principles.

Selection. The choice of elements in a given stockpile resulting in a new stockpile (or a final result) representing a permutation of the old one.

Selection cycle. Unique and complete application of a →selection principle, with a finite or infinite number of selected elements. Finite selection cycles are repeated if more elements are required; infinite and finite selection cycles are stopped when the required number of results has been obtained.

Selection principle. A rule according to which a →selection is made among the elements of a given stockpile.

Series. Arrangement of the elements of a →parameter according to a particular principle. The composer is free in his arrangement of the elements in →list and →table. →Selection principles are available for the formation of series in the →ensemble and score. The result of a completed selection process is also occasionally known as series, →selection cycle.

Set of data. The totality of all the composer's data to be read for the calculation of a variant group.

Sound entry. →Entry point for a tone or →chord as opposed to the entry point of an →autonomous rest.

Structure duration. The duration in seconds, given by the composer, of a →structure variant. Since the number of →entry points is calculated from the structure's duration and the average entry delay, and the entry delays are determined by one of the →selection principles, the sum of entry delays is not certain to correspond with the structure duration.

Structure formula. The structural characteristics of the →parameters and their elements as expressed in the input data, rules for the →selection of elements and for the →hierarchy of the parameters, another designation for →set of data.

Structure variant. If the number and selection of →groups for an →ensemble on the one hand and/or the selection of ensemble elements for the score on the other hand are determined aleatorically, the result is not the only possible interpretation of a →structure formula, but merely a structure variant. All further variants will have other characteristics inherent to the →set of data whilst having similar element stockpiles and →selection principles, and thus will gradually reveal the ambiguity of the structure formula.

Subsequent set. →Set of data referring to an →initial set or a subsequently read set of data and thus not necessarily complete.

Table. Collective term for →groups. Represents a stockpile of groups among which selections can be made for the →ensemble by means of a →selection principle.

Transposition cycle. →Selection cycle of transposition intervals for the transposition of →chords and rows which the composer has declared in the harmony →parameter for the →chord principle or→row principle.

Union. The treatment of the →ensemble as a single unit to which the →selection principle for the score extends regardless of the number of →groups comprising the ensemble. Only one →layer results from union.

Vertical density. Number of tones commencing simultaneously at each timepoint, regardless of the →chord size of the instruments involved. An automatic check sees to it that the total number of superposed tones per →layer does not exceed the number of tones in the→pitch grid.

Wrong element. An element in the →ensemble which is used in violation of the rules inherent to the →structure formula because no element can be found to satisfy the rules. Wrong elements are marked by a →comment in the score.

A BASIC PRINCIPLES

2 VARIABLE DATA AND THEIR SELECTION

In order to understand PR-2 and its practical applications, it is important to make a clear picture of the kind of data to be used, and of their selection. For this purpose we distinguish elements and their index numbers, lists, tables and ensembles. Elements are either numbers or symbols, the index indicating the position of an element in a list, table or ensemble. The above terms are defined as follows:

ELEMENT Number or symbol

NUMBER Whole or decimal number
Examples: *15*
13.9
0.005

SYMBOL Combination of signs
Examples: *PROJECT 2*
TEST 5 - April 1968
37

SIGN Number or letter not representing a numerical value.
Examples: *the digits from 0 to 9,*
the 26 letters of the alphabet,
other signs such as . , " / + - ( ) and the space

LIST Enumeration of elements, each of which is automatically provided with an index.

TABLE Arrangement of list indices in lines and columns. The lines do not have to be of equal length, and are called "groups". The current number of a group is its group index.

ENSEMBLE Enumeration of several groups which are to be regarded as a unit. A single group can also be declared as an ensemble.

Both a list and a table must be made for almost all parameters. The ensemble then results from the required combination of groups. This construction means that a list-element obtains several indices:

(a) a list index,

(b) a group index in the table (line index),

(c) a column index within the group,

(d) an ensemble index. The ensemble index depends on the one hand on the position of the list index within a group and on the other hand on the order of the groups in the ensemble. See example 2-1.

Example 2-1

LIST

index	1	2	3	4	5
elements:	A	B	C	D	E

TABLE

	1	2	3	4	5	← *column index*
1	1	2	3	4	5	}
2	3	4	5			} *list indices*
3	2	2	2	2		}

↑
group index

Order of groups: 2 1

ENSEMBLE

index:	1	2	3	4	5	6	7	8
list indices:	3	4	5	1	2	3	4	5
corresponding to:	C	D	E	A	B	C	D	E

2.1 LIST

For each parameter, a list is first made containing all "allowed" elements; "allowed" does not mean that they occur in a variant, but depends rather on compositorial conditions: if 20 elements are available for a parameter, but the variant is so short that only 17 are required, 3 elements are left over (excluding element repetitions). A list-element can only occur in a variant if it is named in the table and if the group containing the element is selected for the ensemble.

The kind of list-element depends on the kind of parameter. Time-values (for entry delay, duration and rest) are given in the form of numbers (seconds), designations of instruments, dynamics or modes of performance as symbols.

Examples:

DURATION LIST	*0.1 0.2 0.3 0.5 0.7 1.0 1.4*
DYNAMICS LIST	*ppp pp p mf f ff fff*

2.2 TABLE

Besides the list, a table must also be made for most of the parameters. The purpose of the table is to arrange the list-elements in groups or to make a selection among the elements. The elements themselves are never named in the table, but only their corresponding list indices. An index may be named several times within a group. If no groups are to be formed, it is sufficient to name the list indices once - in the form of a single group. See example 2-2.

Example 2-2

LIST	*0.1*	*0.2*	*0.3*	*0.5*	*0.7*	*1.0*	*1.4*	*2.0*
TABLE	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
	1	*2*	*3*					
	6	*7*	*8*					
	2	*4*	*6*	*8*				
	1	*1*	*1*	*3*	*3*	*5*		

The list consists of 8 elements. The table consists of 5 groups, whose sizes are 8, 3, 3, 4, 6. The first group corresponds to the list; it tells us that - if the first group is selected - the entire list is at our disposal.

The second group only contains the first three elements, the third only the last three. The fourth group is a sort of cross-section. The fifth group contains some repetitions. If the fifth group is selected we have element 1 three times, element 3 twice and element 5 once.

If we just look at the fifth group, the following alternatives are equivalent:

Alternative 1	*LIST:*	*0.1*	*0.2*	*0.3*	*0.5*	*0.7*	*1.0*	*1.4*	*2.0*
	TABLE:	*1*	*1*	*1*	*3*	*3*	*5*		
Alternative 2	*LIST:*	*0.1*	*0.1*	*0.1*	*0.3*	*0.3*	*0.7*		
	TABLE:	*1*	*2*	*3*	*4*	*5*	*6*		

2.3 ENSEMBLE

When a variant is worked out, normally one group per parameter is selected and registered in ENSEMBLE. The ensemble then contains either the entire list or at least a selection represented by the selected group. Under certain circumstances (see section 6) several groups can be put together into an ensemble. The groups are then registered in the ensemble in the same order in which they are named according to the given permutation principle (see example 2-1).

The order of the elements (represented by their list indices) in the ensemble is the final arrangement for the variant. The use of this arrangement for purposes of data selection for the score depends on the permutation principle chosen for this.

Since list-elements can be named more than once in the table (whether in one group or distributed among several), repetitions of elements can also occur in the ensemble. However, PR-2 does not take this into account; the permutation principle for the ensemble merely selects ensemble indices regardless of the elements denoted by these indices.

Repetition of elements in the ensemble can lead to undesired results. Consider the following example:

LIST	*fl*	*ob*	*clar*	*hrn*	*fag*
TABLE	*1*	*2*	*3*	*4*	*5*
	1	*3*	*4*		
	3	*4*	*5*		

If groups 2 and 3 are put together to form the ensemble, we have the arrangement:

ENSEMBLE	*1*	*3*	*4*	*3*	*4*	*5*

According to this we have a flute, 2 clarinets, 2 horns and a bassoon, which could result in unperformable situations for a wind quintet (see list).

The final selection of the parameter elements is basically according to this LIST-TABLE-ENSEMBLE principle; assembly and permutation of the ensemble are governed by permutation principles which the composer determines. This arrangement was chosen

(a) to save the composer having to keep feeding new lists. It is sufficient for all elements participating in the composition to be combined once in long lists. One group per assembly is then formed and quoted for the respective variant;

(b) in order to be able to produce other combinations automatically for various variants according to the same list. One of the permutation principles is then responsible for giving each variant a different number and/or selection of table groups.

3 SELECTION AND PERMUTATION

In accordance with the LIST-TABLE-ENSEMBLE principle the elements (indicated only by their list indices in the table and ensemble) "wander", as it were, from one stage to the next. The arrangement of these three stages could thus be said to work in a selective manner, treating a given supply several times and eventually placing a final selection in a particular order in the structure variant.

In order to clarify this process, we shall indicate the elements only by their indices. A list-element can be indicated by L(i) if L = list and i = the number of its position in the list. Every table-element accordingly is in the form L(i), but the number of its position is TAB(g,s) if g = group index and s = column index. For the assembly of the ensemble only groups which are used in their entirety are selected. An ensemble-element is also in the form L(i), but its place number is ENS(TAB(g,s)), which can be substituted by ENS(j). The index notation is not converted back into elements until the transition to the score is made. Since it has become unimportant as to which position an ensemble-element occupied in the table, we can write down the score element SCORE(k) = ENS(j), and must of course look for the corresponding place in the list for each ENS(j): SCORE(k) = L(ENS(j)).

3.1 CHANCE

The composer determines the elements of a list himself and also puts the list indices in a table; on the other hand, table groups (for the ensemble) and ensemble indices (for the score) can only be selected by means of one of the selection programmes which are a constituent of PR-2. This selection makes use of chance in all cases where the composer does not determine the selection or order himself. The connection is thus made between serial music, in which particular orders and permutations play an important part, and aleatoric music, in which it is merely the presence of particular elements, the relative frequency at which they occur and the manner in which they are scattered that is important. The selection programmes (see section 4) attempt to mediate between these two extremes. In this way PR-2 is able to provide the composer with empiric experience in the application of aleatoric principles.

3.2 REPETITION

The term "repetition" has three aspects in PR-2.

(a) "Premature repetition"

If the elements of a given supply merely have to be put in a different order (once or more than once), no element may be repeated until all the others have been "used up". There is therefore a repetition prohibition applying to the entire supply and which is not lifted until the permutation is completed. If we call the number of selective actions in setting up a permutation V_w, and the number of elements in the supply n, the "duration" of the repetition prohibition is then $V_w = n$. In PR-2 this principle is known as "SERIES".

The repetition prohibition could also be lifted sooner, thus allowing repetitions (though not insisting on them) before all the elements would have occured. The "duration" of the prohibition would then be $V_w < n$. This principle of stopping the repetition check "prematurely" is not employed in PR-2.

If, on the other hand, the repetition check is reduced to one single element (and thus virtually cancelled), the "duration" of the prohibition is $V_w = 1$. This principle is known as "ALEA" in PR-2.

(b) "Scattered repetition"

In this case the elements of the supply may be repeatedly selected before the entire supply has been used up. However, the repetitions do not occur in groups but are scattered over the whole permutation. For this purpose each element is given a ratio factor (see 4.3) which indicates the relative frequency of its recurrence. Scattered repetition is known as "RATIO" in PR-2.

(c) "Repetition in groups"

By this we mean a repetition COMMAND. For each element, an indication is given as to how often it must be repeated (the group size). The elements themselves, and the permitted group sizes too, can be selected with or without repetition check, i.e. according to the SERIES or ALEA principles. The principle of repetition in groups is called "GROUP" in PR-2.

Apart from using these selection principles, the composer can also determine the selection himself by giving explicit instructions as to the order in which the elements (table-groups or ensemble indices) are to be selected. This principle, known in PR-2 as "SEQUENCE" is used in any case by the composer when making the tables.

A last principle can also be mentioned here ("TENDENCY") with which it is possible to have directed selection. Here the repetition check is shifted, as it were, over the ensemble elements so that a different connected group of ensemble elements is available for each act (or acts) of selection. - The selection principles are discussed in detail in section 4.

3.3 SELECTION

Elements (or indices) for lists, tables, the ensemble and the score are either "given" by the composer or selected according to a selection programme. Fig. 3-1 is a diagrammatic survey of the manner in which the final data selection for the score is made. The following observations on selection and permutation (see also 3.4) refer to this diagram.

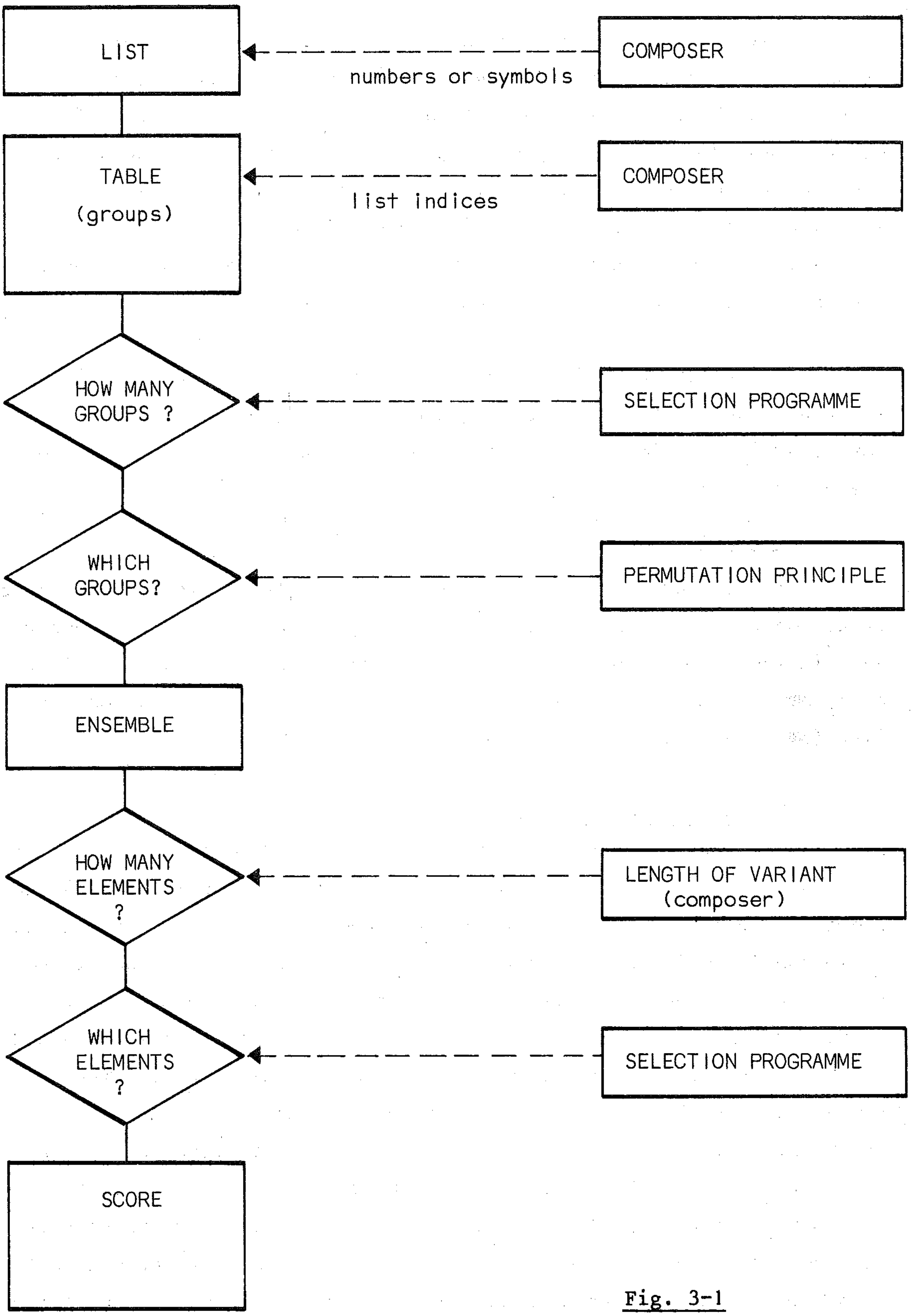

LIST
COMPOSER
numbers or symbols
TABLE
(groups)
COMPOSER
list indices
HOW MANY GROUPS ?
SELECTION PROGRAMME
WHICH GROUPS?
PERMUTATION PRINCIPLE
ENSEMBLE
HOW MANY ELEMENTS ?
LENGTH OF VARIANT (composer)
WHICH ELEMENTS ?
SELECTION PROGRAMME
SCORE

Fig. 3-1

Selection is the choice of elements from a given supply. This supply only contains individual examples, "samples", as it were. The composer can "order" any number of each sample. The sample itself can not be removed from the supply. The rules of the musical composition game decree that occasionally only one example of the sample can be had, sometimes the supply of examples is limited, sometimes unlimited. Put in musical terms: a particular element can only be used once, repeated within limits or selected as often as desired.

The <u>first selection</u> occurs when the composer makes a list of all conceivable data from his imagined supply. Basically each "sample" may occur in the list as often as he wants; but in practice only one example of each sample will be put on the list.

<u>Example</u>:

LIST	*a b c d e*

The computer plays no part in this first selection; the composer writes down the elements, whereupon they are "read" into the computer. An existing list can be replaced by a different one at any time. PR-2 does not have a principle according to which the computer can make a list (based on a calculation, for example).

The <u>second selection</u> occurs when the composer makes a table based on an already existing list. Here each element of the list counts as a "sample", regardless of how often it occurs in the list; repeated elements, just as new ones, are distinguished by their indices. In the table the list-elements are indicated by their indices; it goes without saying that in the table the only indices which may be named are those corresponding to the elements in the most recent list. See example 3-2.

This selection can also only be made by the composer. The table, like the list, may be replaced by any other one at any time. In a table-group, too, each element is basically mentioned only once, although it is possible to have repetitions. The fifth group in fig. 3-2 only makes sense when all the elements in the ensemble are to be selected the same number of times. If, on the other hand, a principle is first to be applied in the ensemble saying that the rate at which the elements occur is in a particular ratio, the fifth group is superfluous; it can be replaced by the second one.

Example 3-2

LIST	*a*	*b*	*c*	*d*	*e*
TABLE	*1*	*2*	*3*	*4*	*5*
	1	*2*			
	3	*4*	*5*		
	1				
	1	*1*	*1*	*2*	*2*

The first group quotes the entire list. Groups 2 and 3 are excerpts which can be put together to make up the entire list (groups 2 and 3 then form an ensemble). The fourth group isolates a single element. The fifth group places two elements in the ratio 2:3 with regard to the rate at which they occur.

Example 3-3

LIST	*a b c d e*
TABLE	*1*
	2
	3
	4
	5

In this case more than one group should form the ensemble. This is however only possible under certain circumstances (see section 6).

The division of the list into groups (groups 2 and 3 in example 3-2) makes it possible for sub-groups to be exchanged. The list can also be permutated element for element if only one of the elements is named for each group (see example 3-3).

A third selection occurs when an ensemble is formed by one or more table-groups. A difference between this and the first two selections is that here whole groups, and not single elements, are selected. Another difference is that it is not the composer who makes this selection, although he does cause it to be made: by setting a selection programme to work.

The selection principles available for this are ALEA, SERIES and SEQUENCE (compare 3.2). If the number or selection are to be left to chance, the random field is automatically confined to the number of table-groups. The random programmes which are applied to the "selection" generate automatically as many numbers as instructed by the "number". But there is no danger in using SEQUENCE for the "selection", because the indices named will be omitted or repeated if necessary.

In example 3-4 some combination possibilities are shown.

3.4 PERMUTATION

By permutation we understand here the (repeated) alteration of a "supply" with regard to the order in which the elements are in the supply. Depending on the permutation principle used, not all the elements have to participate in permutation; some may be repeated during permutation.

At a first glance the difference between selection and permutation appears to be purely theoretical because the elements are selected one after the other, which in itself implies an order; a selection is also a permutation of the selected elements. However, the difference is significant if we permutate a given selection several times or put various selections (with regard to their indices) in the same order. We say that a selection is permutable (it can or should be permutated).

It is perhaps not superfluous to examine this difference between the four stages of the LIST-TABLE-ENSEMBLE principle (see fig. 3-1 again).

<u>Example 3-4</u>

LIST	*a*	*b*	*c*	*d*	*e*
TABLE	*1*	*2*	*3*	*4*	*5*
	1	*2*			
	3	*4*	*5*		
	1				
	1	*1*	*1*	*2*	*2*

(a)

number:	*2*			*(S or R)*	
selection:	*3*	*4*	*2*	*(S)*	
ENSEMBLE:	*3*	*4*	*5*	*1*	

(b)

number:	*3*			*(S or R)*								
selection:	*5*	*2*		*(S)*								
ENSEMBLE:	*1*	*1*	*1*	*2*	*2*	*1*	*2*	*1*	*1*	*1*	*2*	*2*

(c)

number:	*3*			*(S or R)*	
selection:	*3*	*4*	*4*	*(R)*	
ENSEMBLE:	*3*	*4*	*5*	*1*	*1*

(d)

number:	*4*			*(R)*						
selection:	*2*	*3*		*(S)*						
ENSEMBLE:	*1*	*2*	*3*	*4*	*5*	*1*	*2*	*3*	*4*	*5*

ALEA and SERIES are both indicated by R, SEQUENCE by S. All examples refer to the same list and table.

In (a) and (b) the number of selected groups does not agree with the given "number"; this is in any case only possible if the "selection" is made with SEQUENCE. In (a) the extra index is therefore ignored, whereas in (b) the list of group indices is used as often as necessary (a second time in the example). Compare "selection cycle" in the definitions.

Example (c) shows that if ALEA is used, indices can be repeated (this would not have been possible with SERIES). Since the "selection" was made with "R", the number of indices agrees with the "number".

Lastly, in (d) the "selection" is fixed, the "number" (4 in this case) being left to chance. One (or both) of the two halves of the series always arrives in the ensemble, regardless of which number (between 1 and 5) is found according to the "R"-principle.

Example 3-5

	elements	*permutations*	*results*
(a)	*a b c d e*	*1 2 3 4 5* *5 4 3 2 1* *2 4 3 5 1*	*a b c d e* *e d c b a* *b d c e a*
	permutation	*elements*	*results*
(b)	*2 5 3 1 4*	*a b c d e* *e d c b a* *a c e b d*	*b e c a d* *d a c e b* *c d e a b*
(c)	*1 2 3 4 5*	*a b c d e* *f g h i j* *k l m n o*	*a b c d e* *f g h i j* *k l m n o*

In (b) and (c), various lists are read one after the other, but permutated according to the same principle.

If a given selection is to be permutated several times, one of the possible permutations is selected each time. In this sense the selected permutation is in its turn a selection.

The first permutation occurs when the elements of the list are put together to form various groups in the table (the groups can be of varying lengths). The first permutation thus refers to the result of the first selection. Since the composer makes the table himself, he makes at the same time a final selection from the possible permutations of the list unless he replaces the first table by another one later (referring to the same list).

The second permutation occurs when the table-groups are formed into an ensemble. For this, selection programmes are employed; they determine on the one hand the number of groups and on the other hand the selection of the groups. Since the ensemble is formed again for each variant, a permutation can occur for each variant.

The third permutation refers to the respective selection (one of the permutations) from the table-groups: the ensemble contains groups which themselves consist of elements. This sequence of elements in the ensemble can be permutated repeatedly; the permutations are registered in the score. Since the ensemble elements are as a rule frequently brought into play for a musical structure, frequent permutation of the ensemble plays a considerably more important part than the selection that is performed just once.

Selection and permutation overlap. "Selection" is meant to imply that at certain points a selection must be made from imagined or given elements, regardless of possible permutations:

- a list is formed from "possible" elements,
- groups are formed from list-elements,
- an ensemble is put together from groups.

"Permutation" is meant to imply that selected elements (by means of further selections) can be (but do not have to be) put in various orders:

- list-elements can form groups in different orders,
- groups can be combined in various ways to form an ensemble,
- ensemble elements can be registered in the score in various orders.

Here, selection makes a gradual transition to permutation. There would not be much sense in making a given selection of "possible" elements in the form of various permutations into lists: a single list can be permutated into various groups. However, in most cases the composer will form groups of various character, different lengths, so that it is better to speak of repeated selection than of permutation. The third selection is the first to acquire permutational features during the transition from table to ensemble, when variants occur because of various ensembles and the ensembles are based on group-permutations.

The first permutation possibility occurs between list and table, but as we have just seen, it has little significance. The second permutation presents itself when groups are combined, but can only be exploited to a slight extent because of the usually limited number of variants. Furthermore, at the transition to the ensemble, the permutation principle makes the condition that the ensemble elements must always be used in the same order; otherwise it is better to confine permutation to the ensemble. But here the permutation principle comes into its own because - apart from exceptions - the ensemble must often be brought into play for the composition of a variant.

Whilst understanding "permutation" in its strictest sense as the multiple exchange of elements with respect to their order, we can also understand the opposite: the same order (of the indices) of various element-lists; permutation then does not mean the various orders of the same elements, but the same order of various elements. For example:

- an unchanged table can cause the list to be altered in several ways (provided that the lists always have at least the same number of elements),
- a constant selection can cause the table to be altered in several ways for the ensemble (provided that the tables always have at least the same number of groups, at any rate for the SEQUENCE principle),
- a constant permutation for the score can cause the ensemble to be changed again and again (be careful when using SEQUENCE!), either by different combinations of groups, or by new groups, or by new lists (or combinations of these three possibilities).

Example 3-6

	(a)	(b)	(c)	(d)
LIST	x x x y y	x y	x y	x y
TABLE	1 2 3 4 5	1 1 1 2 2	1 2	1 2
ENSEMBLE	1 2 3 4 5	1 1 1 2 2	1 1 1 2 2	1 2
SCORE	x x x y y	x x x y y	x x x y y	x x x y y

In example 3-6 we have a schematic example of the alternatives in the LIST-TABLE-ENSEMBLE principle for passing on elements from the list to the score. The rate of two elements is to be placed in a certain ratio: this can be done in list (a), in table (b), in ensemble (c) or not until the score (d). If only one variant is to be composed, the alternatives are more or less equal. If however several variants are to be worked out, versions (c) or (d) would be preferable, because there are selection programmes for ensemble and score which operate per variant, whilst list or table have to be fed in again for each variant.

Finally it must be mentioned that none of the four stages (list, table, ensemble, score) may be omitted; it is not possible to transport list-elements directly to the ensemble, or table-groups directly to the score. If list-elements are not to be grouped, and several groups are not to be put together to form an ensemble, the list-elements can be copied as a group into the table, and this one group can be taken to the ensemble. The final selection of data for the score always takes place in the ensemble.

3.5 NUMBER OF RESULTS

In permutations with groupwise or scattered repetitions (see 3.2) the result is bigger than the stock of elements. If the repetition prohibition only applies to using the supply once (classic "serial" case), the size of the result is the same as that of the supply. If, however, a selection is made from among the element supply without any check whatsoever, the result can theoretically be infinitely great. It is therefore a good idea to speak about the "number of results" of a permutation (or selection); we shall regard the "result" as complete when the selection principle has been satisfied once, and call it "selection cycle". The "selection cycle" is

- of infinite length without repetition check,
- as long as the supply with "serial" repetition check,
- as long as the sum of all ratio factors with scattered repetitions,
- dependent, in the case of groupwise repetitions
  - (a) on whether there are more elements than group sizes (or vice versa),
  - (b) on the principle according to which elements or group sizes are selected.

Since there is only a finite number of time-points in a structure among which the elements can be distributed according to selection or permutation, it is important to define a reasonable relationship between this number of time-points and the length of the selection cycle. For if the number of time-points is smaller than that of the results, the selected permutation principle does not come completely into its own;

in the reverse case several selection cycles are distributed over the duration of the structure. The "ideal case", which is that the number of results goes exactly into the number of time-points once or more than once, can only be realised in PR-2 in exceptional circumstances. Both quantities are independent of each other (with one exception, see 4.5): the number of results ensues from the composition of the ensemble (which can be subjected to chance) and from the chosen selection programme including the possible ratio factor or group sizes; the number of time-points results from the quotient of the duration of the structure (which can be subjected to chance, see 9.1.3) and the average entry delay (see 9.1.2), which latter is calculated from the ensemble of entry delays, which in its turn can also be the result of random decisions.

4 THE SELECTION PROGRAMMES

For the assembly of an ensemble from table-groups (how many groups? - which groups?) and also for the forming of the score from ensemble data, there are selection programmes which the composer calls by using their call numbers followed (if required) by additional data (the so-called actual parameters). These selection programmes can also be used for other purposes (see sections 5, 7, 8, 9). In this section the selection programmes are described in detail.

4.1 ALEA

Call number: 1

Selection programme ALEA chooses elements of a given supply at random. The elements are regarded as "samples" (see 3.3), and can thus be used repeatedly. There is no check on possible "premature" repetition of elements before the entire supply has been "used up".

The instruction used to select the index of an element in the programme is:

$$i := \text{entier}\ ((z-a+1) \times \text{RANDOM} + a),$$

a and z indicating the random range ($a \leq z$) and RANDOM generating equally distributed numbers between 0 and 1.

If selection programme ALEA is to be called, it is sufficient to indicate call number 1. The programme automatically inserts the actual parameters a and z.

4.2 SERIES

Call number: 2

Selection programme SERIES chooses elements of a given supply at random, the "samples" being removed from the supply. In contrast to ALEA, there is thus a repetition check which sees that no element is repeated until all of them have been selected. As soon as the supply is exhausted, it is automatically regenerated.

For the repetition check the programme has an array r which contains element indices. A counter c indicates how many elements are still available. The instruction for the selection of an index for an element is:

i := r [ALEA (1, c)]

The repetition check is applied by means of the following instructions:

r [i] := r [c];

c := c - 1

If selection programme SERIES is to be called, it is sufficient to indicate call number 2. The programme automatically inserts the actual parameters a and z.

4.3 <u>RATIO</u> (p,p,...,p)

Call number: 3

Selection programme RATIO chooses elements of a given supply at random, each element being given a ratio factor p. The factor p indicates how often the "sample" may be selected before it disappears from the supply. p is an absolute quantity in a single selection cycle, but is a relative quantity with regard to several selection cycles. In any case the relative rate of the elements is defined, their order and repetition both being left to chance. As soon as the supply (and repetitions) is exhausted, it and the ratio factors are regenerated automatically. The random selection applies each time to the total number of elements still available (including repetitions).

If selection programme RATIO is to be called, not only call number 3, but also a ratio factor must be indicated for each list-element. The first ratio factor refers to the first element, the second one to the second element, etc. Ratio factor 0 is allowed; in this case the relevant element is blocked.

RATIO can only be applied to an ensemble which contains one or more groups. Each group is defined by one or more list indices, a ratio factor corresponds to each element index. The respective ratio factor is therefore applied as often as the relevant element occurs in the ensemble. See example 4-1.

Example 4-1

LIST	a b c d e
TABLE	1 2 3 4 5 1 1 4 1 2

(a) *number of groups: 1* *selection of groups: 1* *selection: 3 (3 1 2 0 4)*	*ENSEMBLE* *SCORE (1 cycle)*	*1 2 3 4 5* *b a e e c a e a e c* *(or aleatoric)*
(b) *number of groups: 1* *selection of groups: 2* *selection: 3 (3 1 2 0 4)*	*ENSEMBLE* *SCORE (1 cycle)*	*1 1 4* *a a a a a a*
(c) *number of groups: 2* *selection of groups: 3 2* *selection: 3 (3 1 2 0 4)*	*ENSEMBLE* *SCORE (1 cycle)*	*1 2 1 1 4* *a a b a a a a a a a*

The list contains 5 elements, the table has 3 groups.

(a) If only the first group is selected for the ensemble, each element is available for each selection cycle as often as indicated by the ratio factors in the selection instruction (it is advisable to place the actual parameters in brackets). The fourth element is blocked.

(b) If the second group is selected for the ensemble, the first element arrives in the ensemble twice, and is accordingly available six times.

(c) If two groups are selected for the ensemble and both groups contain the same element, the ratio factor has its turn a corresponding number of times.

The selection cycle in the score would of course be different if another sequence of ratio factors had been chosen.

Example 4-2

(a)	*call:*	*GROUP (1,3,1)*
	result:	*b b | e | b | a a a | c c | e e e | d d*
(b)	*call:*	*GROUP (3,5,2)*
	result:	*d d d a a a a a b b b b | e e e e e a a a a d d d*
(c)	*call:*	*GROUP (2,2,3)*
	result:	*b b c c a a d d e e | d d a a b b c c e e*
(d)	*call:*	*GROUP (1,4,4)*
	result:	*d b b c c c c e e e a a a | c e e b b b b a a a d d d d*

The four examples refer to supply a b c d e. Selection cycles are separated by a vertical line. For the score, as many selection cycles are produced as required for the duration of the structure.

Where SERIES can not be called, but RATIO can, it is possible to substitute SERIES for RATIO; the ratio factor 1 is then given with each list element.

4.4 GROUP (a,z,type)
Call number: 4

Selection programme GROUP produces repetitions in groups. Two decisions are necessary for this: (a) choice of an element, (b) choice of a group size. Both decisions can be made by ALEA as well as by SERIES. The number of elements available for selection results from the respective supply, and is automatically inserted by the programme as an actual parameter. The "allowed" group sizes are always between a and z ($a \leq z$). The selection mode is defined by the parameter "type"; the four possible combinations of ALEA and SERIES for element and group size can be seen in the table below:

call number for type	element selected by	group size selected by
1	ALEA	ALEA
2	ALEA	SERIES
3	SERIES	ALEA
4	SERIES	SERIES

If selection programme GROUP is to be called, not only call number 4, but also the limit values a and z must be indicated for the group sizes and also a call number for the type.

The length of the selection cycle depends on the type:
type 1: one element (repeated if necessary),
type 2: each group size once,
type 3: each element once (repeated if necessary),
type 4: each group size once if $n < m$,
each element once if $n \geq m$,
n = number of elements,
m = number of group sizes ($m = z-a+1$).

See example 4-2.

4.5 SEQUENCE (i,i,...,i)
Call number: 5

The composer uses selection programme SEQUENCE (abbreviated SEQ) to define the order of the elements himself. The indices of the respective supply (table-groups, list elements, ensemble elements) are given as actual parameters. There can be any number of the indices if SEQ is applied to an ensemble; this number may not exceed that of the table-groups if SEQ is used to form an ensemble. The indices may not exceed the number of elements in the respective supply (list, table, ensemble). The selection cycle produced by SEQ is identical with the elements named by the composer.

Selection programme SEQUENCE can only be used where the number of available elements is known with certainty. This is not always the case. We can distinguish three cases schematically:

(a) (1) SEQ is applied directly to a list (tempi, structure durations);
(2) SEQ determines the number of groups which are to form the ensemble.

Since the "supply" is given by the composer in both cases, the number of elements (and thus the highest index) is always known.

SEQ can be used without any difficulty.

(b) SEQ is applied to an ensemble which has been formed by aleatoric means (with regard to number and selection of the groups). The total number of elements in the ensemble is therefore unknown.

SEQ can not be used.

(c) SEQ is applied to an ensemble, the formation of which is known. Examples:
(1) All table-groups are of equal length, the same number of groups is always selected (again with SEQ), the selection can be aleatoric. The total number of elements in the ensemble is known.
(2) The table-groups are of varying lengths; SEQ always selects (aleatorically) only one group and does not specify more elements than are contained in the shortest group. Here the total number of ensemble elements is unknown.

SEQ can be used with caution.

4.6 TENDENCY (d,a1,a2,z1,z2,...,d,a1,a2,z1,z2)

Call number: 6

Selection programme TENDENCY (abbreviated TEND) provides the possibility of defining a particular part in the ensemble and of making selections in this part with ALEA. For this purpose the entire ensemble is covered by a "mask" which only reveals the required part. The limits of the part are called the "edges" of the mask. Since the number of ensemble elements if often unknown, the mask's edges are expressed in percentages.

In a structure variant the mask can assume any number of positions, in which case we speak of "sub-tendencies". The number of positions refers to the total number of entry points or tones in the variant, and is also expressed in percentages. (See 9.1 for the number of entry points.) The percentages for the mask's edges and the number of positions accompany the call for TENDENCY as actual parameters.

The mask can not only take up fixed positions, but can also move at regular speed during the period of time available for a given position (defined as the number of entry points). The two edges of the mask can move independently: towards each other, away from each other, parallel, crosswise.

The actual parameters are defined as follows:

- d "duration" of the sub-tendency, expressed in percentages of the number of entry points or tones,
- a1 left (lower) edge of mask at the beginning of a sub-tendency,
- a2 right (upper) edge of mask at the beginning of a sub-tendency,
- z1 left (lower) edge of mask at the end of a sub-tendency,
- z2 right (upper) edge of mask at the end of a sub-tendency.

Each sub-tendency is defined by these five parameters; the sum of all d's must be 100 (=100%). Fig. 4-3 shows how TENDENCY works.

Since all mask positions with regard to both axes are given in percentages, the call (see fig. 4-3) could be applied to ensembles of any size and variants of any length. However, it sometimes happens that the contents of the ensemble are known precisely, and that the ensemble

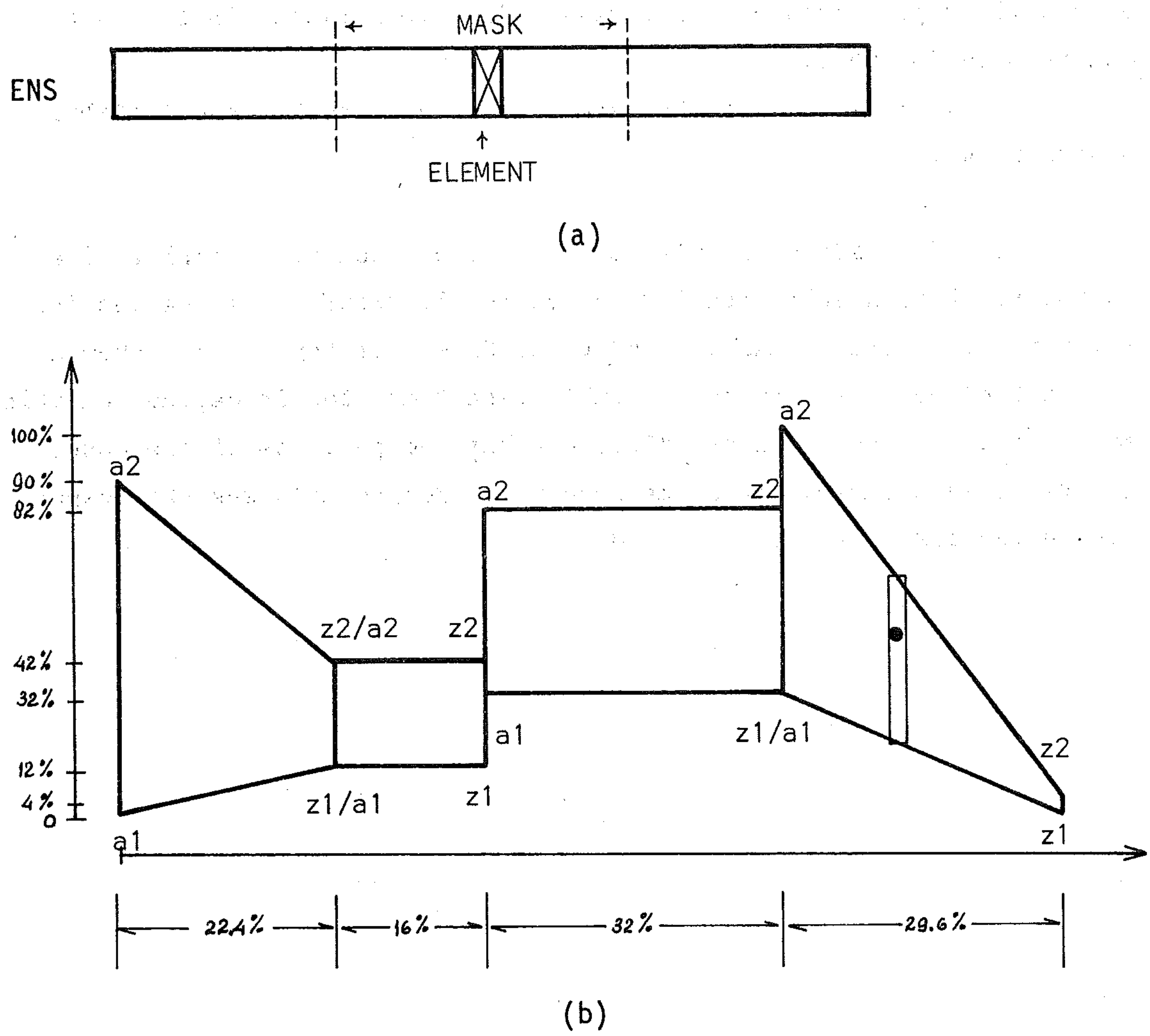


Fig. 4-3

Fig. 4-3(a) shows the ensemble, the momentary position of the mask and an element which is indicated inside the mask by ALEA.

Fig. 4-3(b) shows four sub-tendencies, the momentary position of the mask in the fourth sub-tendency and an aleatorically determined element. The first sub-tendency begins with a mask whose edges leave the area between 0% and 90% of the ensemble visible; the mask narrows down to 12% to 42%.

For the second and third sub-tendency the mask occupies fixed positions between 12% and 42% and 32% and 82% of the ensemble respectively.

In the fourth sub-tendency the mask starts at 32% to 100% to narrow down to the lower 4%.

The complete call for this quadruple tendency would be:

6 (22.4, 0, 90, 12, 42, 16, 12, 42, 12, 42,
32, 32, 82, 32, 82, 29.6, 32, 100, 0, 4).

indices are to be converted into percentages. In this manner the composer can make the mask reveal the required elements and the indices be exactly represented by percentages. The two formulae in fig. 4-4 can be used for this.

Fig. 4-5 will help the reader to understand how the TENDENCY programme works.

The TENDENCY principle can only be applied to ensembles, i.e. to the final data selection for the score. TENDENCY can be called for all parameters except HARMONY and REST. Care must however be taken as to which elements are in the ensemble each time; for it depends on this and additional instructions (MODE) whether the position of the mask alters at each time-point or at each tone. For details see the respective parameters (sections 7 and 8).

$$a = 100 \times \frac{i - 1}{n}$$

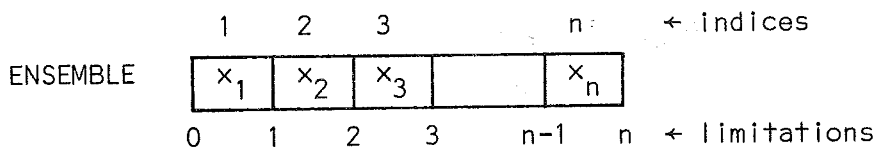
$$z = 100 \times \frac{j}{n}$$

a = lower limit of mask } in %
b = upper limit of mask }

i = index of first element
j = index of last element

n = size of ensemble (number of elements in ensemble)

<u>fig. 4-4</u>



<u>fig. 4-5</u>

The programme first converts the percentages of the mask edges into limitations, and also the percentages of the durations of the sub-tendencies into "number of results". This number of results refers to the total number of entry points or tones in the given variant. The mask's edges are calculated for

start position: lower limit: $A1 = \frac{a1 \times n}{100}$

upper limit: $A2 = \frac{a2 \times n}{100}$

end position: lower limit: $Z1 = \frac{z1 \times n}{100}$

upper limit: $Z2 = \frac{z2 \times n}{100}$

The "number of results" is calculated for a sub-tendency as

$$D = \frac{d \times N}{100}$$

d = duration of sub-tendency (in %)
N = number of time-points in variant

During the sub-tendency the mask must be shifted as often as time-points must be occupied. The shift is calculated with increments

$$shug = \frac{Z1 - A1}{D - 1}$$ for the lower limit,

$$shog = \frac{Z2 - A2}{D - 1}$$ for the upper limit.

The calculated values for the respective mask positions are rounded off to integers in the usual manner and interpreted as limitations. Limitation and index are identical for the upper limits, and the limitation is increased by 1 for the lower limit. If the limitation of the upper limit should become 0, it is replaced by index 1.

For the ratio of the lower limitation (ug) to the upper limiation (og) there are three possibilities:

(a) $ug < og$ (the most frequent case): the elements from which ALEA selects are: x_{ug+1}, x_{ug+2}, ..., x_{og}.

(b) $ug > og$ (occurs if the mask's edges cross or if the actual parameters are given carelessly): the values for ug and og are exchanged, so that we have $ug < og$ again.

(c) $ug = og$ (occurs if the sub-tendency curve results in acute angles): ug and og indicate the same element ($x_{ug+1} = x_{og}$).

5 HORIZONTAL AND VERTICAL ARRANGEMENT

Music can be regarded as successions of sound events. In instrumental music the sound events are produced by musical instruments played by musicians. Every musical sound can therefore be broken up into single components corresponding to the instruments which produce the components. In polyphonic music this results in the problem of registering the sound events in both the vertical context (chords) and the horizontal context (melody). PR-2 is not able to organize melody tones harmonically or chord tones melodically. It is rather the composer's business to use the means available in PR-2 to work out a harmonic structure which does justice to both criteria.

The means in PR-2 can be applied to the horizontal and vertical arrangement as follows:

(a) The musical tone is broken up into several parameters; each parameter is divided into steps which succeed one another.

(b) The time axis of the music is broken up into sections of equal or different size (rhythm). Only one step can be realized per section and per parameter. With respect to the possible change from one step to the next, the parameters are, as it were, "synchronized".

(c) A one-part melody results from (a) and (b), being articulated in various parameters such as pitch, intensity, timbre, etc. The number of tones starting simultaneously can moreover be determined for each time-section: the one-part melody is then replaced by a chord sequence. Each individual tone in the chord is in its turn defined in parameters. - To keep things simple, we can basically speak of "chord", and call single tones "chord size one".

(d) We shall call the limits of the time-sections (compare b) entry points. The time-sections themselves are called entry delays. If the composer makes one or more entry delays in succession zero, the result is still chords, because several entry points occur simultaneously. The difference between melody and chord is thus reduced to a scale of values of entry delays.

(e) Instead of dividing the time axis into sections (entry delays) just once, we can also demand several different kinds of divisions. The result is then "parts" which in their turn can be scored either for one part or in chords. Such parts are called LAYERS (see section 6).

(f) Since the parameters involved in a tone are defined separately, we can regard the tones as a "chord" of the various parameters. However, since many possible parameter combinations must be rejected for practical reasons (the mechanical construction of the musical instrument, performance technique, aural properties), mutual dependencies result. It seems reasonable to establish the parameter which is most important each time (i.e. the one which dominates the structure) independently of the others; these other parameters must then be more or less subordinate according to the unavoidable restrictions. The order in which the parameters are established is therefore of great formal significance.

(g) The order in which the parameters of a structure variant are worked out is called the HIERARCHY of the parameters. We call the (changing) number of tones in the chord the VERTICAL DENSITY of the structure variant.

(h) The restrictions regarding the possibility of combining the parameters into a tone can be different from one instrument to the next. A special combination principle is used to make sure that certain values of the one parameter are only linked with certain values of one or several other parameters (see "group combination" in section 6).

All decisions made by the computer in calculating a musical structure are made one after the other. Mutual dependencies can only occur when later decisions depend on previous ones. This is the case, for example, in selection principle SERIES, where a selected element is blocked until all elements of the supply have been used up.

The same principle of sequential decision also applies to the various parameters of the musical structure. The structure is divided up into the number of layers specified by the composer, each layer being divided up into a particular number of entry points; the parameter programmes are then called in the order established as "hierarchy" by the composer; each individual parameter programme sees to it that the time-points in a layer are occupied by the elements of the respective parameter. Hierarchy and vertical density are discussed in this section, group formation and layers (in terms of points h and e) in the next one.

5.1 HIERARCHY OF THE PARAMETERS

The order in which the parameters are placed in the previously calculated time grid is determined by the composer, partly or completely. If the order is only partly fixed, the remaining parameters are automatically filled in according to selection principle SERIES. Restrictions applying to the hierarchy are discussed in section 9.

Since PR-2 is intended for the composition of instrumental structures, the hierarchy of the parameters is concentrated around the instrument parameter (see fig. 5-1).

It can be clearly seen that nearly all the parameters depend on the instrument and vice versa; the relationships are two-sided. As well as this there are also two pairs of parameters dependent on each other (pitch/register and entry delay/duration), and two one-sided relationships (entry delay-duration/rest and rest/performance). The dependency of the parameters on the instrument is based on the "definition" of the instruments with respect to

- the pitch compass,
- the "allowed" durations, intensities, modes of performance and chord sizes.

Pitches and registers also depend on each other if the instrument is not yet given, because pitches must fit in an already given register, or registers must encompass already given pitches.

Entry delays and durations depend on each other insofar as unison instruments can not play a new tone until the previous one has ceased. In such cases the duration may not be longer than the entry delay. This dependency can however be nullified if superpositions of several tones are tolerated or desired.

Basically, rests are inserted independently of all other parameters. Since however a sustained tone should not be chopped up by subsequently inserted rests, a dependency of the rest on the relationship between entry delay and duration results. This is indicated by the letter P in the lines for entry delay and duration in fig. 5-1.

given parameter	dependent parameters							
	I	H	R	E	D	P	A	M
instrument		H	R		D		A	M
pitch	I		R					
register	I	H						
entry delay					D	P		
duration	I			E		P		
rest								M
intensity	I							
performance	I							

I = instrument
H = pitch
R = register
E = entry delay
D = duration
P = rest
A = intensity
M = performance

Fig. 5-1

Lastly, performance depends on the rest if new modes of performance may only be inserted after rests.

In the left-hand column of fig. 5-1 we have the "given" parameters; "given" means that the respective parameter was calculated before the "dependent" parameters. In the sequence of parameters A B C D E F G H , A would be given for parameters B to H, B for C to H, etc. However, it is frequently the case that not all parameters depend on each other, so that only pairs of parameters are considered, such as C and F; C would then be given, F dependent.

To aid comprehension of the résumé below it must be stated that percussion instruments are indicated by the pitch 0 and the register 0,0. For given percussion instruments a 0 pitch is then generated in HARMONY and a "range" selected in REGISTER (see 8.1). If however pitch and/or register are already given, percussion instruments can only be inserted where a 0 pitch or a 0,0 register have been selected. - Explanations of the "given" parameters follow here:

<u>Instrument given</u>: Pitches can only be inserted if the instrument is a melody one, otherwise pitch 0 is inserted. Registers, durations, intensities and modes of performance are tested as to whether the instrument has been defined for the respective values of these parameters.

<u>Pitch given</u>: Only melody instruments can be inserted. Registers are tested as to whether they encompass the respective pitch.

<u>Pitch "zero" given</u>: Only percussion instruments can be inserted. The register is given the form 0,0.

<u>Register given</u>: Only melody instruments can be inserted. Pitches are tested as to whether they are included in the respective register.

<u>Zero register given</u>: Only percussion instruments can be inserted. The pitch is given the form 0.

<u>Entry delay given</u>: Durations are tested as to whether they are in the allowed ratio to the respective entry delay. If the duration is always to be the same as the entry delay, the duration ensemble is ignored.

<u>Duration given</u>: Instruments are tested as to whether they may play the respective duration. Entry delays are tested as to whether they are in the allowed ratio to the respective duration. The ratio desired by the composer between entry delay and duration applies regardless of the order in which the two parameters are calculated.

Rest is ignored in the hierarchy; rests are inserted in the finished rhythmic context. If rests are to be inserted, this can only be done when the ratio of entry delays to durations (as far as this is relevant for the insertion of rests, see 7.4) does not prevent this (many rests can for instance not be inserted if there are only a few entry points).

Intensity given: Instruments are tested as to whether they may play the respective intensity.

Performance given: Instruments are tested as to whether they may execute the respective mode of performance. Modes of performance can also be required to change at each rest; in this case the rest table (compare 11.3) is consulted for the mode of performance.

This all goes to show that the first parameter in the hierarchy is completely independent; we call it the MAIN PARAMETER. The selection principle assigned to it can freely have its full effect. The second parameter is - if at all - only dependent on one parameter (the main parameter); it is still very probable that its selection principle will have its effect. The third parameter can already depend on two given parameters. If the hierarchy was D-H-A up to this point, the order of the parameters is of no significance because pitch does not depend on duration, nor does intensity depend on either of the other two parameters. If we have R-H-I, pitch depends first of all on the given register (registers can be less than an octave, zero registers are reserved for percussion instruments and exclude pitches >0), the instrument then depends on both pitch (if a pitch is given, all percussion instruments are rejected; if a zero pitch is given, all melody instruments are rejected) and register. The later a parameter has its turn in the given hierarchy, the greater is the probability that the selection principle chosen for it can not have its full effect. Of course, the order of the parameters is entirely irrelevant if all parameters can be combined in any manner; e.g. entry delay and duration independent of each other, register in the form of octaves or larger intervals, no registers in which not all instruments can play, only melody instruments which can play all given durations, intensities and modes of performance, etc. We call these parameters that are unaffected by the hierarchy "compatible" parameters; although the hierarchy formally remains in existence (as the order in which the parameters are to be calculated), it could be altered without any consequences for the compatible parameters.

If this hierarchy would mean that no "allowed" element could be found in some parameter, the ensemble element last tested would be put in the score and provided with a comment reminding the composer that the rules could not be complied with.

5.2 VERTICAL DENSITY

The vertical aspect of a musical structure occurs because of

- the division into layers (see section 6),
- the simultaneous entry of several tones (chords),
- the definition of a tone in the form of several parameters.

By "vertical density" we mean however only the formation of chords; we disregard "pseudo chords" here, which result from single tones with an entry delay of 0. The chord structure is worked out layer for layer; there is no control over the "total density" of several layers.

The composer determines the vertical density by indicating the minimum and maximum number of tones in a chord; selection programmes ALEA, RATIO, GROUP and SEQUENCE can operate between these limits ("autonomous density"). There is also an automatic check in the programme to prevent more tones from occurring simultaneously than indicated for the pitch grid (see 9.3). The composer can avoid this restriction by distributing the context among several layers.

A "chord" is defined on the one hand by the number of tones starting simultaneously and on the other hand by a single entry delay. In other words, if we are operating with chords (compare 8.2), a chord is selected for each entry point. Although the chord-tones commence together, they do not have to be of equal length. If they should be of equal length, a duration is fixed for the chord, otherwise for each tone in the chord.

Vertical density must not be confused with chord sizes, which must be stated for each instrument. If several instruments are involved in a chord, a number of tones is selected for each instrument within the range of its chord sizes; these "sub-chords" are put together until the vertical density is reached. This supplementation of the number of chord tones required for the vertical density can be referred to by a familiar term, the "scoring" of the chord.

Under these conditions we have the following possibilities for the regulation of vertical density in PR-2:

(a) Vertical density can depend on the instrument if INSTRUMENT is main parameter and a specific chord size is determined within the range of the instrument's chord sizes. Pitches can then only be distributed among the chords according to one of the row principles (compare 8.2). Each time-point is occupied by one instrument only.

(b) Vertical density depends on the pitch parameter if the pitches are generated according to the chord principle (compare 8.2); in this case HARMONY is main parameter. The process is divided into two phases:

(1) first, a chord is selected for each time-point according to the chord table (this also provides the number of tones per chord),

(2) then, in the INSTRUMENT parameter (whose position in the hierarchy can be chosen freely, as many instruments are selected per chord as are necessary for the "scoring" of the chord. The number of tones per instrument is established by ALEA between the minimum and maximum chord size. This number of tones is reduced for the last instrument if necessary if the vertical density would otherwise be exceeded.

(c) Vertical density is autonomous if the composer states a minimum and maximum value for the vertical density, which is then determined between these limits by means of a selection programme. In this case only one of the row principles can be used for HARMONY, the main parameter being free. The chords are scored as in (b2).

6 GROUP FORMATION AND LAYERS

We have repeatedly mentioned the LIST-TABLE-ENSEMBLE principle (especially in sections 2 and 3); it means that the composer not only makes a list of elements in a parameter but also a table in which the list elements can be formed into "groups" at will. Several groups can then form an ensemble, which is the final supply for the score. It is frequently not necessary to make use of group combination because individual groups can be arranged in such a way that they already represent the final supply and can be transferred as such into the ensemble. If however several groups form an ensemble they can either be regarded as a uniform group (result: a "layer") or as several single groups (result: as many "layers" as there are groups in the ensemble).

Some rules apply to the combination of groups to form an ensemble; these rules are stated here, and explained in 6.1. The possible "union" of several groups to form a uniform group, and the working-out of several layers, the groups remaining separate, are discussed in 6.2

Rules for combination and union:

(1) Several table-groups can only be assembled to form the ensemble for INSTRUMENT.

(2) Only one group can be indicated for the ensemble for all the other parameters, independent of INSTRUMENT.

(3) If several groups are to form ensembles in the other parameters as well, these may only be groups with the same indices as those also declared as ensemble in INSTRUMENT ("combination").

(4) If the ensembles each only contain one group (with the exception of INSTRUMENT, which may always contain more than one group), the respective variant is executed as one layer.

(5) If the ensembles contain several groups (number and indices of the groups as in INSTRUMENT), they can still be treated as one group; result: one layer ("union").

(6) The groups in the ensembles can also be dealt with separately as in (5); result: as many layers as there are groups in the INSTRUMENT ensemble (and thus also in the ensembles of the "combined" parameters, "no union").

6.1 COMBINATION

As is the case with the hierarchy, the idea of combination and union also comes from instrumental music: manually operated musical instruments which have to execute the structure occupy a central position and are defined with the greatest care. It is often usual in instrumental music to divide the "orchestra" into varying orchestral groups (smaller ensembles); in the original version of PR-2 the formation of an ensemble was only intended for the INSTRUMENT parameter. However, the necessity of providing each instrumental group with its own range of durations or intensities resulted in "combination", so that together with a particular instrumental group the appropriate group of another parameter could also be called. In this aspect combination is thus another type of selection principle, in addition to those already available for group selection.

The "compatibility" of the parameters (see 5.1) is yet another aspect. Without the possibility of combination, two or more parameters would either be compatible (any element of the one parameter can be combined with any element of the other) or incompatible (parameters interdependent in such a way that a preceding parameter restricts the following ones in the selectability of their elements). Combination can be used to turn parameters which are incompatible with regard to their element lists into compatible ones by means of appropriate grouping of elements in the tables; as soon as the combination is de-activated, the previously combined parameters revert to their incompatible state. However, we must state that a certain degree of incompatibility is still present in combined parameters (unless the lists are compatible), because every group-element of one parameter is not necessarily combinable with every element of the relevant group of the other parameter. A resource would be to declare for each instrument its own group and to make the corresponding groups of the parameters that are to be combined only contain elements which are compatible with the respective instrument.

Two examples can illustrate this.

In fig. 6-1 we see the tables of the INSTRUMENT parameter (I) and the tables of any other two parameters (X and Y). Each table contains 3 groups of any length. In order to form the instrument ensemble a selection principle is called which in its turn depends on a decision regarding the number of groups. The number of groups can only be stated for the INSTRUMENT parameter; for all other parameters the number of groups is "one" apart from parameters X and Y, which because of their combination with the INSTRUMENT parameter have the same number of groups. The selection principle for the choice of groups can also be freely selected for each parameter, again with the exception of parameter X and Y, which because of the combination are also coupled with the INSTRUMENT parameter with regard to the group selection. The number and selection of groups are accordingly determined for the INSTRUMENT parameter; the decision arrived at is adopted for the combined parameters X and Y. The selected groups are transferred to the appropriate ensembles. The selection principles for the score are however outside the range of the combination: each parameter (also X and Y) can be called by its own selection principle. Note however that the chosen selection principle applies to the entire ensemble; if several groups are to be in the ensemble they are mixed arbitrarily by selection principles ALEA and GROUP; RATIO could however under certain circumstances block the elements of a particular group. Only if both the number and the selection of the groups were determined by SEQUENCE could the assembly of the score be controlled by SEQUENCE.

The construction in fig. 6-1 is however primarily intended for aleatoric decisions for number and selection of groups.

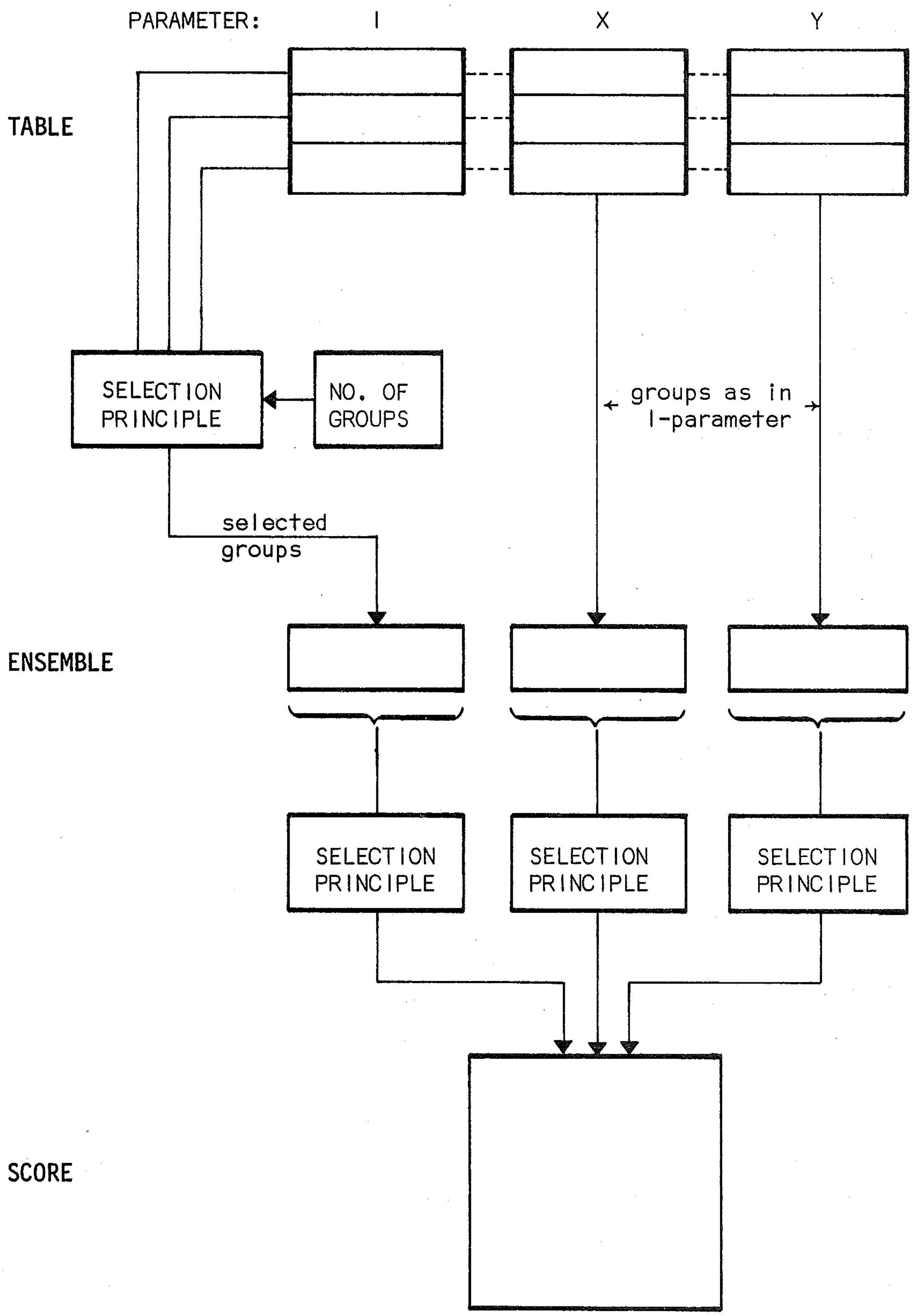

PARAMETER:
I
X
Y
TABLE
SELECTION PRINCIPLE
NO. OF GROUPS
groups as in I-parameter
selected groups
ENSEMBLE
SELECTION PRINCIPLE
SELECTION PRINCIPLE
SELECTION PRINCIPLE
SCORE

Fig. 6-1

Fig. 6-2 shows a construction for the separation of groups in the ensembles. However, for the final selection of elements for the score, TENDENCY, and not any other selection principle, is used. Since all masks assume the same position for the combined parameters (or move between the same limits), only compatible groups are brought into the score when several groups (two in fig. 6-2) are assembled to form the ensemble. It is however a condition that the groups in a table are of equal length, because otherwise the arrangement of the groups in the ensemble with the same mask position would not be free (if we want to leave group selection to chance). The group-length can change from one table to the next because, as we know, the mask is defined in percentages of the size of the ensemble. The masks can of course move freely in the groups which they separate, or assume any number of fixed positions. The movements of the masks can even be in different directions in the parameters. The only condition is that the masks jump simultaneously from one group to the next, and that they respect the limits of the groups (see fig. 6-3).

6.2 UNION

"Union" (or "no union") indicates the part of the ensemble to which the selection principle for the score applies. "Union" is the normal situation here: the ensemble is regarded as a unit which produces a "uniform" score (one "layer"). It does not matter how many table-groups form the ensemble. It should not be forgotten that any number of groups can be used in the INSTRUMENT parameter, but only one group for all other parameters. With "combination" this principle can be departed from: the combined parameters are given the same number of groups and groups with the same indices as in the INSTRUMENT parameter. Examples 6-2 and 6-3 merely showed auxiliary constructions for group separation with simultaneous "union".

If the union is de-activated, the groups in the ensemble stay separate (only those in the INSTRUMENT parameter "without combination", those of all combined parameters "with combination"). As many "variants" are produced as the number of groups in the INSTRUMENT parameter. We only call these variants "layers" because they are calculated for the same structure duration and tempo. The layers are also put together to

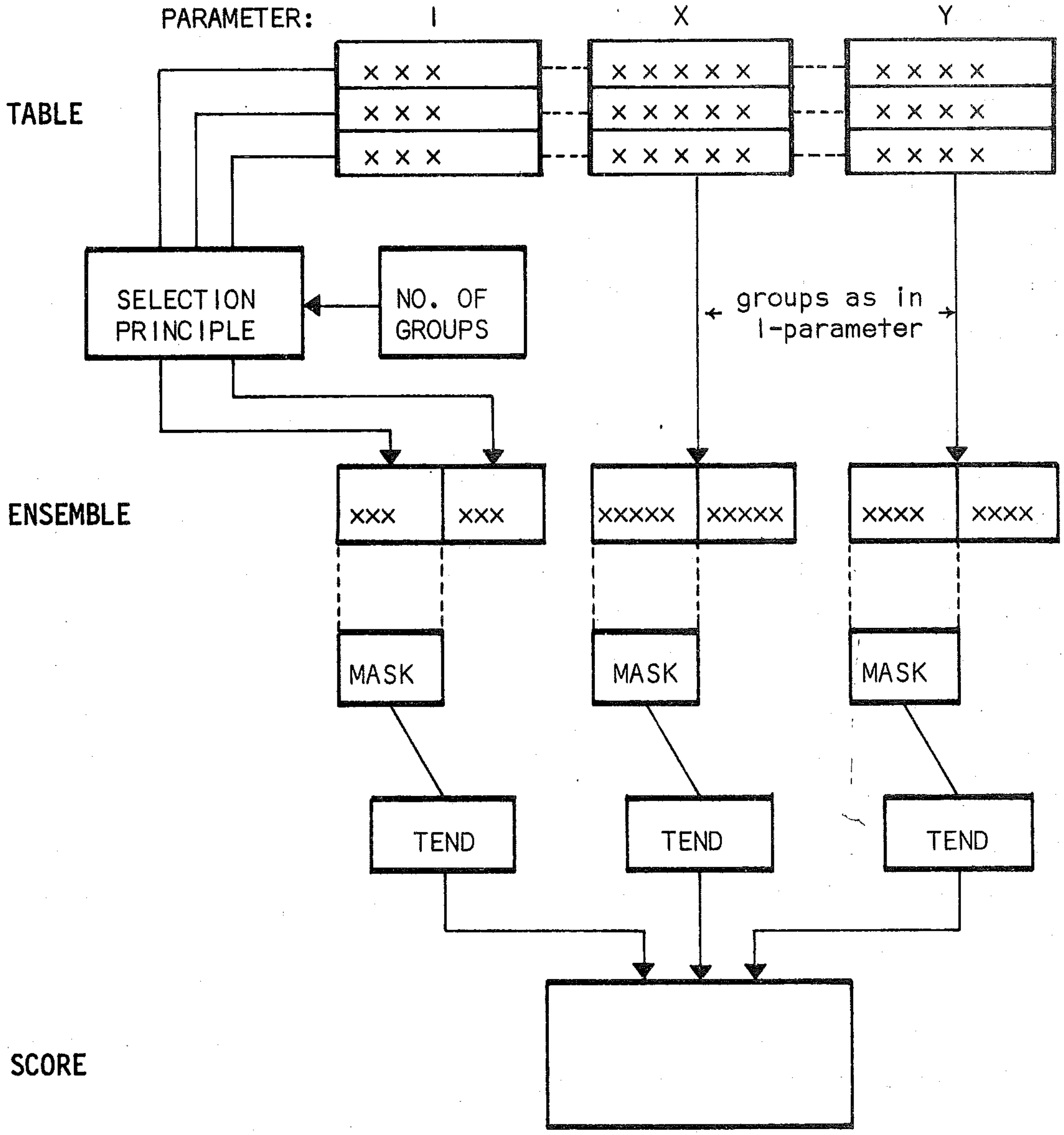

PARAMETER:
I
X
Y
TABLE
x x x
x x x x x
x x x x
SELECTION PRINCIPLE
NO. OF GROUPS
groups as in I-parameter
ENSEMBLE
xxx
xxxxx
xxxx
MASK
TEND
SCORE

Fig. 6-2

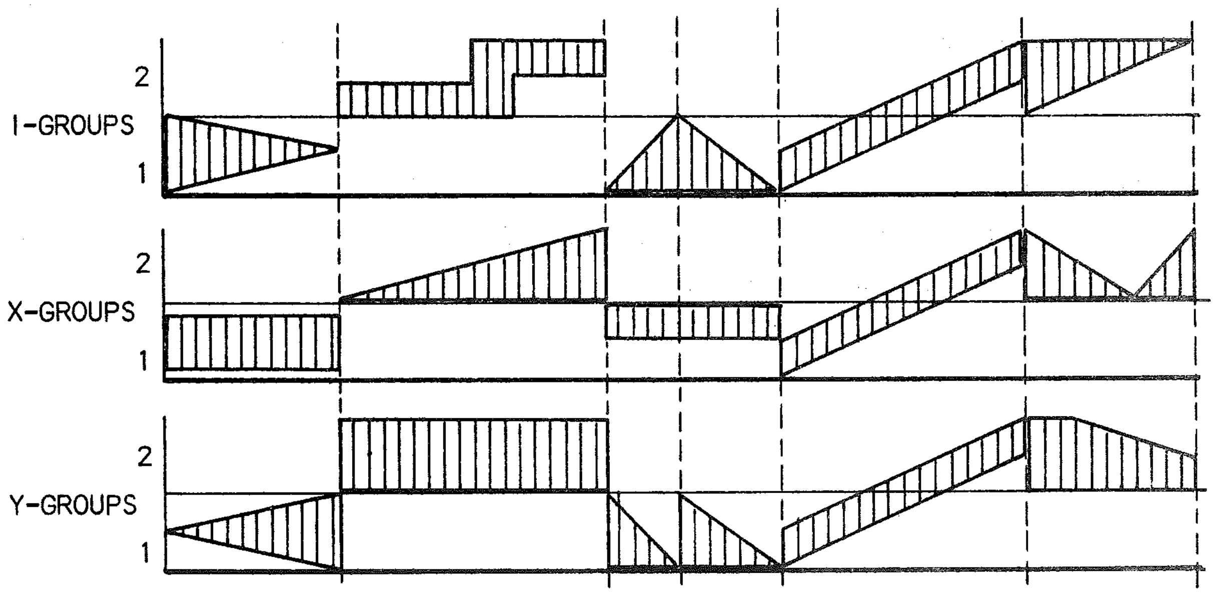

2
I-GROUPS
1
X-GROUPS
Y-GROUPS

Fig. 6-3

form a uniform score, so that all time values (beginnings and ends of tones) are placed in chronological order. Attention must however be paid to the fact that the selection principle chosen for the variant applies to all the layers; it is not possible to have a different selection principle for each layer. Since however a new group is used for each layer, the table-groups can be organized correspondingly.

Fig. 6-4 is a schematic representation of union. Two groups are indicated for the INSTRUMENT parameter (I) and two combined parameters (X and Y). The selection principles for the score first produce the first layer according to the first group and then the second layer according to the second group. The two layers are fitted together before they appear in the output. The composer is therefore able to manipulate the hierarchy in a way not illustrated in fig. 6-4. HARMONY can be calculated in layers (in which case the pitch parameter occupies any position of priority), or it can be calculated after the layers have been fitted together to form the final rhythmic context. In this case the pitch parameter is not called until the very end (even after the rest parameter).

If the fact that it is impossible to call a different selection principle for each layer proves to be a great obstacle, an auxiliary construction can be used. We calculate as many variant groups (only one variant per variant group) as the number of layers to be contained in the score. Admittedly, it is then impossible to work out a common harmony for all the layers, but on the other hand a different selection principle can then be used for each layer (and each parameter).

Example 6-5 uses numbers to illustrate the processes which occur when we have combination and union in groups, ensembles and layers. We distinguish three parameters: I (= instrument), K (any "combined" parameter) and L (a non-combined parameter).

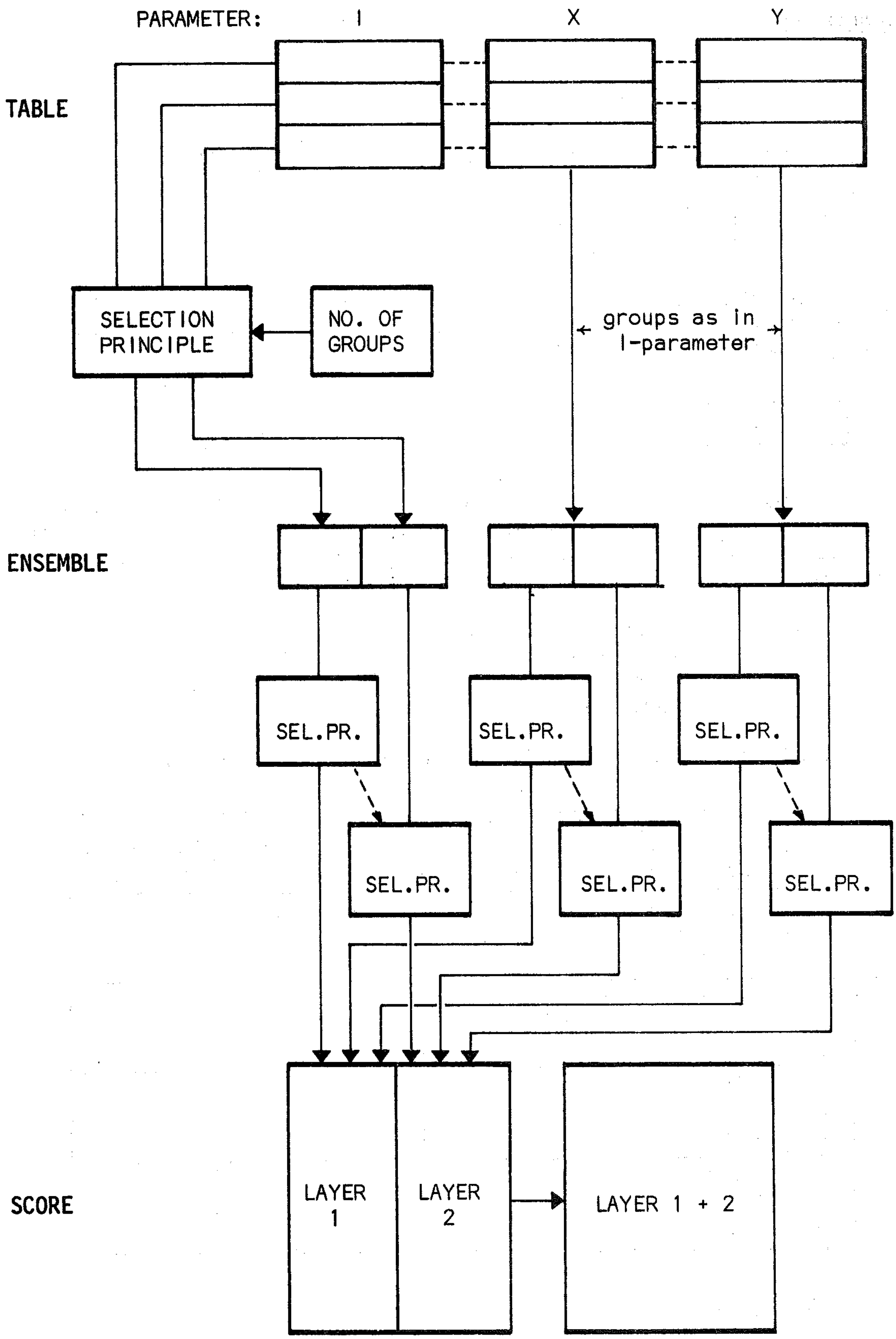

PARAMETER: I X Y
TABLE
SELECTION PRINCIPLE
NO. OF GROUPS
groups as in I-parameter
ENSEMBLE
SEL.PR.
SEL.PR.
SEL.PR.
SEL.PR.
SEL.PR.
SEL.PR.
SCORE
LAYER 1
LAYER 2
LAYER 1 + 2

Fig. 6-4

Example 6-5

Given three tables:

I:

1	2	3
4	5	6
7	8	9

K:

1		
2	3	
4	5	6

L:

1	2	3
4	5	
6		

Assuming the number of groups for the I-parameter to be "two" and the selection principle to have chosen the following groups for the three parameters, we have:

I: | 3 | 1 | 2 | 3 | K: | 2 | 1 | 3 | 2 | L: | 1 | 3 | 1 | 2 |

These group indices are of course only selected as required; but 4 indices have been stated each time in order to demonstrate in the following how the indices are used. - We distinguish the following four cases:

(a) combination / union
(b) combination / no union
(c) no combination / union
(d) no combination / no union

and would have the following list-elements for two variants:

		1st variant		*2nd variant*	
		1st layer	*2nd layer*	*1st layer*	*2nd layer*
(a)	*I*	7 8 9 1 2 3		4 5 6 7 8 9	
	K	4 5 6 1		2 3 4 5 6	
	L	1 2 3		6	
(b)	*I*	7 8 9	1 2 3	4 5 6	7 8 9
	K	4 5 6	1	2 3	4 5 6
	L	1 2 3	6	1 2 3	4 5
(c)	*I*	7 8 9 1 2 3		4 5 6 7 8 9	
	K	2 3		1	
	L	1 2 3		6	
(d)	*I*	7 8 9	1 2 3	4 5 6	7 8 9
	K	2 3	1	4 5 6	2 3
	L	1 2 3	6	1 2 3	4 5

Commentary

(a) Because of the union, only one layer occurs. In the combined parameter the same group indices are selected as in the I-parameter. L only has one group. For the second variant the following group indices are taken into account each time for all parameters.

(b) Two layers result, since no union is required; each layer has one group. In parameter K the same group indices are selected as in parameter I. L has one group for each variant.

(c) Only one layer occurs because of the union. Since no combination is required, parameter K is treated as parameter L: one group per variant.

(d) Two layers result because no union is required; each layer has one group. Since no combination is required, parameter K is treated as parameter L: one group per variant.

B THE STRUCTURE FORMULA

For the parameters of register, entry delay, duration, rest, dynamics and performance, the LIST-TABLE-ENSEMBLE principle explained in section 2 applies practically schematically; they will therefore be discussed first. Only the instrument and harmony parameters deviate. All explanations in sections 7 and 8 refer to the read format (see section 10). A number of additional indications about the structure variant are in section 9. However, the reader is recommended at this point to refer to the table of programme entries (APPENDIX) for purposes of comparison.

7 SCHEMATIC PARAMETERS

The "pattern" of the six schematic parameters consists in each case of 4 programme entries for

(a) reading the list,
(b) reading the table-groups,
(c) assembling the ensemble,
(d) assembly of the score.

The ensemble can be formed by calling selection programmes ALEA, SERIES and SEQUENCE; the score can be assembled by calling selection programmes ALEA, RATIO, GROUP, SEQUENCE and TENDENCY. If we indicate the elements with r and the indices with i, the table in fig. 7-1 results.

We must point out once more that in entry SEQ-GR the number of groups to be selected depends on the "combination"; one group "without combination", selection by the called selection principle; the same number and selection of groups as prescribed for INSTRUMENT "with combination". "With union" the selected groups are treated in the ensemble as one group (result: one layer), "without union" the groups remain, as it were, separate in the ensemble (result: the same number of layers as there are groups in the INSTRUMENT ensemble).

If selection programme SEQUENCE is used, care must be taken. The indices in parentheses must really be available as table-groups or

entry		comment
L	r,r,...,r	*enumeration of any number of elements*
TAB	i,i,...,i i,i,...,i ...	*any number of groups, any number of elements per group, consisting of list indices*
SEQ-GR	1 2 5 (i,i,...,i)	*3 alternative selection programmes; see section 4 for the meaning of the call numbers*
SEQ	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)	*5 alternative selection programmes; see section 4 for the meaning of the call numbers*

Fig. 7-1

ensemble elements. This is not the case as a rule if the number or order of the groups is determined at random. Two examples are shown in example 7-2.

An important principle applying to all "schematic" parameters: if parameters are to be combined, the number of table-groups in the combined parameters must be <u>the same as</u> the number of table-groups in INSTRUMENT; in the uncombined parameters the number is free. This last also applies if the combination is not used (COMB = 0, see 9.6).

Example 7-2

(1) *Given:*

L	*r1, r2, r3, r4, r5, r6*
TAB	*1* *2, 3* *4, 5, 6*
SEQ-GR	*1*
SEQ	*3 (1, 2, 3, 4, 5, 6)*

If only one group can be selected, it remains uncertain as to how many elements arrive in the ensemble in this way. Not until the group itself is named can the order of the ensemble elements be determined by SEQUENCE:

SEQ-GR 5 (3)
SEQ 5 (2, 1, 3)

Things are different when in all cases only the first element of the respective group is to be used:

SEQ-GR 1
SEQ 5 (1)

(2) *Given:*

L	*r1, r2, r3, r4, r5, r6*
TAB	*1, 2* *3, 4* *5, 6*
SEQ-GR	*2*
SEQ	*5 (4, 3, 2, 1)*

If two groups must always be selected (by "combination"), four elements land in the ensemble each time. SEQUENCE can therefore be used to make selections for the score.

Note, that here selection programme SERIES does not definitely exclude repetitions for SEQ-GR. SERIES could generate the index sequence 2 3 1 1 2 3 (two "series" without repetition: 2 3 1 and 1 2 3), thus forming ensembles (2 3), (1 1) and (2 3). In the second ensemble the elements would then repeat themselves: 1 2 1 2; the calling of SEQUENCE for the score would then result in the series 2 1 2 1 (= r2 r1 r2 r1). However, SEQUENCE is called as often as elements are required for the score; if the number of elements required is greater than the number of indices in parentheses, the contents of the parentheses are repeated a corresponding number of times.

7.1 REGISTER

For the harmony (see 8.2) a distinction is made between "relative" and "absolute" pitches. "Relative" pitches (regardless of the octave range) are indicated by the numbers from 1 to 99, "absolute" pitches by the octave number (1 to 9) and the relative tone number. The absolute pitches 101 to 199, 201 - 299 etc., 901 - 999 are accordingly available for the pitch range. Tones to be played by percussion instruments of undefined pitch are given the relative pitch 0. The number of relative pitches per octave is fixed by the composer as tr (see 9.3).

Registers are expressed in absolute pitches and indicate the range in which relative pitches can be localized. A register is defined by two absolute pitches. In the register (401,512) the relative pitch 5 can occupy two positions: 405 or 505. In the register (401,504) relative pitch 5 can only occupy position 405. Relative pitch 5 can not be accommodated in register (401,404). Absolute pitches may not, of course, be greater then tr per octave; the programme searches among the available relative pitches until an "allowed" pitch is found. If this is not possible, a "wrong" pitch is inserted and provided with a comment.

Registers for percussion instruments are indicated by (0,0).

There may be any number of registers in the list. The individual registers may overlap at will. Tendencies in the pitch-space can only be caused by the REGISTER parameter, see examples 7-4 and 7-5.

Fig. 7-6 shows schematically the four combinations of REGISTER, HARMONY and INSTRUMENT.

If selection principle TENDENCY is called for the "order", the mask's position is calculated for each tone.

CALL	CALL NUMBER	CODE	FORMAT
register list	25	L-REG	t1,t2,t1,t2,...,t1,t2
register table	26	TAB-REG	i,i,...,i i,i,...,i ...
group selection	27	SEQ-GR-REG	1 2 5 (i,i,...,i)
order	28	SEQ-REG	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)

Fig. 7-3

Survey of REGISTER parameter

t1 and t2 are the lower and upper limits of a register ($t1 \leq t2$) and form a pair of numbers with a common index. If percussion instruments also occur in the INSTRUMENT list (but only if they do), register 0,0 must be given; it may only occur once in the list and must be at the top of it (is thus always indicated by the index 1 in the table). The function of the 0,0 register depends on the position of the REGISTER parameter in the hierarchy (see fig. 7-6):

- if REGISTER precedes HARMONY, a 0-pitch is selected for the 0,0 register;
- if REGISTER precedes INSTRUMENT, no melody instruments are selected for the 0,0 register;
- if HARMONY precedes REGISTER, the 0,0 register can not be used for tones already selected; if the relative pitch 0 has been selected for HARMONY, the C,0 register is inserted;
- if INSTRUMENT precedes REGISTER, the 0,0 register can not be selected for a melody instrument, but is inserted for a percussion instrument.

Example 7-4

L-REG	*301, 309, 305, 401, 309, 405, 401, 409, 405, 501, 409, 505, 501, 509, 505, 601*
TAB-REG	*1, 2, 3, 4, 5, 6, 7, 8*
SEQ-GR-REG	*1* (dummy for the impossibility of selecting among several groups)
SEQ-GR	*6 (100, 0, 12.5, 87.5, 100)*

If HARMONY or INSTRUMENT should precede REGISTER, 0,0 register is indicated for pitch 0 or a percussion instrument, whilst the "mask" of the TENDENCY principle still keeps moving. If no tone is available in the "current register" (the "visible" section of the ensemble) for the previously selected instrument, this "wrong" register is still inserted, provided with a comment. The same applies vice versa if the instrument chosen later cannot play any tones in the already selected register. If however, as in the example, a register has been fixed (not 0,0), a relative pitch is inserted in HARMONY and thus a melody instrument in INSTRUMENT, should it follow.

Example 7-5

L-REG	*0, 0, 301, 309 ... (as in example 7-4) ... 505, 601*
TAB-REG	*1, 2, 3, 1, 4, 5, 1, 6, 7, 1, 8, 9*
SEQ-GR-REG	*1*
SEQ-REG	*6 (100, 0, 16.6, 83.3, 100)* (percentages altered because now 2 elements in mask)

If it were desired to have REGISTER first and still have a possibility of percussion instruments, the 0,0 register would have to be inserted from time to time. Since selections from the "visible" part of the ensemble are made by ALEA, it is possible, if the mask leaves registers 1 and 2 visible, to select either the 0,0 register or the tone register. Further decisions as to pitch and instrument (or vice versa) would then depend on this selection. The same applies the other way round: if pitch and/or instrument have already been fixed, a suitable register is sought within the mask. If there is no suitable register (e.g. (2,3) in the mask after pitch 0 or a percussion instrument has been fixed), a comment is provided, indicating the conflicting circumstances.

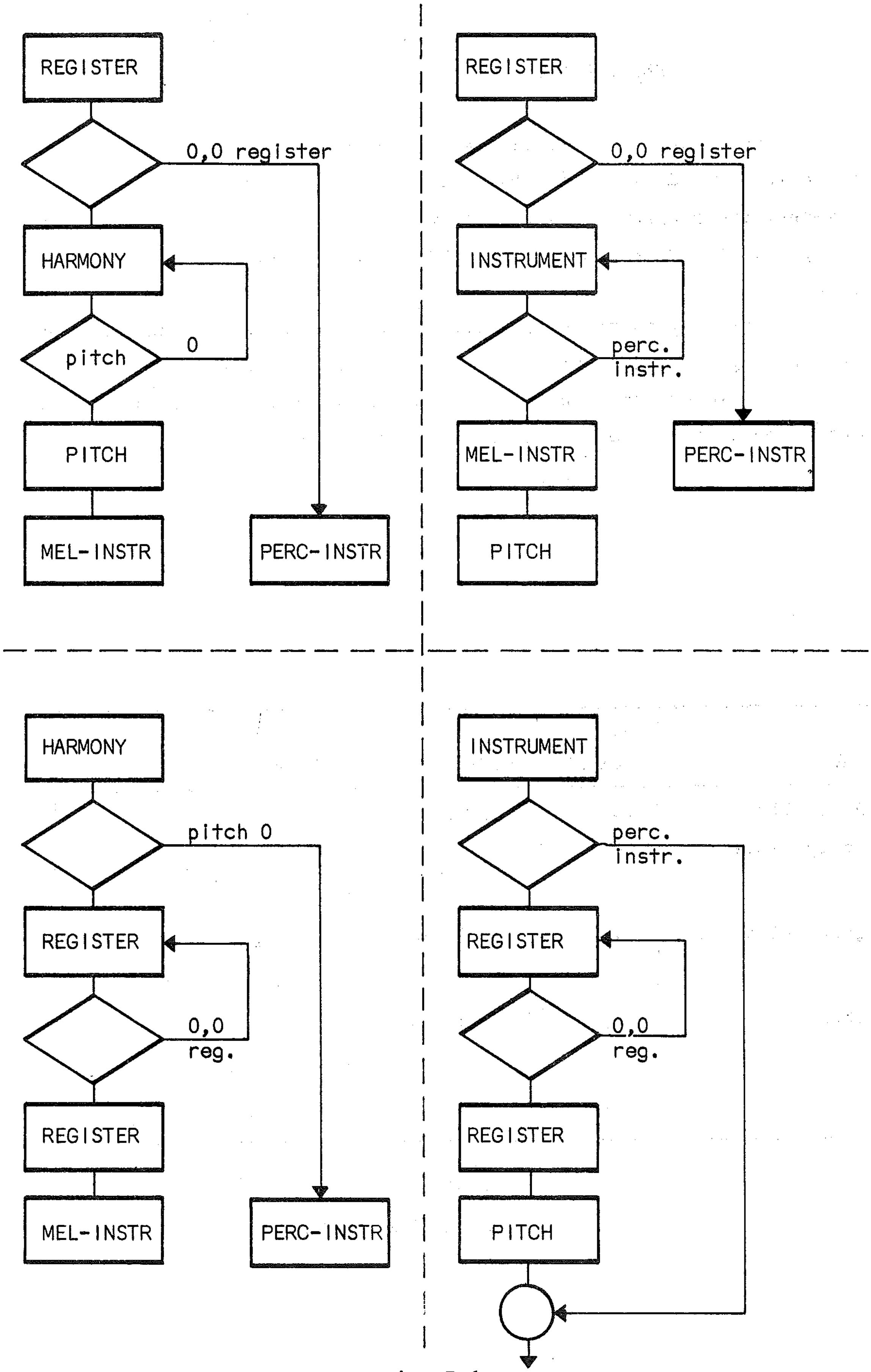

REGISTER
0,0 register
HARMONY
pitch
0
PITCH
MEL-INSTR
PERC-INSTR
REGISTER
0,0 register
INSTRUMENT
perc. instr.
MEL-INSTR
PERC-INSTR
PITCH
HARMONY
pitch 0
REGISTER
0,0 reg.
REGISTER
MEL-INSTR
PERC-INSTR
INSTRUMENT
perc. instr.
REGISTER
0,0 reg.
REGISTER
PITCH

Fig. 7-6

7.2 ENTRY DELAY

Entry delays are expressed in seconds. In the check printout of the composer's data and in the score only two decimal places are printed; however, the programme calculates to a maximum degree of accuracy and accepts up to 12 decimal places. (The same applies to durations and rests.)

With regard to the relation between entry delay and duration see 7.3 Fig. 7-7 is a survey of the programme entries.

If selection principle TENDENCY is called for the "order", the mask's position is calculated for each entry point.

CALL	CALL NUMBER	CODE	FORMAT
delay list	29	L-ENTRY	r,r,...,r
delay table	30	TAB-ENTRY	i,i,...,i i,i,...,i ...
group selection	31	SEQ-GR-ENTRY	1 2 5 (i,i,...,i)
order	32	SEQ-ENTRY	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)

Fig. 7-7

Survey of ENTRY DELAY parameter

7.3 DURATION

Durations, like entry delays, are expressed in seconds. The "chart" of DURATION has two entries more, in order to be able to determine the ratio to the entry delays and durations in chords with several tones.

For the ratio of the entry delay to the duration, there is a choice of three alternatives (see fig. 7-8, DUR-ENTRY).

Basically, rests occur if the duration is shorter than the entry delay (this has nothing to do with REST); no rests occur if entry delay and duration are equal; superpositions occur if the duration is longer than the entry delay. See example 7-9 for the treatment of these two parameters.

As well as the possible dependency of duration on entry delay (or vice versa), there is also a basic dependency on the instrument. If INSTRUMENT comes first, only one allowed duration can be selected for the respective instrument; if DURATION comes first, the choice of instruments is confined to those which are allowed to play the given duration (see 8.1).

The "mode" (entry 38) regulates the relationship of durations in the chord (see fig. 7-8, MOD-DUR).

If selection principle TENDENCY is called for the "order", the calculation of the positions of the mask depends on the mode; it occurs

- if MOD-DUR = 0 per entry point,
- if MOD-DUR = 1 per tone.

If the duration list contains the duration 0, there must be at least one instrument with a lower duration limit of 0.

CALL	CALL NUMBER	CODE	FORMAT
duration list	33	L-DUR	r,r,...,r
duration table	34	TAB-DUR	i,i,...,i i,i,...,i ...
group selection	35	SEQ-GR-DUR	1 2 5 (i,i,...,i)
order	36	SEQ-DUR	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)
relation	37	DUR-ENTRY	1 *(duration = entry delay, MOD-DUR = 0)* 2 *(duration ≤ entry delay)* 0 *(duration independent)*
mode	38	MOD-DUR	1 *(different durations in chord)* 0 *(equal durations in chord)*

Fig. 7-8

Survey of DURATION parameter

If DUR-ENTRY = 1, the first parameter determines the common time-values, which are then adopted for the other parameters; the ensemble of the second parameter is not taken into account. Of course this mechanism only functions when the tones in the "chord" (tones with the same entry-point) are of equal length. Therefore we must have MOD-DUR = 0. If the hierarchy is established by random decision, it is necessary that the same instructions be issued for both parameters (same lists, same tables, same principle for selection of groups and elements). If, on the other hand, the hierarchy is established by the composer, it is sufficient to provide the entries of the previous parameter with the necessary data; then only dummies have to be inserted for later parameters (see APPENDIX). In other words, the call number for DUR-ENTRY also applies if the durations are fixed first. IMPORTANT. Entries 29-32 must in any case be completely equipped with data for the "initial set" (see section 10), because the "average entry delay" can not otherwise be calculated. -
If DUR-ENTRY = 1, INSTRUMENT and ENTRY DELAY coming first, the durations (always the same as the entry delays) can come into conflict with the instrument definitions. In such cases they are shortened to the "allowed" duration if necessary, but never lengthened.

If DUR-ENTRY = 2, valid lists must be given and valid ensembles must be formed for both parameters. If ENTRY DELAY comes first, elements are rejected in the DURATION ensemble if they are greater than the selected entry delay. If no "allowed" durations are available, a "wrong" duration is provided with a comment. If, on the other hand, DURATION comes first, elements in the ENTRY DELAY ensemble which are smaller than the selected duration are rejected. If such "allowed" entry delays are not available, the "wrong" entry delay is provided with a comment.

If DUR-ENTRY = 0, there must be valid lists and ensembles for both parameters. However, the two parameters are computed independently of one another, regardless of the hierarchy.

If MOD-DUR = 1, one duration per tone is selected;
if MOD-DUR = 0, one duration per entry point is selected.
Since chords still have to be "scored", the mode becomes dependent on the selected instruments, or instruments to be selected. With regard to the "vertical density" we distinguish three possibilities (compare 5.2, but see also 9.2):

the dependency on the instrument,
the dependency on the harmony (chord principle),
autonomous density.

In the first case INSTRUMENT is main parameter, the instrument thus being fixed before one or more durations can be selected. -
In the second case the pitches are already given (HARMONY being main parameter), the order of INSTRUMENT and DURATION still being open. -
The same applies to the third case, where it is only the number of tones per chord that is fixed, the choice of main parameter being free. -
In the last two cases we can distinguish two situations:

(a) DURATION precedes INSTRUMENT. If the durations in the chord are equal, instruments can only be selected which can play the selected duration. If the durations in the chord are not the same, this question is posed for each duration and instrument.

(b) INSTRUMENT precedes DURATION. A common duration can only be inserted if it is allowed for all involved instruments. Various durations are compared one by one with the instruments which are supposed to play them. This arrangement also applies to the first case (INSTRUMENT main parameter).

Example 7-9

(a) *Duration same as entry delay (ENTRY DELAY has priority in the hierarchy):*

L-ENTRY	0.1, 0.2, 0.3, 0.5, 0.8, 1.3
TAB-ENTRY	1, 2, 3, 4, 5, 6
SEQ-GR-ENTRY	1
SEQ-ENTRY	3 (4, 3, 2, 2, 1, 1)
L-DUR	0
TAB-DUR	1
SEQ-GR-DUR	1
SEQ-DUR	1
DUR-ENTRY	1
MOD-DUR	0

(b) *Duration not longer than entry delay:*

L-ENTRY	0.1, 0.2, 0.3, 0.5, 0.8, 1.3
TAB-ENTRY	1, 2, 3, 4, 5, 6
...	
L-DUR	0.1, 0.2, 0.3, 0.5, 0.8, 1.3
TAB-DUR	1, 2, 3, 4, 5, 6
...	
DUR-ENTRY	2
MOD-DUR	1 *or* 2

For each entry delay we have an equal duration and (with the exception of the shortest duration) at least one shorter duration - regardless of the hierarchy. In the case of MOD-DUR=1 this also applies to all the tones in a chord.

(c) *Duration usually shorter than entry delay.*
If the durations may occasionally be longer than the entry delays, suitable lists can be made:

L-ENTRY	0.4, 0.5, 0.7, 0.9, 1.2, 1.5
...	
L-DUR	0.1, 0.2, 0.3, 0.4, 0.5, 0.7
...	
DUR-ENTRY	0
MOD-DUR	1 *or* 0

(d) *Ratio entry delay/duration is constant.*
For this the two parameters can be coupled by the SEQUENCE principle if the "series" of entry delays (and durations) may be repeated literally:

L-ENTRY	0.3, 0.5, 0.8, 1.3, 2.1
...	
SEQ-ENTRY	5 (2, 5, 4, 1, 3)
L-DUR	0.2, 0.3, 0.5, 0.8, 1.3
...	
SEQ-DUR	5 (2, 5, 4, 1, 3)
DUR-ENTRY	0
MOD-DUR	0

7.4 REST

Rests, like entry delays and durations, are expressed in seconds. The "chart" contains not only the four standard entries but also a fifth one for the "mode".

In the case of rests, too, we can also speak of entry delay and duration: the entry delay from one rest to the next, and the duration of the rest. But whereas with tones the entry delay and duration are calculated starting from the same entry point, the rest delay starts at the end of the rest. This arrangement was made because rests ought not to overlap in the same way as tones do, and their entry delays may therefore not be smaller than their durations. In order to clarify the terms we shall speak of "rest duration" (PD) and "rest delay" (PE). We shall call the rests which are inserted because of the REST parameter "autonomous rests", and those resulting from the duration of a tone being shorter than its entry delay "pseudo rests". Fig. 7-11 shows graphically the difference between the entry delay and the rest delay. The first rest delay in a structure variant is calculated starting from the beginning of the variant.

To determine the rest entry we have the terms "sound entry" and "general entry". Sound entry is used to denote every entry point at which a tone begins, regardless of previous tones which might still be "sounding". A general entry, on the other hand, is a sound entry not overlapped by sustained tones; it can be said to be preceded by a "general rest" (whose value can be 0). The composer can use the "mode" (entry 43) to indicate whether the autonomous rests should start instead of a sound entry or instead of a general entry. If they should start instead of a sound entry, any tones that might still be "sounding" are sustained according to their duration. But all tones starting at the sound entry, and all tones starting later, are shifted on in time by the length of the rest. It can occur here that the rest duration is shorter than the duration of the sustained tones. In this case the rest is "covered" ("concealed rests"). Rests are always calculated and inserted per layer.

CALL	CALL NUMBER	CODE	FORMAT
rest list	39	L-REST	r,r,...,r
rest table	40	TAB-REST	i,i,...,i i,i,...,i ...
group selection	41	SEQ-GR-REST	1 2 5 (i,i,...,i)
order	42	SEQ-REST	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i)
mode	43	MOD-REST	0 *(no rests)* 1 (d1,d2) *(rest before next sound entry)* 2 (d1,d2) *(rest before next general entry)*

Fig. 7-10

Survey of REST parameter

Selection programme TENDENCY is missing here from the entry SEQ-REST, because the number of rests to be inserted is not generally large enough for a "transition".

In the last entry the REST parameter can be de-activated by means of MOD-REST=0. The data for entries 39 to 42 are then ignored. This can be useful for altering the other parameters if there is a large number of tests, whilst leaving the rest-data unchanged, and only "activated" when required. If the REST parameter should basically be not used at all, dummies (see APPENDIX) can be given for entries 39 to 42.

In the cases of MOD-REST=1 and MOD-REST=2, rests are inserted after all the other parameters have been computed; in other words, REST always comes last and has no part in the hierarchy (can thus never be main parameter). Exceptions:

(a) "no union, layers with common harmony" (see 6.2); since the common harmony of several layers cannot be computed until the rhythmic context (including rests) has been established, REST is computed in this case at the last place but one, HARMONY coming last.

(b) "performance per rest" (see 7.6); since performance cannot be computed until the rests have been established, REST is computed at the last place but one, PERFORMANCE coming last.

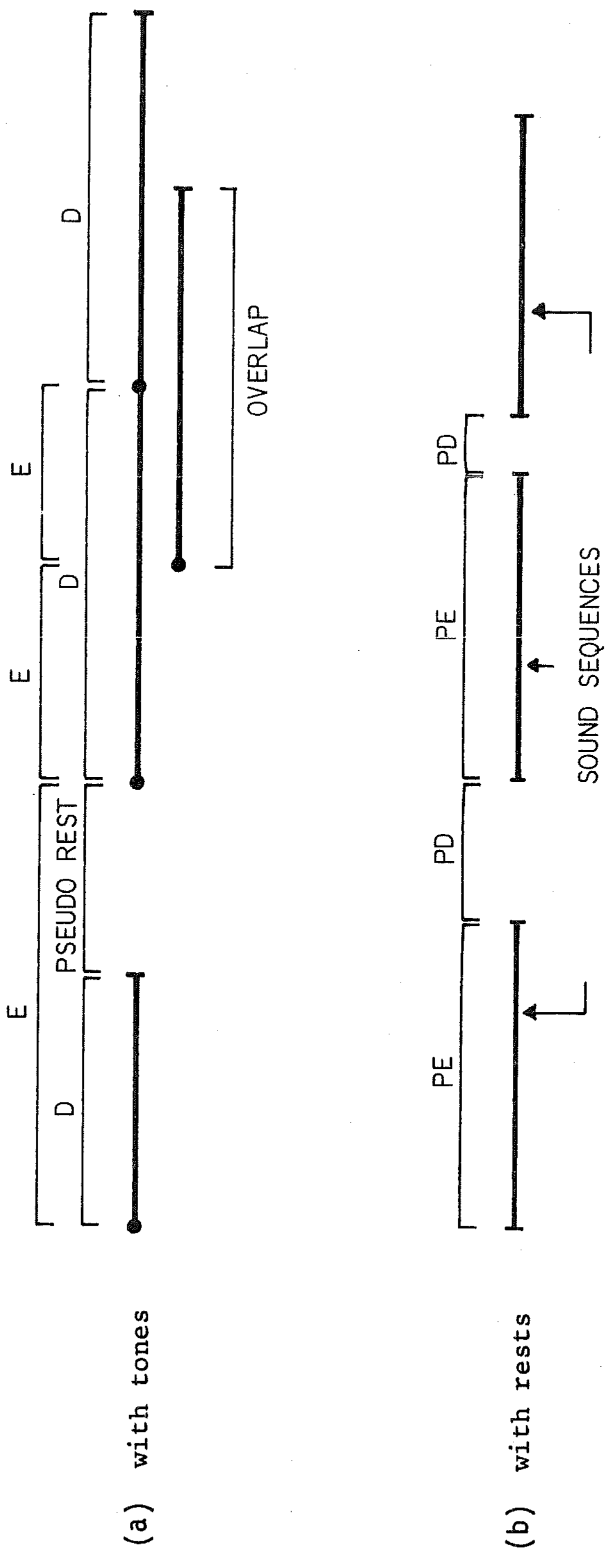
(c) combination of (a) and (b); order of the last three parameters in the hierarchy: REST, PERFORMANCE, HARMONY.

In order to calculate the entry points of rests, the composer declares an "entry range" (d1 and d2 in entry 43). Both limits of the range are stated in percentages of the structure duration. ALEA is used to find a "provisional" entry point (dp) between these range limits; the next sound or general entry is finally calculated starting from dp. This is then the "definite" rest entry. It can occur that a pseudo rest precedes a general entry (any non-overlapped sound entry is also regarded as a general entry), so that the autonomous rest is lengthened by this pseudo rest. Fig. 7-12 shows schematically the various time-points during the search for the definite rest entry.

A one-sided dependency exists between REST on the one hand and ENTRY DELAY and DURATION on the other. For whether a sound entry or general entry can be found in the entry range depends on the length of the entry delays or on the ratio of the entry delays to the durations. Especially general entries can, because of numerous overlappings, be so scarce that there is barely an opportunity for inserting rests in a structure variant. The frequency of sound entries, on the other hand, is only dictated by the average entry delay; durations of extra length can not prevent rests, although they can conceal them.

The "entry range" is used exclusively to determine the provisional entry point; the definite entry point can by all means be outside this range. When stating the extreme values d1 and d2, we must be aware of the fact that there is a direct dependency between these quantities and the number of expected rests. If MOD-REST=1 (5,15) and the "average entry delay" $\leq$ 1/10 of the structure duration, we can expect a rest around every 10% of the structure duration; this is 9 or 10 rests. The shortest rest delay would be about 5%, the longest about 15%. Approximately the same number of rests could be obtained with MOD-REST = 1 (8,12) or even 1 (10,10); the rest delays would then admittedly be very similar or even the same (if the entry delays are very short in relation to the structure duration).

The time-shift of all tones starting simultaneously or later is made per rest entry, so that the calculation of the next rest entry refers to the respective corrected rhythmic context. d1 and d2 always refer to the structure duration stated by the composer (entries 51 and 52, see 9.1.3), and thus not to the respective remaining duration of the structure. The effective structure duration exceeds the structure duration stated by the composer by the sum of all autonomous rests.



<u>Fig. 7-11</u>

entry delays and durations

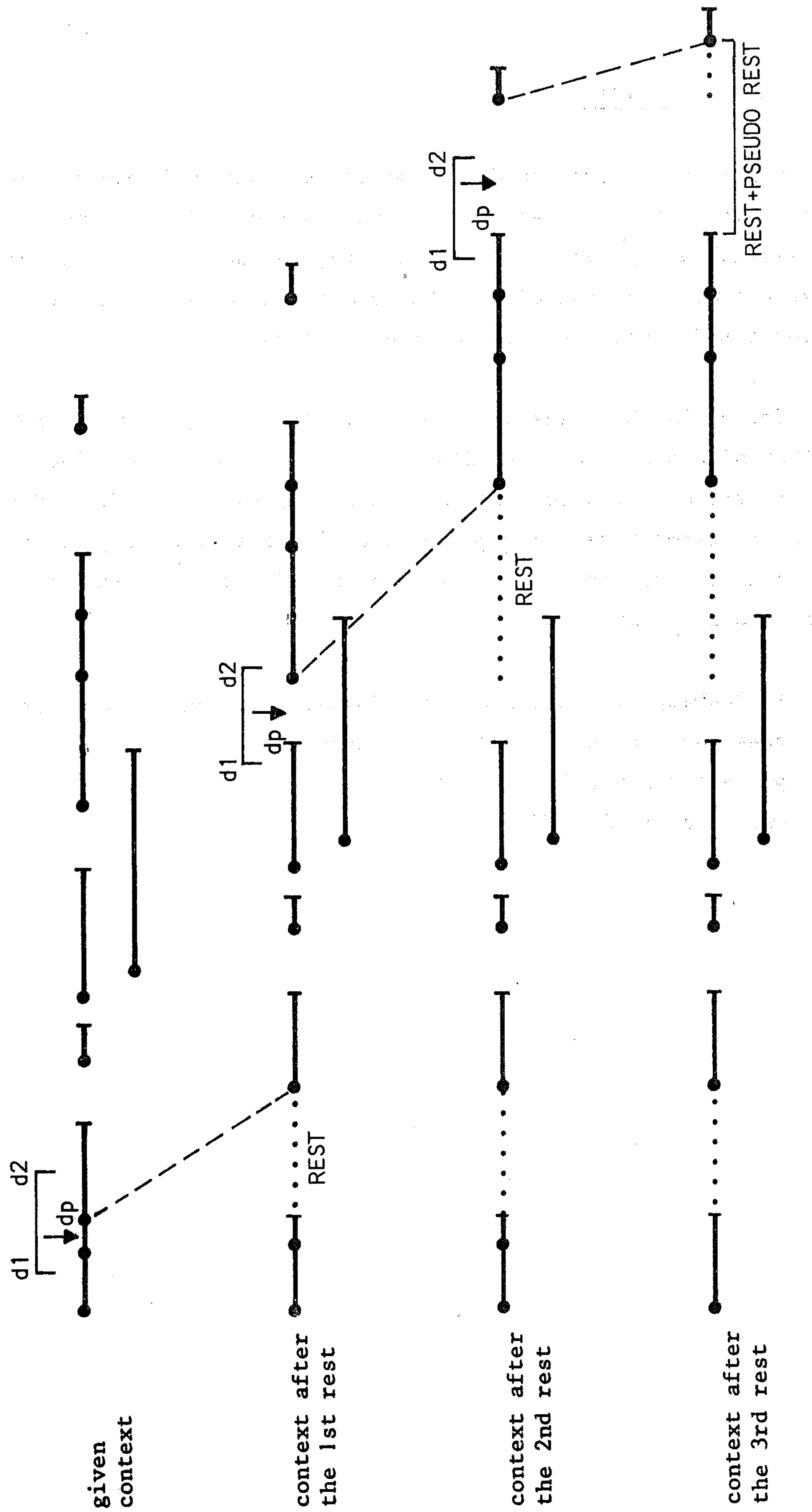

given context
context after the 1st rest
context after the 2nd rest
context after the 3rd rest
d1
d2
dp
REST
REST+PSEUDO REST

Fig. 7-12

7.5 DYNAMICS

Dynamics are expressed as symbols. Besides combinations of letters, numbers may be used, but especially the symbols < and >. These "swells" apply, as do all other data, to one entry-point (or tone) at a time. The "chart" for the dynamics also has an additional entry for a mode (for the ratio of the dynamics in the chord).

The DYNAMICS parameter depends hierarchically only on INSTRUMENT. If INSTRUMENT comes before DYNAMICS, dynamics may only be selected for a given instrument if this instrument can play them. If DYNAMICS come first, an instrument can only be selected for the given degree of intensity if it can play this intensity. See 8.1 with respect to instrument definitions.

If TENDENCY is called for the "order", the calculation of the mask's edges depends on the mode, and occurs

- per entry point if MOD-DYN = 0,
- per tone if MOD-DYN = 1.

CALL	CALL NUMBER	CODE	FORMAT
dynamics list	44	L-DYN	w,w,...,w
dynamics table	45	TAB-DYN	i,i,...,i i,i,...,i ...
group selection	46	SEQ-GR-DYN	1 2 5 (i,i,...,i)
order	47	SEQ-DYN	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)
mode	48	MOD-DYN	1 *(different intensities in chord)* 0 *(equal intensities in chord)*

Fig. 7-13

Survey of DYNAMICS parameter

The mode refers to the ratio of the dynamics within a chord. If we have MOD-DYN = 0, only one intensity is selected for each entry-point (and thus for all tones starting at the same time), whilst MOD-DYN = 1 means that a new intensity is inserted for each single tone (regardless, then, of the distribution of the tones in chords).

7.6 PERFORMANCE

Manners of performance are expressed as symbols. The "chart" contains an extra entry for a mode by which manners of performance can be altered for each tone, chord or rest. The selection principle to be indicated for entry 61 refers to this mode.

PERFORMANCE is a later addition to PR-2. This is why the entries of this parameter come after entry 57, originally the last entry, instead of following entry 48.

The indication "performance" does not express exactly the possibilities offered by this parameter. Apart from the mode it does not really differ from DYNAMICS: the lists of both contain symbols, both adhere to the LIST-TABLE-ENSEMBLE principle, the same selection principles are available to both, both can be main parameters, both depend on the INSTRUMENT. The difference, however, is in the possible dependency of the REST. By "performance", we do not only mean performance techniques in the stricter sense (staccato, legato, fluttertongue, etc.), but also instructions for phrasing in general, directions for performance, tempo indications (metronome tempi, "faster", "slower", etc.); in short, PERFORMANCE is an articulation parameter.

If the REST parameter's only task is to divide the structure variant up into groups to which the articulation parameter then refers, we can work with the rest value of 0. In this case nothing is changed in the rhythmic context, a new mode of performance occurring where the 0 rest is inserted. The two parameters, DYNAMICS and PERFORMANCE, are of course interchangeable, so that the articulation can be composed with the former and the intensity with the latter. Care must however be taken that this does not change anything in the instrument definition (8.1).

The list of modes of performance should be worked out together with the INSTRUMENT parameter (see 8.1), since each instrument must be characterized by a range of articulation. By "range" we mean an uninterrupted sequence of list-elements. If several instruments are to use the same modes of performance in different combinations, it can be difficult to arrange the elements in such a way that each instrument

CALL	CALL NUMBER	CODE	FORMAT
performance list	58	L-PERF	w,w,...,w
performance table	59	TAB-PERF	i,i,...,i i,i,...,i ...
group selection	60	SEQ-GR-PERF	1 2 5 (i,i,...,i)
order	61	SEQ-PERF	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)
mode	62	MOD-PERF	0 *(new performance for each chord)* 1 *(new performance for each tone)* 2 *(new performance for each rest)*

Fig. 7-14

Survey of PERFORMANCE parameter

The mode refers to the order of ensemble elements and indicates when the choice of a selection principle means that a new manner of performance should be inserted in the score. MOD-PERF = 0 generates a new performance for each entry point; MOD-PERF = 1 does the same for each tone (under certain circumstances, thus, more than once for each entry point). If a new mode of performance is to be selected by means of MOD-PERF = 2, PERFORMANCE moves to the last place in the parameter hierarchy, REST to the last but one. If, however, we have "no union" at the same time too (compare 6.2), resulting in the same number of layers as there are groups in the INSTRUMENT ensemble, HARMONY is then the last parameter, PERFORMANCE last but one, with REST before it.

If there is "no union", and MOD-PERF = 0 or MOD-PERF = 1, HARMONY comes last, REST last but one; the position of PERFORMANCE is then free. Hierarchy and its restrictions are discussed in 9.4 in context.

has a "range". In circumstances such as these the user must not hesitate to name each mode of performance in the list as often as is necessary for the definitions of the instruments, even if this means that each instrument has its own group of "performances". The list-elements do not have to be quoted in full in the table, so that the list of definitions can be adapted to the instruments, but the table can be adapted to the selection mechanisms for the score.

If selection principle TENDENCY is to be called for the "order", the calculation of the position of the mask depends on the mode; this is done:

- if MOD-PERF = 0 per entry point,
- if MOD-PERF = 1 per tone,
- if MOD-PERF = 2 per rest.

In the last case we must be prepared for the mask's position to change by leaps and bounds, especially if only a few rests are inserted. In order to have this under control, it is better either to have only a few modes of performance or to have the mask big enough for there to be sufficient elements for the aleatoric principle within the mask.

Example 7-15

Let us assume that the modes of performance a, b, c, d, e are to be distributed among instruments A, B and C as follows:

instrument	*performance*
A	*a b c*
B	*a b d*
C	*a b d*

The list of performances (call 58) might then be:

c	*a*	*b*	*d*	*a*	*b*	*e*

This would result in the following ranges for the definition of the instruments (call 7):

1, 3, 2, 4, 5, 7

The distribution of the performance modes among table-groups (call 59) does not formally depend on this; the indices would be 2, 3, 1, 4, 7 for performance modes a, b, c, d, e.

8 DEVIATING PARAMETERS

The parameters deviating from the "chart" (see fig. 7-1) refer to instruments and harmony. They are discussed individually in the following. For additional indications which can be expected of the programme, see section 9.

8.1 INSTRUMENT

PR-2 is chiefly intended for the composition of instrumental music; the instruments required for the performance of a composition must therefore be defined in many respects. These definitions yield dependencies (in terms of the "hierarchy") on other parameters which have already been mentioned (see section 7) or which will be mentioned (8.2). Nonetheless, these dependencies will be dealt with in detail here.

The instruments are divided into melody and percussion instruments. The composer decides which instruments are to be treated as melody or percussion instruments. It might be conceivable to treat a violin as a percussion instrument, whereas cow-bells might be treated melodically. Instruments are defined in the following parameters:

- name of instrument,
- mode of performance,
- pitch compass,
- playable durations,
- playable dynamics,
- chord size.

As well as these definitions, the survey for INSTRUMENT also contains the "schematic" entries for table, group selection and order (entries 12-14).

CALL	CALL NUMBER	CODE	FORMAT
instrument list	6	L-INSTR	w,w,...,w
modes of performance	7	AMB-PERF	a,z,a,z,...,a,z -a,-z,...,-a,-z
pitches	8	AMB-FRQ	t1,t2,t1,t2,...,t1,t2 0,n,0,n,...,0,n
durations	9	AMB-DUR	a,z,a,z,...,a,z -a,-z,...,-a,-z
intensities	10	AMB-DYN	a,z,a,z,...,a,z -a,-z,...,-a,-z
chord sizes	11	AMB-CHORD	a,z,a,z,...,a,z
instrument table	12	TAB-INSTR	i,i,...,i i,i,...,i ...
number of groups (p)	13	N-GR-INSTR	1 2 5 (i,i,...,i)
group selection (q)	13	SEQ-GR-INSTR	1 2 5 (i,i,...,i)
union (s)	13	UNION	0 *(union, one layer)* 1 *(no union, common harmony)* 2 *(no union, harmony per layer)*
order	14	SEQ-INSTR	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)

Fig. 8-1

Survey of INSTRUMENT parameter

In the following commentary to the survey 8-1, the entries for INSTRUMENT are dealt with singly. For purposes of orientation, a line from the survey heads each explanation.

instrument list	6	L-INSTR	w,w,...,w

For each instrument a "name" is given, a name being in the form of a symbol. The symbol may not contain the comma (,) or semicolon (;). The names in the instrument list should differ in at least one sign so that they are unambiguous in the score output. They are adequately identified by their list index for the programme. If for any reason an instrument is to be named more than once, it is advisable to provide the name with a continuous number, e.g. VC1, VC2, VC3, etc. Instruments occurring multiply in the orchestra (choral setting) need only be named once in the list; it is sufficient to state their index a corresponding number of times in one of the table-groups.

modes of performance	7	AMB-PERF	a,z,a,z,...,a,z -a,-z,...,-a,-z

For each instrument a performance range is given, referring to the L-PERF list (see 7.6). The limit values a and z must be indices and indicate modes of performance which the relevant instrument may execute. Both limit values must be either positive or negative; positive limit values contain the "allowed" modes of performance, negative values contain "prohibited" ones, themselves belonging to the "allowed" ones.

Example: *(3,5) means that elements with an index <3 or >5 may not be used. (-3,-5) means that elements with an index >3 and <5 may not be used. Note that these indices always refer to the list (thus not to the ensemble). Positive and negative pairs of numbers are alternatives; it is not possible to combine indices with different signs.*

Each pair of numbers is registered by the programme under a common index, so that a performance range corresponds to each instrument under the same index. If different instruments are to be given the same mode of performance or vice versa, the L-PERF list must be constructed in such a way (if necessary by stating the same mode of performance several times) that each instrument can be characterized by a single range; it is then impossible to name several ranges for one instrument.

The performance range participates in the hierarchy: for a given instrument, only modes of performance can be selected which are in the performance range; for a given mode of performance, only instruments can be selected which may use this mode of performance. If an allowed element can not be found, a "wrong" one is inserted and provided with a comment.

pitches	8	AMB-FRQ	t1,t2,t1,t2,...,t1,t2 0,n,0,n,...,0,n

The pitch range is given in the form of a pair of numbers for each instrument, the numbers being registered by the programme under a common index. The pairs of numbers (t1,t2) and (0,n) are alternatives and can not be mixed.

t1 and t2 are absolute pitches for melody instruments, t1 indicating the lower limit, t2 the upper one (e.g. 408,612). The pair of numbers (0,n) indicates percussion instruments; the constant 0 informs the programme that a percussion instrument is meant and that n ranges are available (e.g. 0,3 for a percussion instrument with three ranges, perhaps low, medium, high). ALEA selects among these ranges for the score. The PERFORMANCE parameter can be used for a more differentiated description of a percussion instrument; in this case n=1 would be stated and perhaps a separate instrument declared for each range if several are to be provided.

Pitch compass participates in the hierarchy: registers can only be selected for a given melody instrument if they have at least one tone in common with the respective pitch compass; a pitch is sought in the remaining compass. If the harmonic principle means that no pitch is available, a "wrong" pitch is inserted and provided with a comment.

Pitches can only be selected for a given melody instrument if they occur in its pitch compass (the programme only works with instruments with compasses of at least one octave). A suitable register is sought for the selected pitch; if not available, a "wrong" register is inserted and provided with a comment.

For a given percussion instrument, 0-pitch and 0-register are automatically inserted.

These rules also apply the other way round:

A melody instrument can only be used if its pitch compass has at least one tone in common with the given register.

A percussion instrument can only be used if the 0,0 register or pitch 0 have been selected beforehand. If no instrument can be found because of this condition, a "wrong" melody or percussion instrument is inserted and provided with a comment (compare fig. 7-6).

durations	9	AMB-DUR	a,z,a,z,...,a,z -a,-z,...,-a,-z

Independently of the list L-DUR (see 7.3), a duration range is indicated in seconds for each instrument; a and z marking the shortest and longest duration which the instrument may play. Both limit values must be either positive or negative; positive limit values contain the "allowed" durations, negative limit values contain "prohibited" durations, themselves belonging to the "allowed" ones.

Example: *(0,1) means that only short durations up to and including one second may be played. (-0.8,-3.5) means that the duration range between >0.8 and <3.5 is blocked (values 0.8 and 3.5 themselves are still allowed) and that therefore only very short or very* *long durations may be played.*

Every pair of numbers is given a common index, so that one duration range corresponds to each instrument. The duration range applies to both melody and percussion instruments.

The duration range participates in the hierarchy: durations can only be selected for a given instrument if they are in the allowed duration range.
This also applies the other way round: instruments can only be selected for a given duration if they can play this duration. If no allowed element can be found here, a "wrong" one is inserted and provided with a comment.

intensities	10	AMB-DYN	a,z,a,z,...,a,z -a,-z,...,-a,-z

A dynamic range, referring to the list L-DYN (see 7.5), is given for each instrument. The limit values a and z must be indices. a and z are either both positive or both negative, positive values contain "allowed" dynamics, negative values contain "prohibited" ones, themselves belonging to the allowed ones.

Example: *(3,5) means that dynamics with an index <3 or >5 may not be used. (-3,-5) means that dynamics with an index >3 and <5 may not be used (only index 4 is blocked in this case). Note that these indices always refer to the list (thus not to the ensemble).*

Each pair of numbers is given a common index, so that a dynamic range corresponds to each instrument. The dynamic range applies to both melody and percussion instruments.

The dynamic range participates in the hierarchy: dynamics can only be selected for a given instrument if they are in the allowed dynamic range.
The same applies the other way round: instruments can only be selected for a given intensity if they can play it. If no allowed element can be found, a "wrong" one is inserted and provided with a comment.

chord sizes	11	AMB-CHORD	a,z,a,z,...,a,z

A range of allowed chord sizes is given for each instrument, a representing the smallest chord and z the largest. ALEA selects between these limit values. Each pair of numbers is given a common index, so that a range of chord sizes corresponds to each instrument. The chord size range applies to both melody and percussion instruments.

The chord size plays an important part in the "instrumentation" of the score (compare 5.2 and 9.2). For whether the "vertical density", which basically is independent of the chord size, depends on the selected instrument, or is given by an already selected chord, or has priority above all other parameters as "autonomous" density: the chord size indicates the number of tones with which the respective instrument participates in the entire chord. If the vertical density depends on the instrument (INSTRUMENT then being main parameter), vertical density and chord size are identical; the chord size selected aleatorically between a and z for the respective instrument determines at the same time the total density of the respective entry point. In the other two cases the vertical density, stated as a chord or "autonomous", must still be scored for instruments, which means that another instrument must be selected, and for it a chord size between a and z, and that this must go on happening until the total density is reached. The chord size of the instrument last selected is reduced if necessary to the remaining number of tones in the entire chord.

This is done by selecting an instrument, and for it a chord size between a and z (corresponding to the instrument definition). The pitches of the chord are then compared with the instrument definition one by one, the instrument only being inserted for "allowed" (absolute) pitches and not more times than provided for between a and z. If there are fewer "allowed" tones than intended for the instrument, the number of tones for the instrument is reduced. If all the tones in the chord have been checked, or if enough "allowed" tones have been found beforehand, another instrument and chord size (between the appropriate limit values) is selected. This goes on until all chord tones have been "scored".

Since the chord sizes for percussion instruments are also defined, percussion instruments - and for each one a randomly selected number of tones between a and z - can participate in the "scoring" of the vertical density. A condition for this is however that the hierarchy has not already established pitches or registers which automatically exclude all percussion instruments.

Here we must recall (but compare section 3 too) the selection cycle, which is different for each selection principle. Apart from ALEA, the principles have finite selection cycles which are repeated as soon as the "supply" is exhausted. In this way very short selection cycles can mean that an instrument is selected several times in the same chord, although only one example of this instrument is present in the orchestra. This danger is especially acute in the case of chords with many tones played by instruments which can only play one tone at a time. However, the programme "expects" there to be enough instruments in the vertical density to score the entire chord without instrument repetition. If there are not enough instruments, each repeated instrument is provided with a comment which under normal circumstances tells the composer that the rules could not be adhered to. This mechanism might prove misleading if the composer is writing for, say, piano, determines the actual chord size by means of the vertical density (see 9.2) and defines the range of allowed chord sizes as (1,n). Since a random selection must occur between 1 and n, the entire chord might have to be performed by several pianos, not available in the supply. It is therefore advisable always to define the chord size as (n,n) for solo pieces; if necessary this chord size will then be reduced to the vertical density without comment during the "scoring".

instrument table	12	TAB-INSTR	i,i,...,i i,i,...,i ...

As in the "schematic" parameters, instruments can be put together into groups. There may be any number of groups, which may be of any size.

number of groups (p)	13	N-GR-INSTR	1 2 5 (i,i,...,i)
group selection (q)	13	SEQ-GR-INSTR	1 2 5 (i,i,...,i)
union (s)	13	UNION	0 *(union, one layer)* 1 *(no union, common harmony)* 2 *(no union, harmony per layer)*

Entry 13 represents a table with 3 groups, named p, q and s in the survey. p is the number of groups intended to form an ensemble, q the selection of groups for the ensemble, s regulates the "union" (see 6.2). Note that for p and q a choice can be made among selection Programmes 1, 2 and 5. The values for i (in selection programme 5) mean for p the number of groups, but for q their order. It goes without saying that the values for i in groups p and q may not exceed the number of table-groups in TAB-INSTR; the number of i's may not be greater than the number of groups either. Example 8-2 shows some constellations of selection programmes for p and q.

The third group of this table contains a call number for union.

If s = 0, all the groups in the ensemble are treated as one single group, so that the selection programme called for SEQ-INSTR extends to all the groups which have been put together.

If s = 1, the groups in the ensemble are dealt with separately; as many "layers" are computed as there are groups in the ensemble (for each layer one group - in the same order as the groups are in the ensemble). The same assembly (and thus the same distribution among layers) applies to all "combined" parameters, whilst for the ensemble a new group is selected for each layer in the uncombined parameters. HARMONY automatically becomes the last parameter (REST comes before HARMONY), so that the pitches are not inserted until all the layers have been put together to form a variant. If however the performance is changed per rest (MOD-PERF = 2), PERFORMANCE is inserted between REST and HARMONY.

If s = 2, the groups in the ensemble are also treated separately; however, HARMONY retains its place in the hierarchy and is computed per layer.

Example 8-2 (all examples refer to 6 table-groups)

(a) $p = 1, q = 1$

For each variant, ALEA selects between 1 and 6; p indicates the number of groups which are to be put together for the ensemble. This number is automatically adopted for all "combined" parameters. For each variant, ALEA selects p times between 1 and 6; the found numbers denote the groups which are to form the ensemble. This sequence of numbers is automatically adopted for all "combined" parameters.

(b) $p = 2, q = 2$

The same as in example (a), but SERIES selects instead of ALEA. Different selection programmes can of course be called for p and q, but note that in the case of q = 2 the same group can appear several times in the ensemble (compare commentary to example 7-2(2)).

(c) $p = 1, q = 5$

For each variant, ALEA selects the number of groups, the composer determining their order. Since the respective number is unknown, the respective combination is also undefined. For 4 variants, the numbers 2, 3, 2, 5 can result for p; if the composer has given q = 5 (1,2,3,4,5), the following groups are put together:

variant 1: 1, 2
variant 2: 3, 4, 5
variant 3: 6, 1
variant 4: 2, 3, 4, 5, 6

The result for p = 2 would be on the same lines.

(d) $p = 5, q = 1$

Four variants for p = 5 (1,2,3,4):
variant 1: 5
variant 2: 3, 6
variant 3: 1, 2, 1
variant 4: 3, 4, 3, 5

The results for q = 2 would be on the same lines.

Note that the groups arrive in the ensemble in the order in which the selected programme chose them; this is especially important if further decisions are made with TENDENCY or several layers are to be composed.

(e) $p = 5, q = 5$

For both the number and order of the groups, the programme follows indications with which the composer supplied selection programme SEQUENCE.

order	14	SEQ-INSTR	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i) 6 (d,a1,a2,z1,z2,...)

After the ensemble has been formed, the score is assembled according to the selection principle for SEQ-INSTR. The chosen selection principle extends to the entire ensemble if s = 0, per layer to the next group respectively if s = 1 or s = 2.

If selection programme TENDENCY is called for the "order", the mask's edges are calculated per time-point. If surprises are to be avoided, either a large mask should be provided so that a choice of several instruments is possible in the chord; or the instruments should play entire chords; or to be quite safe vertical density should depend on the instruments. Compare remark on "scoring" in AMB-CHORD.

8.2 HARMONY

For the computation of the harmony, there are three harmonic principles, named CHORD, ROW and INTERVAL. The last two are occasionally both placed under the heading of "row principles". Although only one of these principles may be used in a variant group, data must still be given for all of them. Dummies which can be used for this can be seen in the APPENDIX.

Basically, if HARMONY is main parameter and use is made of the CHORD principle, the vertical density is given together with the selected chord; in this case the vertical density can therefore neither be autonomous nor depend on the instrument. The distribution of the instruments among given chords (the "scoring") obeys the rules already quoted (8.1, commentary to AMB-CHORD; see also 9.2). Vice versa, no use can be made of the CHORD principle if HARMONY is not the main parameter. However, these restrictions do not apply to harmonic principles ROW and INTERVAL.

For HARMONY, only relative pitches are given, including the 0 for percussion instruments; the 0 then keeps a place free, as it were, for a percussion instrument if HARMONY comes before INSTRUMENT.

In the commentary to survey 8-3, the entries for HARMONY are dealt with singly. For purposes of orientation, a line from the survey heads each explanation.

CALL	CALL NUMBER	CODE	FORMAT
principle	15	HARM	1 *(CHORD)* 2 *(ROW)* 3 *(INTERVAL)*
CHORD table	16	TAB-CHORD	t,t,...,t t,t,...,t ...
sequence of chords	17	SEQ-CHORD	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i)
transposition of chords	18	TRANSP-CHORD	0 *(no transposition)* 1 *(transposition with ALEA)* 2 *(transposition with SERIES)* 3 (t,t,...,t) *(transposition with given row)*
ROW	19	ROW	t,t,...,t
transposition of row	20	TRANSP-ROW	0 *(no transposition)* 1 *(transposition with ALEA)* 2 *(transposition with SERIES)* 3 *(chromatic transposition)* 4 *("serial" transposition)*
INTERVAL table	21	TAB-INT	matrix 2 *(chord instead of matrix)*
chord	22	CHORD-INT	t,t,...,t 0 *(matrix instead of chord)*
forbidden tones	23	XCL-FRQ	t,t,...,t 0 *(no forbidden tones)*
inversion	24	BIT	1 *(inversion of matrix)* 0 *(no inversion)*

<u>Fig. 8-3</u>

Survey of HARMONY parameter

principle	15	HARM	1 *(CHORD)* 2 *(ROW)* 3 *(INTERVAL)*

The required principle is selected by call numbers 1, 2 and 3. Although indications are only necessary for the chosen principle, they must be made for all entries (by means of dummies if necessary).

CHORD table	16	TAB-CHORD	t,t,...,t t,t,...,t ...

Any number of chords can be given, one chord per group. Maximum number of tones per group = pitch grid tr (compare 9.3). Each t can assume all values between 0 and tr inclusive (0 = percussion instrument). Chords starting with 0 are excluded from transposition, regardless of the numbers which follow. Chords only consisting of 0's are allowed.

sequence of chords	17	SEQ-CHORD	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i)

The values for i (for SEQUENCE) may not be greater than the number of chords which have been given; the number of i's may not exceed the number of chords either.

transposition of chords	18	TRANSP-CHORD	0 *(no transposition)* 1 *(transposition with ALEA)* 2 *(transposition with SERIES)* 3 (t,t,...,t) *(transposition with given row)*

As soon as a chord has been used, it is transposed and inserted in the table again. The starting tones of the chords given by the composer are stored separately and used as reference tones for transposition. Transposition is carried out by means of a transposition interval made available by ALEA, SERIES or a tone-row given by the composer. The chord-tones in the form prescribed by the composer are raised by the transposition interval modulo tr (according to the reference tones). Zeros (= no pitch) are ignored. ALEA and SERIES provide transposition intervals between 1 and tr (transposition by tr is the same as transposition by 0, thus resulting in the original pitch); the composer's transposition row may also only contain relative pitches between 1 and tr inclusive. Example 8-4 shows some examples of transposition.

<u>Example 8-4</u> (for tr = 12)

chord table: *1 2 4*
11 10 0
5 6 7 8 9
12 5 6 0
0 5 6 10

order with SERIES: *3 5 4 2 1 4 1 3 2 5 ...*

transposition with SERIES: *3 8 5 7 2 6 10 12 ...*

group index	*chord (original version)*	*chord (current position) → SCORE*	*transposition interval*	*transposition → TABLE*
3	5 6 7 8 9	5 6 7 8 9	3	8 9 10 11 12
5	0 5 6 10	0 5 6 10	-	0 5 6 10
4	12 5 6 0	12 5 6 0	8	8 1 2 0
2	11 10 0	11 10 0	5	4 3 0
1	1 2 4	1 2 4	7	8 9 11
4	12 5 6 0	8 1 2 0	2	10 3 4 0
1	1 2 4	8 9 11	6	2 3 5
3	5 6 7 8 9	8 9 10 11 12	10	6 7 8 9 10
2	11 10 0	4 3 0	12	4 3 0
5	0 5 6 10	0 5 6 10	-	0 5 6 10

The first column contains the group index in the order in which the chords were selected by SERIES. For checking purposes the column contains the original version of the chords, which are however gradually replaced by transpositions. The current (transposed) chord position can be seen in the third column. This is the form in which the chords are inserted in the score. The fourth column shows the assumed transposition intervals. The fifth column, lastly, shows the transpositions in which the chords are returned to the table. This transposition is identical with the current position (third column) if the relevant chord is chosen again for the score. - Note that a transposition interval is not selected for chords beginning with the 0-pitch.

In example 8-4 several chords start with the same tone (8); this might have been avoided if all chords in the table had started with 1. True, all chords would have started with the same tone in the first pass, but the transposition principle (e.g. SERIES or a given row) would have been clearly expressed in the starting-tones. Starting-tones, however, are of slight importance, since the decision as to the octave register of the individual tones in the chord is not made until REGISTER.

When transposition intervals are used only once in a particular arrangement, we call this a "transposition cycle" (analogous to the "selection cycle"). In ALEA this is of infinite length, in SERIES it is limited to tr tones, and in the case of a given row its length is equal to the number of tones in the row. Each transposition cycle is started once for each variant group; in the variant group each new variant simply continues the series of chords and the row of transposition intervals. If this is not desirable, a new variant group must be started for each variant (number of variants = 1); see 9.7 and section 10.

The same applies if several layers have their own harmony; the transposition cycle, like the selection cycle, passes over the layers and can only make a fresh start with a new variant group.

ROW	19	ROW	t,t,...,t

Any number of relative pitches between 0 and tr inclusive can be given. If INSTRUMENT comes after HARMONY a percussion instrument is inserted for pitch 0. Pitch 0 is ignored, however, if INSTRUMENT comes first and a melody instrument has been selected. The tones in the row are distributed in turn among the entry points, and per entry point among the chord tones ("vertical density").

transposition of row	20	TRANSP-ROW	0 *(no transposition)* 1 *(transposition with ALEA)* 2 *(transposition with SERIES)* 3 *(chromatic transposition)* 4 *("serial" transposition)*

Once the row as a whole has been used, it is transposed. See TRANSP-CHORD for reference tone and transposition cycle.

The development of the row (including transpositions) is not broken off at each variant; a new row only starts again with any certainty together with a new variant group. ALEA and SERIES provide transposition intervals between 1 and tr, call 3 causes the ascending sequence of numbers from 1 to tr, for "serial" transposition the given row acts as a row of transposition intervals, call 0 always means that no transposition takes place. Remember that all 0-tones in a row are excluded from transposition.

INTERVAL table	21	TAB-INT	matrix 2 *(chord instead of matrix)*

The interval principle produces an "infinite" row, the tones of which are distributed one for one among the entry points and chord tones. Although repetition of tones is postponed for as long as possible, there is no certainty that complete rows of tr tones will result. The order of the tones is determined according to a matrix containing for each of the possible intervals (minor second to major seventh if tr=12) allowed and forbidden succeeding intervals; forbidden succeeding intervals are indicated by 1, allowed ones by 0. Take care that if a line (succeeding intervals) only contains forbidden intervals, the column with the same index (given intervals) is also blocked in its entirety (fig. 8-5, assuming tr=12); in other words take care to make each allowed succeeding interval, when turned into a given interval, continue the row. Since interval 5 (5th line) has no continuation here, the same interval must be forbidden for all given intervals (column 5).

Fig. 8-5

	1	2	3	4	5	6	7	8	9	10	11
1					1						
2					1						
3					1						
4					1						
5	1	1	1	1	1	1	1	1	1	1	1
6					1						
7					1						
8					1						
9					1						
10					1						
11					1						

At the beginning of the selection cycle any pitch (with the exception of the forbidden tones, compare XCL-FRQ) is selected by ALEA, and then any interval is determined where at least one 0 occurs in its line. This constitutes a first "given" interval. The second interval is found from among the allowed succeeding intervals, resulting in the third tone. The second interval is now a given interval for which the matrix contains one or more allowed succeeding intervals. If an allowed succeeding interval results in a forbidden tone, another succeeding interval is sought; if this is not possible, a "wrong" tone is printed, together with a comment.

Whether or not a new tone may be inserted in the score depends on three conditions:

(1) it may not be one of the forbidden tones,

(2) it must fit in the register if REGISTER comes first, or the instrument must be able to play it if INSTRUMENT comes first (or both),

(3) it can be rejected if repeated prematurely in the row.

If no allowed tone can be found on these terms, they are ignored in reverse order, so that at worst only the conditions of the matrix and forbidden tones still apply. Infringements of conditions in REGISTER and INSTRUMENT are accompanied by a comment.

The interval principle only produces tones between 1 and tr inclusive, which excludes percussion instruments if HARMONY comes first. If however INSTRUMENT comes first, the interval principle is skipped for each percussion instrument, so that the remaining pitches appear chronologically in the order produced by the matrix conditions. The interval principle is started pro variant or pro layer. If a chord is given instead of a matrix (see CHORD-INT), we must have TAB-INT = 2.

chord	22	CHORD-INT	t,t,...,t 0 *(matrix instead of chord)*

If a matrix is given, we must have CHORD-INT = 0, otherwise a chord of relative pitches (1 - tr) can be given at that place and analyzed according to its interval content. This results in the corresponding matrix. When the interval content is calculated, every possible interval in the chord is compared with the other tones, as is shown in example 8-6.

forbidden tones	23	XCL-FRQ	t,t,...,t 0 *(no forbidden tones)*

The relative pitches (1 - not more than tr) stated for this entry are forbidden, so that they are excluded from the "infinite row". Care must be taken that tones are only forbidden if there is at least one alternative for them in the matrix. When the first tone of the infinite row is selected, forbidden tones are of course not taken into account. If no tones are forbidden, we must have XCL-FRQ = 0.

inversion	24	BIT	1 *(inversion of matrix)* 0 *(no inversion)*

Call number 1 causes the matrix to be inverted, which means that all forbidden intervals become allowed, and vice versa. If the matrix is not to change, we must have BIT = 0.

The forbidden tones in entry XCL-FRQ are not affected by this because the matrix only calculates intervals; the tones resulting from the intervals are compared with the list of forbidden tones.

Example 8-6 (assuming that tr = 12)

Given chord: 5 6 10

Analysis:

given interval	*allowed succeeding interval*
5/6 = 1	6/10 = 4
5/10 = 5	10/6 = 8
6/10 = 4	10/5 = 7
6/5 = 11	5/10 = 5
10/5 = 7	5/6 = 1
10/6 = 8	6/5 = 11

Matrix

1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1

In this example, which only contains different intervals, there is only one possible succeeding interval for each given interval; this means that the matrix can only produce one single chord (a transposition of the given chord), which is repeated continuously. The harmonic scope is wider if the given chord consists of more tones, or at least contains interval repetitions (examples: 3 4 5, 4 6 8 or 2 5 11).

9 ADDITIONAL DATA

The additional data serve purposes not provided for by the parameter programmes, but which regulate the interplay of the parameters and perform auxiliary functions. In this category are the organization of the time structure, control over vertical density, the hierarchy of the parameters and other purposes to be discussed in this section. Since the composer must provide data for these operations, these data are explained in the light of the various programme entries.

9.1 TIME STRUCTURE

The time structure is governed by the parameters ENTRY DELAY (7.2), DURATION (7.3) and REST (7.4), supplemented by data on structure duration and tempo (9.1.3, 9.1.4). It is important to visualize the interaction of all the time quantities in PR-2. The "entry point" and the "average entry delay" play a significant part.

9.1.1 Entry points

The duration of a structure variant is divided up into entry points, which result from the continuous addition of the entry delays. Each entry point is characterized by the entry of a tone or chord; to keep things simple we shall speak of the entry of a chord and at the same time indicate the number of tones in the chord (which may be a minimum of 1 or a maximum of tr). Percussive sounds of indeterminate pitch also count as chord tones of course.

Every entry point is given one or more durations: one duration if it is a single tone or a chord with equal durations, several durations if it is a chord with independent durations. We might say that a duration is selected either per tone or per entry point. The end point of a tone accordingly results from the sum of entry point and duration.

Rests ("pseudo rests") occur automatically if the duration is shorter than the entry delay; duration and entry delay of a tone or chord are calculated starting from the same time point (compare 7-11). If a previous tone is still sustained the "pseudo rest" is "concealed". Additional "autonomous" rests inserted because of the REST parameter modify the sequence of entry points (durations are unaffected). All entry points following the entry point of an autonomous rest are increased by the duration of the rest.

Because of the fact that the durations in a chord can be independent of one another, we have a double definition of the chord with more than one tone:

(a) chord tones begin and end simultaneously,

(b) chord tones begin simultaneously and end one after the other.

It is not possible to have chord tones starting one after the other.
Auxiliary constructions:

- "arpeggio" indication,
- single tones with short entry delays and longer durations; however, it is not possible in this case to have the chord tones ending simultaneously.

Simultaneously starting tones also result from entry delay 0 ("pseudo chords").

9.1.2 Average entry delay

In order to achieve a previously established structure duration it would be obvious to calculate all the parameter values for each entry point and to go on doing this until the sum of the entry points was about equal to the structure duration. However, this method must be eliminated for two reasons: (1) for selection programme TENDENCY the number of entry points must be known in advance because the step-for-step movement of the mask could otherwise not be calculated; (2) the present computer configuration at Utrecht makes it impossible either to accommodate the entire programme in core storage or to transfer the parameter subprogrammes per entry point from background storage to core storage.

The individual parameters are therefore calculated one after the other in the selected hierarchy. This method necessitates establishing the number of entry points (which is the same in all parameters) in advance. This is done by means of the average entry delay. During the first pass of the programme, the ensembles are put together for all the parameters. The average entry delay is then calculated from the ensemble of entry delays

- for ALEA, GROUP and TENDENCY without modification,
- for RATIO taking the ratio factors into account (each ensemble

element p times, divided by the sum of all p's); list elements not in the ensemble are of course ignored; repeated list elements are calculated a corresponding number of times,

- for SEQUENCE only for the elements explicitly mentioned.

Example 9-1 shows the calculation of the average entry delay for a given ensemble.

The selected structure duration is divided by the average entry delay, which results in the number of entry points. This calculation is independent of the hierarchy of each structure variant. When the ENTRY DELAY is calculated at its appointed time, an entry delay is selected for every entry point according to the chosen selection programme, regardless of whether the sum of entry delays coincides with the structure duration or not. Experience shows that the deviations from the required structure duration are relatively small. This especially applies to a short selection cycle with relation to the number of entry points and to all selection principles with the exception of TENDENCY; it is a precondition of this principle that all ensemble elements have more or less the same chance. It is therefore recommended when using TENDENCY to assemble only elements in the ensemble which are actually to be used. But even then greater deviations from the desired structure duration can be expected if chiefly small or large time quantities are selected for TENDENCY because of the mask's movements. Additional deviationsfrom the desired structure duration also arise of course because of longer durations, since the effective structure duration = last entry point + last duration. If the structure duration is not to be essentially exceeded or fallen short of, it is recommended to have a test version calculated first and then to correct the structure duration accordingly.

Autonomous rests are basically ignored when the number of entry points is being calculated; this means that the actual structure duration is increased by the sum of all inserted rests.

Example 9-1

LIST 0.2, 0.4, 0.6, 0.8, 1.1, 1.5, 2.0

ENSEMBLE 2 2 2 4 4 5

(a) *order with ALEA, GROUP or TENDENCY, average entry delay =*

$$\frac{0.4 + 0.4 + 0.4 + 0.8 + 0.8 + 1.1}{6} = 0.65$$

(b) *order with RATIO, ratio factors (referring to list):*
3 3 3 2 2 2 1 . Average entry delay =

$$\frac{0.4\times3 + 0.4\times3 + 0.4\times3 + 0.8\times2 + 0.8\times2 + 1.1\times2}{15} = 0.6$$

(c) *order with SEQUENCE (1,3,5), average entry delay =*

$$\frac{0.4 + 0.4 + 0.8}{3} = 0.53$$

9.1.3 Structure duration

CALL	CALL NUMBER	CODE	FORMAT
list	51	L-DUR-VAR	r,r,...,r
order	52	SEQ-DUR-VAR	1 *(order with ALEA)* 2 *(order with SERIES)* 5 (i,i,...,i) *(order with SEQUENCE, number of i's ≤ number of structure durations)*

Fig. 9-2

Survey of STRUCTURE DURATION

The structure duration is given in seconds. A list contains any number of structure durations; the order of structure durations can be determined with three selection programmes. A new structure duration is selected per variant. The selection cycle runs through all variants of a variant group and is repeated if necessary.

9.1.4 Tempo

CALL	CALL NUMBER	CODE	FORMAT
list	49	L-TEMP	r,r,...,r
order	50	SEQ-TEMP	1 *(order with ALEA)* 2 *(order with SERIES)* 5 (i,i,...,i) *(order with SEQUENCE, number of i's ≤ number of tempi)*

Fig. 9-3

Survey of TEMPO

The metronome tempo is defined as the number of metres per minute (decimals permitted); we shall therefore regard the metre as a time-unit (= beat). The tempo is irrelevant for the computation of the time structure in seconds. The tempo is, however, taken into account when the parts are printed in metrical notation (see section 11). A list contains any number of metronome indications; the order of the tempi can be defined by means of three selection programmes.

The selection cycle runs through all variants of a variant group and is repeated if necessary.

In the printout of the parts the rhythmic context is subdivided into metres which the composer can arrange as bars in any way he pleases. The duration of a metre = tempo/60. All metres of a structure variant are of equal length. Additional metronome indications given perhaps in PERFORMANCE have no influence on the calculation of the duration of the metre.

9.2 VERTICAL DENSITY

The number of tones to start simultaneously depends on one or more of the following conditions:

(1) number of 0-entry delays in succession;

(2) minimum and maximum number of tones which an instrument can play simultaneously, and the number of selected tones (or tones remaining in the chord) within this range;

(3) number of tones in the chord - selected according to the CHORD principle;

(4) minimum and maximum number of tones for the "vertical density".

Apart from the first one, these conditions have already been explained in 5.2. Fig. 9-4 shows a survey of the vertical density.

The vertical density is only taken into account if the ROW or INTERVAL principle is selected for the HARMONY parameter. If, on the other hand, the CHORD principle is selected, HARMONY becomes main parameter and thus decisive for the vertical density. See fig. 9-5 and examples 9-6.

Vertical density can only be computed in layers. If several layers are to be computed (with or without common harmony), all of the above conditions apply to each individual layer.

CALL	CALL NUMBER	CODE	FORMAT
vertical density	53	DENSITY	0 *(vertical density depends on instrument, INSTRUMENT is main parameter)* a,z *(minimum and maximum density, irrelevant if HARM=1)*
order of densities	54	SEQ-DENS	1 3 (p,p,...,p) 4 (a,z,type) 5 (i,i,...,i)

Fig. 9-4

Survey of VERTICAL DENSITY

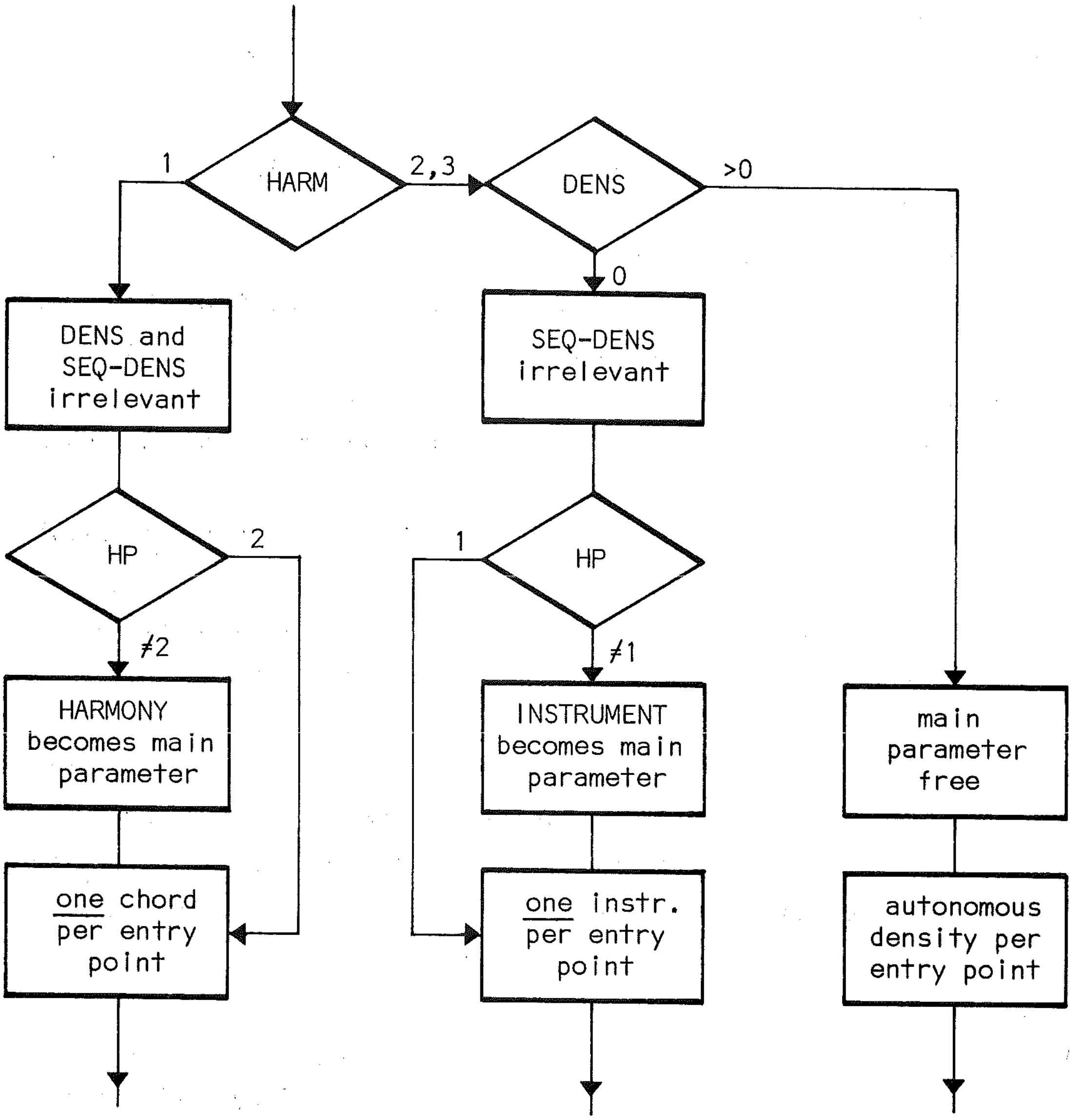


Fig. 9-5

If DENSITY = 0, the indications for SEQ-DENS are ignored. INSTRUMENT automatically becomes main parameter, even if the composer has decided otherwise. For each entry point, ALEA is used to determine the number of chord tones in the chord size of the selected instrument. The harmony can be determined by means of a row or the interval table.

If DENSITY = a,z , any parameter may be main parameter, although the CHORD principle may not be selected for HARMONY since HARMONY would then automatically become main parameter, and entries 53 and 54 would be ignored. The illustration clearly shows the contexts:

first the harmonic principle is tested, and only if ROW or INTERVAL are selected is the vertical density tested. If it depends on the instrument, INSTRUMENT is main parameter, if not, it is "autonomous". In this case, any parameter can be main parameter.

The number of possible chord sizes in the vertical density is z-a+1. The same number of ratio factors must be given to selection programme RATIO in call 54. The number of i's for SEQUENCE may not exceed the sum of z-a+1.

Example 9-6

(a) *Only one instrument is to enter at each entry point:*

HARM	2 *or* 3
DENSITY	0
SEQ-DENS	1 *(dummy)*
HIER	1 ... *(otherwise INSTRUMENT is automatically in the first place)*

The vertical density depends on the instruments selected for the score, and on how these instruments are defined according to their chord sizes (entry 11). The vertical density for each entry point can never be greater than the maximum chord size of the respective instrument.

(b) *Autonomous density:*

HARM	2 *or* 3
DENSITY	a, z
SEQ-DENS	...
HIER	... *(free)*

The vertical density is autonomous, only restricted by the numbers for a and z (entry 53). The vertical density is independent of the instruments playing a chord together.

(c) *Vertical density according to the CHORD principle:*

HARM	1
TAB-CHORD	...
SEQ-CHORD	...
TRANSP-CHORD	...
DENSITY	... *(irrelevant)*
SEQ-DENS	... *(irrelevant)*
HIER	2 ... *(otherwise HARMONY is automatically in the first place)*

The vertical density is identical to the size of the chord selected for each entry point; HARMONY/CHORD is main parameter.

Of course the actual vertical density can be much greater if durations are longer than entry delays, resulting in superpositions of chords. However, an automatic check in the programme prevents more than tr tones from ever overlapping. In examples (a) and (c) INSTRUMENT or HARMONY are main parameters, the chord selection thus takes place before the time structure was established. For the calculation of the time parameters however, suitable choice of entry delays and durations ensures that no more than tr tones overlap. The same applies to example (b) if the chords are fixed before the time structure is established. If on the

other hand the time structure is already established, the minimum density (a) is if necessary not reached when the chord is selected.
Durations for percussion instruments also count as "tones" here.

9.3 THE PITCH-TIME GRID

The pitch-time grid, like the PERFORMANCE parameter, is a later addition to PR-2. The three variables for the pitch-time grid were originally constants. Fig. 9-7 is a survey of the entries.

The pitch-time grid provides the composer with the opportunity of

(a) determining the number of tones per octave; this number is the same for all octaves,

(b) determining the maximum number of metrical subdivisions; this information is only taken into consideration for the printout of the parts, and is of no account for the score (in seconds),

(c) fixing the smallest time-value which is applicable to the subdivision of the metre (triplets, quintuplets, etc.). All subdivisions below this limit value are then ignored.

The tones within the octave are numbered; the composer is free to interpret the numbers of the tones. Since the octave numbers are integral multiples of 10^2 for the representation of "absolute" pitches, a maximum of 99 tones per octave is available.

For the printout of the parts, all time-values are converted into metric expressions. For this purpose the duration of the metre is first calculated according to the metronome tempo: dm = 60/tempo. The absolute durations of the metric subdivisions are accordingly dm/1, dm/2, dm/3, ..., dm/fr.

If we assume that not more than 10 tones per second can be played on a particular instrument, we must have $d = dm/fr \geq 0.1$. (However, tr, fr and d apply to all instruments of a structure variant.) But since dm/fr depends on the tempo, which can change in each structure variant, the smallest time-value d is not dependent on the shortest entry delay of the list L-ENTRY. Example 9-8 is an example of the calculation of the durations of the metric subdivisions.

CALL	CALL NUMBER	CODE	FORMAT
grid	63	DIVISION	tr,fr,d

Fig. 9-7

Survey of PITCH-TIME GRID

tr = number of tones per octave ($1 \leq tr \leq 99$),
fr = maximum number of metrical subdivisions,
d = smallest time-value for metrical subdivision (in seconds).

Example 9-8

Given tempo: 100
DIVISION: 12, 10, 0.08

subdivision into	*duration of metric subdivision*
1	*0.6 seconds*
2	*0.3*
3	*0.2*
4	*0.15*
5	*0.12*
6	*0.1*
7	*0.086*
8	*0.075*
9	*0.067*
10	*0.06*

Since d is 0.08 here, the subdivisions into 8, 9 and 10 are not taken into consideration.

9.4 BLOCKING

Blocking applies to INSTRUMENT, HARMONY, REGISTER and DYNAMICS, and prevents the same instruments, relative pitches, registers or intensities from being superposed. Fig. 9-9 is a survey of the data for this entry.

The four parameters mentioned can be blocked in any combination. Blocking reaches its full effect when the rhythmic context has been fixed, so that undesired "doublings" (time superpositions) can be prevented. The rhythmic context can however not be examined until ENTRY DELAY and DURATION have been fixed. It is therefore advisable to prescribe at least parameters 4 and 5 as main parameters for the hierarchy (see 9.5) if even only one parameter is to be "blocked".

Any comment there might be does not show whether the designated parameter value has disregarded the blocking instruction or other rules.

CALL	CALL NUMBER	CODE	FORMAT
blocked values	55	BLOCK	s,s,s,s *(s = 1: blocking, s = 0: no blocking, order: INSTRUMENT, HARMONY, REGISTER, DYNAMICS)*

Fig. 9-9

Survey of BLOCKING

When INSTRUMENT is blocked, an instrument can not be inserted in the score if an instrument with the same list-index is being "played".

When HARMONY is blocked, a relative pitch can not be inserted in the score if this same relative pitch is still being "played". This does not apply to relative pitch 0.

When REGISTER is blocked, a register can not be inserted in the score if one or more absolute pitches in the register with the same index are being "played".

When the DYNAMICS parameter is blocked, a dynamic value can not be inserted in the score if one with the same list-index is being "played".

9.5 HIERARCHY

The purpose of hierarchy is to establish the order in which the individual parameters are to be calculated. If the parameters are interdependent, it is only possible to choose elements for the second parameter which do not contradict those of the first parameter. The same applies to the elements of the third parameter with regard to those of the first two, and so forth. If the ensembles of all the parameters are "compatible", i.e. if every element of one parameter can be combined with all the elements of all the other parameters, hierarchy is not relevant.

So that the hierarchy can be established, each parameter has a call number, as shown in fig. 9-10. Fig. 9-11 is a survey of the necessary data.

An automatic check ensures that the REST parameter is always last in the hierarchy (but compare the restrictions), and that the hierarchy is corrected if necessary. These corrections are made if

- the CHORD principle is used in HARMONY: HARMONY is main parameter;
- the vertical density has to depend on the instrument (compare 9.2): INSTRUMENT is main parameter;
- several layers with the same harmony have to be calculated: REST is last parameter but one, HARMONY comes last;
- the mode of performance has to change at each rest: PERFORMANCE is last parameter but one, REST comes last;
- the mode of performance changes at each rest <u>and</u> several layers have to be calculated with common harmony: HARMONY is last parameter, PERFORMANCE is last but one, with REST just before it.

Any corrections are made so that the relevant parameter is put in its correct place and the other parameters are shifted to the right or left as necessary.

<u>Example</u>: *HIERARCHY:* *5 4 6 2 7 1 3 8*

(HARMONY according to CHORD principle, which is why there must be a correction): *2 5 4 6 7 1 3 8*

Fig. 9-12 shows in table form the eight cases in which the hierarchy is corrected if necessary.

call number	parameter
1	INSTRUMENT
2	HARMONY
3	REGISTER
4	ENTRY DELAY
5	DURATION
6	DYNAMICS
7	PERFORMANCE
8	REST

Fig. 9-10

Parameters 1 to 7 can be main parameters, REST can not. These call numbers are not identical with the order in which the parameters are named in the data form (APPEMDIX).

CALL	CALL NUMBER	CODE	FORMAT
hierarchy	56	HIER	0 *(hierarchy with SERIES)* i,i,...,i *(hierarchy with SEQUENCE, unstated parameters with SERIES, 1 ≤ i ≤ 7)*

Fig. 9-11

Survey of HIERARCHY

HIER = 0 means that the order of parameters 1 to 7 is determined by SERIES. Parameter 8 (REST) always comes last.

Otherwise the composer can determine the order of the parameters himself. In this case it is not necessary to state all the parameters; the unstated ones are automatically added by SERIES.

	place in hierarchy	1	2	3	4	5	6	7	8
1	no further restrictions								P
2	vertical density depends on instr.	I							P
3	harmony with chords	H							P
4	performance per rest							P	M
5	layers with common harmony							P	H
6	combination of (5) and (4)						P	M	H
7	combination of (2) and (4)	I						P	M
8	combination of (3) and (4)	H						P	M

Fig. 9-12

RESTRICTIONS

9.6 COMBINATION

The purpose of combination is to link the instrument groups selected in the INSTRUMENT parameter with similar groups (groups with the same indices) of other parameters. The selection of the groups is in this case only made for INSTRUMENT, the selected index order being automatically adopted for the combined parameters. The selection principles assigned to the combined parameters are not taken into account. For the non-combined parameters, a group is assigned to the respective ensemble according to the respective selection principle. The tables of combined parameters must always contain the same number of groups; the tables of non-combined parameters are free. Compare section 6.1. Fig. 9-13 is the survey.

The rules in the commentary to fig. 9-13 can place the composer in an awkward position if he has to define dummy groups for combined parameters. Let us assume that the composer has prepared lists and tables for all variant groups, and that he intends to activate the combination for a particular variant group. Perhaps he has declared 6 instrument groups, but only needs the first three for the combination. However, the rules mean that he must declare 6 groups for each of the parameters to be combined, although only the first three participate in the combination. If he will be needing 6 groups anyhow, the situation is simple: only the groups corresponding to INSTRUMENT will be used for the combination. Otherwise he must declare dummy groups, perhaps like this:

```
TABLE:  i,i,...,i
        i,i,...,i
        i,i,...,i
        1
        1
        1
```

The disadvantage of this way is that the composer can

first — only use selection principle SEQUENCE for INSTRUMENT during combination, in order to avoid groups 4, 5 and 6 being quoted; and

secondly — after de-activating the combination in the previously combined parameters, he has to deal with dummy groups which in their turn can only be manipulated by SEQUENCE.

In view of these consequences, special tables should be declared for variant groups with combination, tables containing no dummies. A

CALL	CALL NUMBER	CODE	FORMAT
combination	4	COMB	0 *(no combination)* i,i,...,i *(combination of stated parameters with INSTRUMENT; 3 ≤ i ≤ 8)*

Fig. 9-13

Survey of COMBINATION

If COMB = 0, no combination occurs. For INSTRUMENT, as many groups are assembled to form an ensemble as indicated in N-GR-INSTR (call 13). For all other parameters (including REST) only one group per variant is used.

Otherwise the parameters mentioned explicitly are combined with INSTRUMENT. A maximum of 6 parameters may be named (parameters 3 to 8). For the parameters stated, the ensemble is formed from groups with the indices which have been stated or selected for INSTRUMENT in SEQ-GR-INSTR (call 13). For the unstated parameters, only one group per variant (and per layer if UNION = 1 or UNION = 2 in call 13) is used (as in the case of COMB = 0).

The groups forming the ensemble are treated as one single group if UNION = 0. If UNION = 1 or UNION = 2, the groups are separated; one layer per group is calculated.

> IMPORTANT: With combination the number of groups in the tables of parameters to be combined must be the same as the number of groups in the INSTRUMENT parameter, although this number is free in the parameters which are not to be combined.

special consideration should be the purpose for which the combination is to be activated; this purpose is particularly unambiguous if several layers are to be composed so that a new group is employed for each layer (compare section 6).

9.7 RANDOM GENERATOR

CALL	CALL NUMBER	CODE	FORMAT
start number	5	START	r $(0 \leq r < 1)$

Fig. 9-14

Survey RANDOM GENERATOR

All random decisions in PR-2 are made by a RANDOM programme which generates equally distributed numbers between 0 and 1. This random generator is called by selection programmes ALEA, SERIES, RATIO and GROUP, and also in the mask of TENDENCY. The index which makes an arbitrary selection among given elements is calculated by

$$i := entier\ ((z-a+1) \times RANDOM + a)$$

a and z being the limits of the range. The number sequence produced by RANDOM is reproducible; this makes it possible for a mainly random programme run to be repeated literally.

The start number, which must be less than 1, has no predictable influence on the further course of the programme. It can therefore also be used for the classification of various tests or several variant groups (e.g. 0.1, 0.11, 0.12, 0.2, etc.).

The random generator must be provided with a start number at least once, in the first set of data. This start number is automatically re-inserted for each variant group if the composer does not provide a new start number. It is therefore advisable to provide a new start number for each variant group in order to avoid identical decisions. This is especially important if the next variant group is calculated for a structure formula which barely differs from the previous one.

9.8 NUMBER OF VARIANTS

CALL	CALL NUMBER	CODE	FORMAT
number of variants	3	N-VARIANTS	n *($n \geq 1$)*

Fig. 9-15

Survey of NUMBER OF VARIANTS

For every set of input data, the number of variants to be calculated according to the structure formula must be stated. If the variants themselves consist of several layers, these layers do not count as variants, although they are treated as such in terms of programme technique (see section 10).

Example: *number of variants: 4*
number of groups for layer composition: 3
result: 4 variants, each with 3 layers.

Note that all selection principles used per variant (except tempo and structure duration) are called 12 times in this case.

9.9 PROGRAMME SPECIFICATION

CALL	CALL NUMBER	CODE	FORMAT
user specification	1 2	USER CHIFFRE	w w

Fig. 9-16

Survey PROGRAMME SPECIFICATION

Each set of input data must contain information about the composer, at least his name. As well as this, the composer can provide each individual variant group with an explanatory commentary. The following symbols can be used for these data:

- the capitals and minuscules of the alphabet,
- the digits from 0 to 9,
- specials signs such as . : - + / × = ? " ( ) < > .

The comma (,) and semicolon (;) may not be used.

C INPUT AND OUTPUT

10 THE DATA FORM

The data form consists of the data for 63 programme entries. The diagrammatic surveys of entries 1 to 56 and 58 to 63 were discussed in sections 7 to 9 (see section 11 for entry 57). A simplified version of the data form appears as the APPENDIX.

The data for the 63 entries are also known as input data. PR-2 uses the input data to calculate a variant group consisting of as many variants as the composer requires as N-VARIANTS (entry 3). This means that the composer's data cannot be altered within a variant group.

The data provided by the composer for a variant group are called a "set of data". This set of data must be "complete" for the first variant group, i.e. it must contain complete data for all 63 entries ("initial set"). For further variant groups the composer only needs to register data for the entries which have to be changed with regard to the previous variant group ("subsequent set"). Unrevised entries are automatically taken over from the last variant group, so that the computer always receives a complete set of data.

Entries of no significance for the initial set can be provided with dummies (this also applies to subsequent sets). Care must be taken here that the dummies also agree with the respective format. We recommend the dash (-) as dummy for alphanumeric information.

If a table is to contain more than one group, the call number must be repeated for each group. The following entries are to be treated as tables:

12	instruments
13	instrument groups and union
16	chords
26	registers
30	entry delays
34	durations
40	rests
45	intensities
59	modes of performance

Note that entry 13 also counts as a table (always three groups for p, q and s), entries 7 to 11 being treated as lists.

It is sufficient to state the call number for each entry (repeat for each group in the case of tables); the code text is superfluous. The data are separated for each entry (or group of tables) by a semicolon (;); list elements are separated by commas (,). All decimal numbers can have up to 12 decimal places, only 2 of which, however, are printed in the output. Non-English readers note that a decimal point must always be written, since commas are only for the distinction of elements. Wherever selection programmes are called, it is advisable to put round brackets around the actual parameters. See example 10-1.

The input data finish - in the call number column - with a zero (0) if a further set of data is to come, and with -1 if no further set of data is to come.

Example:

```
1   .....;
.   .....;
.   .....;
63  .....;
0
.   .....;
.   .....;
0
.   .....;
.   .....;
-1
```

The entries must be stated in ascending sequence for each set of data.

Since subsequent sets can only be given for entire variant groups, it is important to make a distinction between data which can be used per variant and those used per variant group. Fig. 10-2 is a table showing which data are to be taken into account per variant group, per variant and per time-point (or tone). It does not matter whether the data are re-defined or adopted automatically from the previous set of data.

The table shows clearly that lists and tables are basically taken into account per variant group; the selection of the groups (the formation of groups into an ensemble) occurs per variant, the order being established per time-point, per tone or according to other view-points.

Example 10-1

(a) alphanumeric information:

2 string trio, test 30, 6.8.1970 / structure with transitions;
44 pp,pp<,mf,ff>,fff;

(b) lists:

11 1, 1, 1, 3, 3, 7;

29 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1.2, 1.5, 1.8, 2.2, 2.7, 3.5;

(c) tables:

34 1;
34 1-6;
34 4, 7;

13 5 (1);
13 2;
13 0;

(d) selection programmes:

28 6 (100, 10, 60, 90, 100);

32 4 (3, 8, 4);

Repeated elements need only be stated once and can be provided with the appropriate factor.

Repeated pairs of elements can be placed in square brackets; the repetition factor then comes straight after the bracket. (But this does not apply to groups of 3 or more elements).

Continuous sequences of integers, either increasing or decreasing, can be indicated by a dash.

Examples:

for repeated elements: *5×8* (instead of 5,5,5,5,5,5,5,5),
for pairs of elements: *[301,307]3* (instead of 301,307,301,307,301,307),
for increasing sequences: *1-8* (instead of 1,2,3,4,5,6,7,8),
for decreasing sequences: *33-27* (instead of 33,32,31,30,29,28,27).

However, the composer must be clear about the difference between elements which can be given per variant group (lists, tables and also general information as in the first five entries) and data for the selection principles. Fig. 10-2 makes no distinction here. For the number of groups (entry 13p) a selection principle is determined per variant group, but this principle is in action per variant. It is unimportant whether one or more layers are to be calculated; the number of groups applies to the entire variant and is automatically adopted for all combined parameters. The same applies to the selection of groups (entry 13q). The group selection in REGISTER, ENTRY DELAY, DURATION, REST, DYNAMICS and PERFORMANCE only occurs "per layer" however if these parameters are "uncombined"; otherwise the group selection is made per variant - in INSTRUMENT.

Neither does fig. 10-2 tell us anything about how HARMONY is treated. Elements and other data are given per variant group; but whether the selection of chords, row tones or interval tones occurs per variant or per layer depends on the union (entry 13s).

Technically, the data are fed into the computer in such a manner that first of all the composer's text is transferred literally onto punch tape. A special preliminary programme puts this set of data into a new form as required by the read programme of PR-2. (The difference between composer's data and the preliminary programme's version is mainly in counting operations which might be a nuisance to the composer, and also in the consideration of extremely exact format instructions which might lead to errors if the tape were punched by hand.) The computer in its turn makes a punch tape of the new version, and this tape is the input material for PR-2. Since all sets of data (initial sets, containing the data for all 63 entries and all subsequent sets for re-defined entries) are dealt with in one run by means of the preliminary programme, the computer functions as a memory for data which are to be taken over for further variant groups. The new punch tapes with complete sets of data for all 63 entries can then be presented to the PR-2 programme independently of one another and at any time. This system can easily be adapted if disc or tape storage are to be used as intermediaries.

entry	description	per variant group	x = per variant xx = per layer	within a variant per
1	user	x		
2	specification	x		
3	number of variants	x		
4	combination	x		
5	start number	x		
	INSTRUMENT			
6	list	x		
7	modes of performance	x		
8	pitches	x		
9	durations	x		
10	intensities	x		
11	chord sizes	x		
12	table	x		
13	number of groups		x	
13	selection of groups		x	
13	union	x		
14	order			entry point
	HARMONY			
15	principle	x		
	CHORDS			
16	table	x		
17	order			entry point
18	transposition			entry point
19	ROW	x		
20	transposition			row
	INTERVALS			
21	matrix	x		
22	chord	x		
23	forbidden tones	x		
24	inversion	x		
	REGISTER			
25	list	x		
26	table	x		
27	selection of groups		xx	
28	order			tone

entry	description	per variant group	x = per variant xx = per layer	within a variant per
	ENTRY DELAY			
29	list	x		
30	table	x		
31	selection of groups		xx	
32	order			entry point
	DURATION			
33	list	x		
34	table	x		
35	selection of groups		xx	
36	order			entry point, tone
37	relation	x		
38	mode	x		
	REST			
39	list	x		
40	table	x		
41	selection of groups		xx	
42	order			rest
43	mode	x		
	DYNAMICS			
44	list	x		
45	table	x		
46	selection of groups		xx	
47	order			entry point, tone
48	mode	x		
49	TEMPO-list	x		
50	order		x	
51	STRUCTURE DURATION-list	x		
52	order		x	
53	VERTICAL DENSITY	x		
54	order			entry point
55	blocking	x		
56	hierarchy		xx	
57	output	x		
	MODE OF PERFORMANCE			
58	list	x		
59	table	x		
60	selection of groups		xx	
61	order			entry point, tone, rest
62	mode	x		
63	grid	x		

Fig. 10-2

As already mentioned above, each set of data finishes with a 0 if more sets are to follow. As soon as a -1 appears at the end of a set of data, the computer is no longer able to take in any more composer's data. If the composer should wish to continue his work on the same project at a later date, he would have to write an initial set again. If such a continuation is planned (perhaps because the computer output has to be transcribed before further decisions can be made), and if the continuation is to refer to the last set of data, -2 can be written at the end of the last set of data instead of -1. PR-2 then produces a transitional punch tape which will later enable the preliminary programme to accept a subsequent set referring to the last set of data before the interruption.

11 OUTPUT

The data computed by PR-2 are printed by a line printer in the form of tables (text and numbers). The composer is therefore forced to transcribe these tables into legible notation (traditional or musical graphics). Although the organization of the score tables and parts offer the composer a maximum of convenience, traditional notation (or even musical graphics with composer-defined symbols) would be preferable.

With the cooperation of the computer centre at Utrecht University, a PLOT programme was developed at the Institute of Sonology for a simplified form of notation. This plot programme has not been used yet, since not enough computing time is available. So as to manage with less computing time, the plot programme has been adapted for the line printer. However, this notation form did not turn out to be expedient: on the one hand the space between the symbols is too large so that there is only room for a little amount of text on one page, and on the other hand the composer still has to copy the printout.

For these reasons the line printer programme has been removed from PR-2. Both notation programmes can however be reactivated for special purposes.

Finally, the score data can be requested in a punch tape version, which however is intended as input material for PROJECT 3 (in preparation) and therefore of little use as yet. We shall not deal with the notation and punch tape versions (call numbers 3 and 4) here; they only appear in fig. 11-1 for the sake of completeness.

CALL	CALL NUMBER	CODE	FORMAT
output	57	LAYOUT	1 *(score)* 2 *(score and parts)* 3 *(score and parts, also score in simplified notation)* 4 *(punch tape with score data)*

Fig. 11-1

Survey OUTPUT

Call number 1 causes not only the composer's data and check data to be printed out but also the complete score, layer for layer, and chronologically in seconds. The score printout serves as a reference for the printout of the parts (if required), and also as a model for a musical graphic to scale.

Call number 2 causes not only the score but also all parts comprising it (i.e. instruments) to be printed out singly and in metric notation. By a metre we mean a unit fixed arbitrarily by the composer (e.g. a quarter-note) which can be integrally subdivided. The given metronome tempo results in the duration of the metre: METRE = 60/TEMPO. The metres are numbered in chronological order, the metre counter being printed.

The subdivision of the metre into duplets, triplets, etc., occurs automatically. For this purpose each original time-value is compared with the possible subdivisions of a metre (also in seconds) and rounded off to the nearest subdivision point. The number of subdivisions taken into account here is determined by the composer in entry 63. The subdivision to be found per instrument and per metre is the one which occurs most frequently; if various subdivisions occur equally often, the finer subdivision is selected because the rounding-off error factor is less. All beginnings and ends of tones are involved in this calculation.

The result is a uniform subdivision for a given metre to which both entries and ends of the given instrument can be rounded off. If an instrument (e.g. piano) be notated "polymetrically", it must be defined a corresponding number of times in the instrument list (entry 6), or mentioned once in the list and a corresponding number of times in a table group.

The composer determines the finest subdivision (e.g. eight subdivisions for 8 thirty-second notes to a quarter-note), and also the shortest period in seconds which is still to have an effect in the form of a subdivision. This shortest period determines the finest subdivision (compare 9.3).

11.1 COMPOSER'S DATA

The output starts with the printout of the composer's data in 5 columns:

(1) asterisk for "new" data; not printed before data automatically taken over from the last set of data;

(2) call number of respective call;

(3) code text of respective entry;

(4) data count, of no importance to the composer;

(5) data; the numeric material appears in 10 columns to two decimal places exactly.

11.2 CHECK DATA I

In order to adapt PR-2 to the existing system at the computer centre, it was divided up into three programme-parts. The programme texts are in disc storage, the data for the transition from one part of the programme to the next pass through drum storage (compare fig. 1-1).

The first part of the programme is for the feeding of the composer's data and for the selection of a tempo, a duration for the structure, the formation of an ensemble for the various parameters, the calculation of the average entry delay and the resulting number of entry points. As well as this, the hierarchy and the order of the chord sizes are established.

Check data I are followed by a printout of the data placed in drum storage during the transition to the second part of the programme. They are used for checking errors in the programme, and are in general of no interest to the composer. If this printout is missing, it was removed previously in the institute for checking purposes.

Fig. 11-2 shows the check data I which are printed per variant. The "specification" (entry 2) is printed per variant at the top right-hand side for purposes of more rapid orientation. The numbers in the first column of fig. 11-2 are not printed; they are merely references for the commentaries to the table.

	text	format	commentary
1	COMPUTED DATA VARIANT	*n*	
2	TEMPO:	*r*	
3	TOTAL DURATION:	*r*	
4	INSTRUMENT GROUPS:	*n m i i i ...*	1
5	REGISTER GROUPS:	*n m i i i ...*	1
6	ENTRY GROUPS:	*n m i i i ...*	1
7	DURATION GROUPS:	*n m i i i ...*	1
8	INTENSITY GROUPS:	*n m i i i ...*	1
9	REST GROUPS:	*n m i i i ...*	1 2
10	PLAYING MODE GROUPS:	*n m i i i ...*	1
11	NUMBER OF ATTACKS PER LAYER:	*n n n ...*	3
12	CHORD SIZE BY INSTRUMENT		4
13	ENSEMBLE:	*p i i i ...*	5 6
14	NUMBER OF GROUPS FOR CALCULATING SEQUENCE OF INSTRUMENTS REDUCED TO	*p*	6 7
15	SEQUENCE:	*i i i ...*	6 8
16	SEQUENCE OF INSTRUMENTS:	*i i i ...*	
17	SEQUENCE OF CHORD SIZES:	*i i i ...*	
18	CHORD SIZE BY HARMONY		4
19	NUMBER OF GROUPS FOR CALCULATING CHORD SEQUENCE REDUCED TO	*p*	7
20	SEQUENCE OF CHORDS:	*i i i ...*	
21	SEQUENCE OF CHORD SIZES:	*i i i ...*	
22	CHORD SIZE INDEPENDENT		4
23	NUMBER OF GROUPS FOR CALCULATING CHORD SIZE REDUCED TO	*p*	7
24	SEQUENCE OF CHORD SIZES:	*i i i ...*	
25	NUMBER OF TONES PER LAYER:	*n n n ...*	3
26	HIERARCHY FOR LAYER 1:	*i i i i i i i i*	
	HIERARCHY FOR LAYER 2:	*i i i i i i i i*	
	...		

Fig. 11-2

Survey CHECK DATA I

Commentary to fig. 11-2

(1) n = number of list elements,
m = number of groups in ensemble,
i = group indices.

(2) Is only printed if rests have been calculated.

(3) Number of entry points or tones per layer.

(4) The data-groups 12-17, 18-21 and 22-24 represent alternatives and refer to the respective regulation of the vertical density.

(5) p = number of elements in the ensemble,
i = enumeration of the list indices.

(6) Lines 13, 14 and 15 are repeated for each layer.

(7) p = number of ensemble elements. The reductions referred to in fig. 11-2 only occur if the order of the elements was selected by GROUP (with type = 2 or type = 4) and the number of possible group-sizes is greater than the number of ensemble elements or the number of chords or chord sizes. This line only appears in this case.

(8) Is only printed if the order of the instruments was determined by SEQUENCE.

11.3 CHECK DATA II

The second part of the programme is used to work out the matrix for the INTERVAL principle (if necessary) and to work out the score.

Check data II are followed by a printout of the data which were in drum storage during the transition to the third part of the programme. These data are used to check programme errors, and are in general of no interest to the composer. If this printout is missing, it was removed previously in the institute for checking purposes.

Fig. 11-3 shows the check data II which are printed per variant. The numbers in the first column of fig. 11-3 are not printed; they are merely references for the commentaries.

	text	format	commentary
1	TXCL	*matrix*	1
2	COMPUTED DATA VARIANT	*n*	
3	AVAILABLE	*n*	2
4	INSTRUMENT		3
5	ENSEMBLE:	*p i i i* ...	4 5
6	NUMBER OF GROUPS FOR CALCULATING INSTRUMENTS IS REDUCED TO	*p*	4 6
7	SEQUENCE:	*i i i* ...	4 7
8		*j j j* ...	4 8 9
9	HARMONY		3 10
10	ORDER FOR CALCULATION OF HARMONY:	*i i i* ...	11
11	TRANSP:	*i i i* ...	12
12	ADMITTED TONES:	*i i i* ...	13
13	TRANSP:	*i*	14
14	TRANSPOSED ROW:	*i i i* ...	14
15	ADMITTED TONES:	*i i i* ...	14
16		*j j j* ...	15
17	REGISTER		3
18	ENSEMBLE:	*p i i i* ...	5
19	NUMBER OF GROUPS FOR CALCULATING REGISTERS IS REDUCED TO	*p*	6
20	SEQUENCE:	*i i i* ...	7
21		*j j j* ...	8
22	ENTRY DELAY		3
23	ENSEMBLE:	*p i i i* ...	5
24	NUMBER OF GROUPS FOR CALCULATING ENTRIES IS REDUCED TO	*p*	6
25	SEQUENCE:	*i i i* ...	7
26		*j j j* ...	8
27	DURATION		3
28	ENSEMBLE:	*p i i i* ...	5
29	NUMBER OF GROUPS FOR CALCULATING DURATIONS IS REDUCED TO	*p*	6
30	SEQUENCE:	*i i i* ...	7
31		*j j j* ...	8

Commentary to fig. 11-3

(1) The matrix TXCL is only printed if the INTERVAL principle is used.

(2) n = number of locations still available in the computer. This quantity gives the composer an idea as to the storage space which could still have been occupied by data. Since the need for space for various auxiliary quantities increases with the amount of data, the number of locations actually available to the composer is less.

(3) The check data for the individual parameters are printed in the order of the respective hierarchy.

(4) These indications are omitted if INSTRUMENT is main parameter and was thus already calculated in the first part of the programme. In this case see CHECK DATA I (fig. 11-2).

(5) p = number of elements in the ensemble,
i = enumeration of the list indices.

(6) p = number of ensemble elements. The reductions referred to in fig. 11-3 only occur if the order of the elements was selected with GROUP (with type = 2 or type = 4), and the number of possible group sizes is greater than the number of ensemble elements. This line only appears in this case.

(7) Is only printed if the order of the elements was determined by SEQUENCE.

(8) Order in which the elements were selected for the score.

(9) The instrument numbers refer to tones in the order in which they are selected by the programme (compare "scoring" of chords in 8.1).

(10) The data for 11, 12-15 and 16 refer to the CHORD, ROW and INTERVAL principle respectively, and are only printed as required.

(11) Is only printed if HARMONY is last parameter.
i = line index of score.

(12) Order of transposition intervals; only printed if TRANS-CHORD = 1 or = 2. The order of the chords is printed in CHECK DATA I.

(13) Row given by composer, forbidden tones being omitted.

(14) Indications 13-15 designate the selected transposition interval for each transposition (only if TRANSP-ROW = 1 or = 2), the transposed row and the resulting row of "allowed" tones. The three indications are repeated for each transposition.

(15) Row of relative pitches generated according to the INTERVAL principle. Any inserted 0-pitches for percussion instruments are not printed.

	text	format	commentary
32	INTENSITY		3
33	ENSEMBLE:	*p i i i* ...	5
34	NUMBER OF GROUPS FOR CALCUL- ATING DYNAMICS IS REDUCED TO	*p*	6
35	SEQUENCE:	*i i i* ...	7
36		*j j j* ...	8
37	PLAYING MODE		3
38	ENSEMBLE:	*p i i i* ...	5
39	NUMBER OF GROUPS FOR CALCUL- ATING PLAYING MODES IS RED- UCED TO	*p*	6
40	SEQUENCE:	*i i i* ...	7
41		*j j j* ...	8
42	REST		3 16
43	ENSEMBLE:	*p i i i* ...	5
44	NUMBER OF GROUPS FOR CALCUL- ATING RESTS IS REDUCED TO	*p*	6
45	SEQUENCE:	*i i i* ...	7
46	REL.T. ABS.T. INDEX BEGIN	REST END	17

Fig. 11-3

Survey CHECK DATA II

Commentary to fig. 11-3

(3) The check data for the individual parameters are printed in the order of the respective hierarchy.

(5) p = number of elements in the ensemble,
i = enumeration of the list indices.

(6) p = number of ensemble elements. The reductions referred to in fig. 11-3 only occur if the order of the elements was selected with GROUP (with type = 2 or type = 4), and the number of possible group sizes is greater than the number of ensemble elements. This line only appears in this case.

(7) Is only printed if the order of the elements was determined by SEQUENCE.

(8) Order in which the elements were selected for the score.

(16) Is only printed if rests were calculated.

(17) Table with 6 columns.
REL.T. = provisional rest delay between the end of the last rest and the provisional entry point dp (compare fig. 7-12).
ABS.T. = provisional rest delay, counting from the start of the variant.
INDEX = entry point, referring to line index of the score.
BEGIN = definite rest delay, counting from the start of the variant.
REST = duration of rest.
END = definite end of rest, counting from the start of the variant.

11.4 SCORE

The printout of the score follows the CHECK DATA. The score is in the form of tables and gives the time values per layer for beginnings and ends of tones, arranged in chronological order, in seconds. The score printout is meant to be a complete collection of the data for a musical structure, regardless of its legibility or playability. The score printout is chiefly suitable for the transcription of the data into musical graphics, it being possible to convert the time quantities (seconds) into any desirable scale. Refer to the parts for transcription into traditional notation (see 11.6). The comments accompanying the score are explained in 11.5.

The score data are distributed over 9 columns; all parameter values of a single tone are printed in each line. In the columns for instrument, mode and intensity, repetitions are not printed. Chords can be recognized by their common entry points. Fig. 11-4 shows the 9 columns to contain the following information:

caption	information
(none)	line index
INSTRUMENT	name of instrument
MODE	performance indication
PITCH	absolute pitch
(REGISTER)	index of register employed
INTENSITY	intensity
ENTRY	time-point of tone-entry counting from start of variant
END	time-point of tone-end counting from start of variant
COMMENT	comment referring to parameter values inserted against the rules

Fig. 11-4

Within a variant group the individual variants are numbered. The "specification" is repeated on every page.

11.5 COMMENT

In the last column of the score comments are printed if necessary which inform the composer that because of the given rules one or more parameter values could not be found. These rules are either inherent to the structure formula or result from the construction of PR-2. In the latter case we speak of restrictions.

The comment is printed in the form of a capital letter indicating the parameter. It does not provide any enlightenment as to the respective cause. The letters have the following meanings:

I	INSTRUMENT
H	HARMONY
R	REGISTER
E	ENTRY DELAY
D	DURATION
A	INTENSITY
M	MODE OF PERFORMANCE

No comment is provided for REST, since no restriction exists and the rules can basically be adhered to. Here follows a list of reasons why the above-mentioned parameters might have comments.

<u>INSTRUMENT</u>

A comment is provided in the following cases:

(a) HARMONY comes first, no instrument can be found to play the selected pitch.

(b) HARMONY comes first according to the CHORD principle, all allowed instruments have been used before the chord is completely scored. In this case, moreover, repeated instruments are only entered for one more tone at a time.

(c) REGISTER comes first, no suitable instrument can be found for the selected register.

(d) DURATION comes first.
  (1) No suitable instrument can be found for the selected duration.
  (2) Equal durations are requested in the chord, but no instrument can be found to play the given duration.

(e) DYNAMICS comes first.
  (1) No suitable instrument can be found for the selected intensity.
  (2) Equal intensities are requested in the chord, but no instrument can be found to play the given intensity.

(f) PERFORMANCE comes first.

(1) No suitable instrument can be found for the selected mode of performance.

(2) Equal modes of performance are requested in the chord, but no instrument can be found to execute the given mode of performance.

HARMONY

If HARM = 1 no comment can be given, because HARMONY is then main parameter. However, a comment is given in the following cases:

(a) INSTRUMENT comes first, a melody instrument has been selected for which a suitable pitch cannot be found.

(b) REGISTER comes first, a register has been selected for which a suitable pitch cannot be found.

(c) HARMONY according to the ROW principle.

(1) INSTRUMENT or REGISTER come first, a percussion instrument or a 0,0-register have been selected, but the next row-tone is a relative pitch between 1 and tr. This tone is replaced by 0.

Note: blocked row-tones (entry 55) are skipped without comment. The same applies also to blocked instruments, registers and intensities.

(2) INSTRUMENT or REGISTER come first and request a relative pitch between 1 and tr which cannot be found either among the remaining tones in the row or in the next transposition of the row.

(d) HARMONY according to the INTERVAL principle. For the given interval the matrix does not contain any allowed succeeding intervals. Here the comment is: INTERVAL RESTRICTIONS TOO STRICT.

Note: Zeros inserted for percussion instruments in the "infinite" row do not give rise to comment.

DURATION

A comment is given in the following cases:

(a) INSTRUMENT comes first.

(1) No suitable duration can be found for the selected instrument.

(2) Equal durations are requested for the chord, but no duration can be found which is suitable for all instruments involved in the chord. This comment is always printed for the first tone of the chord, although it can also refer to other tones in the same chord.

Important: If the duration is to be the same length as the entry delay, but instrument and entry delay are already given, the automatically inserted duration (the same length as the entry delay) will perhaps not be suitable for all instruments involved in the chord. This duration is then shortened if necessary to the duration allowed for the instrument, but never lengthened.

(b) ENTRY DELAY comes first, duration must not be longer than entry delay, but such a duration cannot be found.

DYNAMICS

A comment is given in the following cases:

(a) INSTRUMENT comes first.
 (1) No suitable intensity can be found for the selected instrument.
 (2) Equal intensities are requested for the chord, but no intensity can be found which is suitable for all instruments involved in the chord. This comment is always printed for the first tone in the chord, although it can also refer to others in the same chord.

PERFORMANCE

A comment is given in the following cases:

(a) INSTRUMENT comes first.
 (1) No suitable mode of performance can be found for the selected instrument.
 (2) Equal modes of performance are requested for the chord, but none can be found to be suitable for all instruments involved in the chord. This comment is always printed for the first tone of the chord, but can also refer to others in the same chord.
 (3) A new mode of performance is requested per rest, but none can be found to be suitable for all instruments up to the next rest. This comment only appears for the first tone after the rest.

11.6 PARTS

The parts are printed after the score. This printout only occurs if LAYOUT = 2. The parts are intended to facilitate the transcription of the computer results into traditional notation. For this purpose the rhythmic context is divided up into metres which the composer can assemble to form measures as he sees fit. If necessary the metre is subdivided periodically so as to result in duplets, triplets, etc. The composer fixes the maximum number of subdivisions. Each time-value is compared with the possible subdivisions of the metre and rounded off to the nearest divisional point. This occurs per metre and per instrument so that there is no overlapping of different subdivisions.

Metrical subdivision is indicated by means of a two-digit number (more digits are used if there are more than 9 subdivisions), the first digit showing the number of subdivisions, the second the intended divisional point. A non-subdivided metre therefore only has a single metrical point: 11 (or 21, 31, etc.). A duplet consists of the metrical points 21 and 22, a triplet of 31, 32 and 33. More than 8 subdivisions will not be used as a rule. But still finer subdivisions can be clearly represented, e.g. subdivision into eleven: 111, 112, ..., 1110, 1111.

Both starting points and ends of tones are represented in this fashion. The respective metrical quantity indicates for the beginning of a tone its entry point, but for its end the entry point of the next rest or of another tone. If a tone's metrical duration is (42,43) it is a quarter of a metre, if the duration is (83,87) it lasts four eighths of a metre (starting at the third eighth). However, it can happen that a tone is given a duration of, say, (33,33) because the subdivision for the whole metre is calculated as three parts and the end of the tone is rounded off to 33. The tone is then actually shorter than a third of the metre and might be marked staccato.

The duration of the metre is arrived at per variant according to the metronome tempo, and also applies to several layers. If the metronome tempo is fast the metres can be so short that it is impossible to apply further subdivisions. The metres are numbered so that each tone is completely defined with regard to its position in time by the number of the metre and its metrical two divisional points.

A part is produced for each instrument used in the score in the order of the instruments in the instrument list. The comments are not reprinted. Each tone occupies two lines, one for the entry point, the other for the end point. These two lines are not always together, but can be recognized by their same index (column 1). The 7-column tables for the parts are captioned per variant with the current number of the variant and the name of the respective instrument, and also per instrument with the "specification". The seven columns contain the following information:

caption	information
INDEX	line-number of respective tone in the score
METRE	metre index
BEGIN	entry point of a tone in metric notation
END	end point of a tone in metric notation
PITCH	absolute pitch (as in score)
DYN	intensity (as in score)
MODE	mode of performance (as in score)

Fig. 11-5

APPENDIX : DATA FORM

1	user	
2	specification	
3	number of variants	
4	combination	0 no combination 3 register 4 entry delay 5 duration 6 dynamics 7 performance 8 rest
5	start number	
	INSTRUMENT	
6	list	
7	modes of performance	
8	pitches	
9	durations	
10	intensities	
11	chord sizes	
12	table	
13	number of groups	1 ALEA 2 SERIES 5 SEQ:
13	group selection	1 ALEA 2 SERIES 5 SEQ:
13	union	0 one layer 1 several layers (common harmony) 2 several layers (separate harmony)
14	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:

		HARMONY	dummies
15	principle	1 CHORD 2 ROW 3 INTERVAL	
	16 table		1
	17 order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ:	1
	18 transposition	0 no transposition 1 ALEA 2 SERIES 3 row:	0
	19 row		1
	20 transposition	0 no transposition 1 ALEA 2 SERIES 3 chromatic 4 serial	0
	21 matrix	MATRIX: 2 no matrix	2
	22 chord	: 0 no chord	1
	23 forbidden tones	: 0 no forbidden tones	0
	24 inversion	0 no inversion 1 inversion	0
		REGISTER	
25	list		
26	table	*(if combination, number of groups = number of I-groups)*	
27	group selection	1 ALEA 2 SERIES 5 SEQ:	
28	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:	

		ENTRY DELAY	
29	list		
30	table	*(if combination, number of groups = number of I-groups)*	
31	group selection	1 ALEA 2 SERIES 5 SEQ:	
32	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:	
		DURATION	dummies
33	list		0
34	table	*(if combination, number of groups = number of I-groups)*	1
35	group selection	1 ALEA 2 SERIES 5 SEQ:	1
36	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:	1
37	relation	0 independent of entry delay 1 duration = entry delay 2 duration ≤ entry delay	1
38	mode	0 equal durations in chord 1 different durations in chord	0
		REST	dummies
39	list		0
40	table	*(if combination, number of groups = number of I-groups)*	1
41	group selection	1 ALEA 2 SERIES 5 SEQ:	1
42	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ:	1
43	mode	0 no rests 1 d1: d2: (sound entry) 2 d1: d2: (general entry)	0

DYNAMICS		
44	list	
45	table	*(if combination, number of groups = number of I-groups)*
46	group selection	1 ALEA 2 SERIES 5 SEQ:
47	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:
48	mode	0 equal intensities in chord 1 different intensities in chord
49	TEMPO, list	
50	order	1 ALEA 2 SERIES 5 SEQ:
51	STRUCTURE DURATION, list	
52	order	1 ALEA 2 SERIES 5 SEQ:
53	VERTICAL DENSITY	0 depends on INSTRUMENT a: z: (autonomous density)
54	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ:
55	blocking	no yes instrument 0 1 harmony 0 1 register 0 1 intensity 0 1
56	hierarchy	0 SERIES 1 instrument 2 harmony 3 register 4 entry delay 5 duration 6 dynamics 7 performance
57	output	1 score 2 score, parts 3 score, parts, notation 4 punch tape

MODE OF PERFORMANCE		
58	list	
59	table	*(if combination, number of groups = number of I-groups)*
60	group selection	1 ALEA 2 SERIES 5 SEQ:
61	order	1 ALEA 3 RATIO: 4 GROUP a: z: type: 1 2 3 4 5 SEQ: 6 TEND d: a1: a2: z1: z2:
62	mode	0 per tone 1 per entry point 2 per rest
63	grid	t pitch grid f maximum subdivision d smallest time value

The entries must be defined in the ascending order of their call numbers, in full for the initial set (unless the last set of data finished with -2), with any degree of completeness for subsequent sets. The programme is executed when there is a 0 after the last entry of a set of data. The 0 can be replaced by -1 (no more data sets) or -2 (if the programme is to be continued later in accordance with the last set of data).

The data for an entry always finish with a semicolon (;), this also applies to the groups of a table, the call number being repeated for each group. Elements are separated by commas (,); always use decimal points in numbers.

The selection principles represent alternatives. A colon (:) indicates that more data are to come (put in brackets). Other call numbers referring to various modes, etc., are also alternatives. Exceptions: in entries 4 and 56, if no 0 is noted, any amount of call numbers can be stated, but each of them only once. Four call numbers are required for entry 55, either a 0 or a 1 for each of the stated parameters.

LIST OF ILLUSTRATIONS

INDEX

Page-numbers in bold type refer to passages where the respective item is discussed in detail; page-numbers in italics refer to illustrations or examples.