národní
úložiště
šedé
literatury

PC-GUHA Brief Manual

Harmancová, Dagmar
1994

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

PC-GUHA
brief manual

Dagmar Harmancová

Technical report No. 617

June 1994

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+422) 6605 2088 fax: (+422) 8585789
e-mail: dasa@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

PC-GUHA
brief manual

Dagmar Harmancová

Technical report No. 617
June 1994

Abstract

The manual is designed to help less experienced users to begin working with PC-GUHA software package.

Keywords

exploratory data analysis, GUHA method, PC-GUHA software package

1 Introduction

GUHA (General Unary Hypotheses Automaton) is a method for exploratory data analysis, which enables the recognition of regularities in empirical data - hypotheses based on relations between variables. The objective of the GUHA method is to extract all interesting information from the data. What can be "interesting", in this context, has some formal limitations imposed by GUHA, and is then specified by the user at the time of processing.

GUHA processes data in the form of a rectangular matrix, in which each row represents an individual record of a subject, and each column contains some quantified data regarding its attributes. It is also possible to work with incomplete data, that is, data which contains unknown values for some attributes.

GUHA searches the data for general, multifactorial relations between variables, that is, it forms a singular hypothesis concerning all the data in a sample, and interactions and interdependencies between some selected group of attributes.

By determining a class of possibly interesting hypotheses, the user defines what constitute so-called "relevant queries". These are propositions whose validity is checked in the given data. Valid relevant queries then become relevant statements. The results of processing data using GUHA consist of a list of all relevant statements about the data.

Various formal specifications of "interestingness" and different requirements on the types of variables studied invoke different GUHA-procedures. One of them (the most widely used), is the ASSOC procedure. This procedure processes binary, possibly categorical variables, that is, variables which may have only finitely many pre-defined values. Other variables (for example, having as a domain all real numbers) may be often transformed into categorical variables, for example, by dividing this domain into finitely many parts.

PC-GUHA is a software package for personal computers, which performs the ASSOC procedure. Apart from its own procedures, it contains an environment for processing data, for creating inputs for ASSOC, and for interpreting its results.

This manual is designed to help less experienced users to begin working with PC-GUHA. It doesn't cover the theoretical background of the GUHA method, or details of all the possibilities for using the system. (More specific information about the GUHA method can be found in references.) In searching for different ways to work with the system, the user can make use of PC-GUHA's comprehensive help system. Also this manual takes into account the availability of these help files, and does not go into great detail for topics which are covered more fully in the on-line help, which is accessed by pressing ⟨ F1 ⟩ on the keyboard. In the same way, this manual also does not include instructions and information which are displayed on-screen while the software is running, and it is highly recommended that users pay particular attention to the lower-most (information) line of the display.

2 Basic concepts

As was mentioned earlier, ASSOC processes only variables having finitely many different values. Variables transformed into finitely-valued variables by means of PC-GUHA system are called categorized variables.

Dichotomous variables, whose values can be expressed using "YES" and "NO", will be called **properties**. The core of the PC-ASSOC procedure (which is based on mathematical logic) operates using just properties: if it holds that categorical variable V possesses value K, then the new binary variable (property) V:K possesses the value "YES". Properties created in this way are called **basic properties**. It is clear that from any n-valued variable it is possible to create n different basic properties. Searching for multiple relationships between variables is, in the case of ASSOC, reduced to the examination of coherencies between the occurrence of properties derived from one group of variables and those derived from another group of variables.

To simplify expression, let's introduce some terms:

Literal is a basic property or its negation.

Elementary conjunction is a conjunction (using connective "and") of literals in which there are no occurrences of different literals created from the same variable. Basic properties, and their negations, are also considered to be elementary conjunctions . An elementary conjunction is a property - it has the value "YES" if and only if all literals in it have the value "YES".

Four-fold table of properties A and S for given data matrix is a quadruple of natural numbers a, b, c, d, in which a is the number of objects having the property A as well as the property S, b is the number of objects having the property A but not having the property S, c is the number of objects not having the property A but having the property S, and d is the number of objects not having the property A nor the property S:

	S	$NOT\ S$	
A	a	b	r
$NOT\ A$	c	d	s
	k	l	n

Let us also introduce the following notation: $r = a + b,\ s = c + d,\ k = a + c,$ $l = b + d,\ n = k + l = r + s,\ \ n$ is the number of cases in the data matrix.

Quantifier, in this manual, refers to a function, defined over four-fold tables, having the value 0 or 1. Because a four-fold table describes the occurrence of two properties, a quantifier with the value 1 will express that mentioned properties are in the relation described by quantifier.

PC-ASSOC generates **relevant queries** of the form

$A \approx S$ "A is associated with S"

or

$A \rightarrow^* S$ "A makes S likely",

where A and S are elementary conjunctions having no common variable, A is called the antecedent, and S the succedent, $\approx$ denotes association, $\rightarrow^*$ denotes (quasi)implication, and $\approx$ and $\rightarrow^*$ are quantifiers. It is possible to generate conditioned relevant queries $A \approx S/C$, where the condition C is a literal. Then the validity of $A \approx S$ is verified for all cases in which C is valid. The terms antecedent and succedent are generically referred to as 'cedents'.

Each time ASSOC is run, there must be a uniquely specified quantifier (and maybe a condition), and the procedure systematically generates antecedents and succedents (that is, permissible combinations of basic properties, and their negations) and verifies the validity of relevant queries on the corresponding four-fold tables. The output is then a list of the relevant queries (which are now called **relevant statements**), which are true for the given data matrix, that is, the corresponding quantifier has the value 1.

There are six quantifiers available in PC-ASSOC, each of which contains several parameters which the user can vary. These quantifiers are as follows:

Simple deviation:

Parameters: $BASE, \delta$.

It has value 1 iff $a \geq BASE$ and

$$a \cdot d > e^{\delta} \cdot b \cdot c.$$

Fisher quantifier:

Parameters $BASE, \alpha$.

It has value 1 iff $a \geq BASE$, $a \cdot d > b \cdot c$ and

$$\sum_{i=a}^{min(r,k)} \frac{\binom{k}{i} \cdot \binom{l}{i}}{\binom{n}{r}} \leq \alpha$$

Chi-Square quantifier:

Parameters: $BASE, \alpha$.

It has value 1 iff $a \geq BASE$, $a \cdot d > b \cdot c$ and

$$\frac{n \cdot (a \cdot d - b \cdot c)^2}{r \cdot s \cdot k \cdot l} \geq \chi_1^2 (1 - 2\alpha)$$

where $\chi_1^2 (1 - 2\alpha)$ is the $1 - 2\alpha$ quantile of the χ^2 distribution with one degree of freedom.

FIMPLE (founded almost implication):

Parameters: $BASE, CP$.
It has value 1 iff $a \geq BASE$ and

$$\frac{a}{r} \geq CP$$

.

LIMPLE (lower critical almost implication):
Parameters $BASE, CP, \alpha$.
It has value 1 iff $a \geq BASE$ and

$$\sum_{i=a}^{r} \binom{r}{i} \cdot CP^{\,i} \cdot (1 - CP)^{\,r-i} \leq \alpha$$

UIMPLE (upper critical almost implication):
Parameters: $BASE, CP, \alpha$.
It has value 1 iff $a \geq BASE$ and

$$\sum_{i=0}^{a} \binom{r}{i} \cdot CP^{\,i} \cdot (1 - CP)^{\,r-i} > 1 - \alpha$$

The first three of these presented quantifiers are called correlational, while the last three are called implicational. Correlational quantifiers express that an antecedent and a succedent "most often" occur together, implicational quantifiers express that the presence of an antecedent is "usually" followed by the presence of a succedent.

Most of the described quantifiers are based on well-known statistical tests for four-fold tables which suppose that the investigated cases are randomly sampled from a previous designated population. More detailed information concerning this topic can be found in the corresponding on-line help sections of PC-GUHA, or in statistical literature.

3 Working with PC-GUHA

Items of the main menu, which appear before the user when the program is started, roughly correspond to steps which are necessary to complete during experimental data analysis using PC-GUHA. Input for the procedures are the data to be analyzed, and the formulation of a problem, which is the denotation of the set of relevant queries (the term problem is used to mean one run of ASSOC). The output consists of all relevant statements saved in a solution file. Because the number of hypotheses found can be quite large, the system provides a means for working with them (such as sorting, selecting according to given criteria, and the like), and also a means for producing printed reports.

The first menu item, SET DIRECTORY, allows the user to specify (or create, if necessary) a working directory where the input files and outputted results are stored (if another directory is not specified). It is better to work with a dedicated directory, one which will contain only the working files for a particular project, instead of the

directory where the PC-GUHA software is located. This command should be used first when starting a project in PC-GUHA (please consult the help system for more details).

The second menu item, DATA PROCESSING, allows the creation, modification, and so on, of files with data to be analyzed. The triplet of files which are in the form used by PC-ASSOC is called the data matrix. For information about working with data matrices, see the chapter 4.

The next menu item, PC-ASSOC, leads to starting the actual calculations. The calculations will start directly, only when the data matrix and the file containing the related problem definition (the specifications for the set of relevant queries) are already prepared and available. If a suitable problem definition has not yet been created, then the system allows the user to do so at that time. For more information about formulation of the problem definition, see the chapter 5.

The item PROBLEM DEFINITION allows simplified repetition of calculations (for example, with changed parameters) using a previously saved problem definition, or the modification of a saved to create different problem definitions. Details regarding this process are found in the chapter 5.

The item INTERPRETATION allows the user to easily perform further operations on the solution file. A detailed description of these possibilities is contained in the chapter 6.

Selection of the item TEXT OUTPUT allows the user to create printed reports of the found hypotheses, which can either be sent directly to a printer, or saved to a file on disk. The user can select either a brief report format containing only the numbers of variables and categories, or a more lengthy text description, choose the accompanying numeric characteristics, output format of the hypotheses, etc. These options are outlined in the chapter 7.

Menu item TEXT EDIT may be used to invoke an external text editor which may be used not only for reviewing the created text output, but also for making some additional, non-standard changes. The editor which is provided with PC-GUHA uses commands which are similar to most standard text editors. In view of the fact that the information line and the on-line help system provide sufficient instructions for using its use, this manual will not discuss the editor any further. It may be helpful to point out, however, that after entering into the text editor, it is not possible to return to the main menu by pressing ⟨ESC⟩ on the keyboard, but rather by using ⟨ALT-X⟩ (holding down the ALT key and then pressing X). Users may also use their favorite text editor with PC-GUHA, by calling the program with the parameter /Exxxxxxxx, where xxxxxxxx is the full name (including the complete directory path to where the editor is located).

Some general suggestions for working with PC-GUHA:

In addition to the on-line help system and the information line, any alerts, warnings, or questions from the program, displayed on the first line of the screen, should also be heeded.

The user can select the active command by using the keyboard to move the cursor (highlighted field) to the desired item, and then pressing ⟨Enter⟩ to execute it.

Returning to the menu from which a command was executed, is usually possible by pressing the ⟨ Esc ⟩ key; in a few cases, in order to prevent accidental exiting from the current environment, the ⟨ Esc ⟩ only moves the cursor to the menu item which enables a return to the previous menu.

4 Input of data to be processed

PC-GUHA has its own facility for direct inputting of data, or it can read data acquired in some other manner. Input data to be read by PC-GUHA must be in numerical format.

If the user has some data files available, which are already saved on computer, they must first be modified for input into PC-GUHA. Acceptable formats are the BMDP file format, or a text file in which related data for each case (record) is grouped together, each attribute (field) is separated by a comma or a space, all variables are written in the same order each time, and data for each case begins on a new line, or is separated from the previous one by a slash (/).

Data management in PC-GUHA is started from the main menu by selecting the menu item

DATA PROCESSING.

It is then possible to create a new Data Matrix, or import a data file, by choosing

CREATE A NEW DATA MATRIX,

or one of the items

IMPORT ASCII FILE
IMPORT BMDP FILE.

To process data from a BMDP file it is necessary to use command IMPORT BMDP FILE. A detailed description of this procedure may be found in PC-GUHA's on-line help system.

If a data matrix is created using the IMPORT ASCII FILE command, the names of variables should be on first line text data file. All variables are assumed to have values that are real numbers with no more than two decimal places, with a maximum range of ten thousand. After the data is finished being read, the user can select the menu item

MODIFY THE DATA MATRIX

to arrange the formats of variables so they will agree with read data. (The format of a variable is written as a pair of numbers, the first of which is the length of the variable (number of characters), and second one giving the number of decimal places. (For the case of a number with a non-zero number of decimal places, it is necessary to take into account that the written form of a number includes a decimal point and a sign too.) Using this command, it is not possible to read incomplete data written

in columns in which missing values are marked by an empty field. The program skips over blanks and reads the next available value into the variable. Missing values must be marked by an arbitrary non-numerical character (each occurrence of this character - even if entered by mistake - is interpreted as one missing piece of data) or by a specific numerical code, which must be later designated as the code for missing data.

If the pre-defined format 8.2 is not suitable for most of the variables, or if, for some reason, it is difficult in the data text file to create the line with names of variables, it is better to use the command CREATE A NEW DATA MATRIX, after which the program first asks for creation of the description of variables. During its creation, it is only required to assign names for the variables, but the user is recommended, in this case, to specify a suitable format for each variable, as well. In the event that there are some missing values in the data, marked by a numerical code, it is possible, for each variable, to specify this number as the code for missing data. When the description of variables is finished, the system will proceed to the menu MODIFY THE DATA FILE which includes the items

COMMENT
DATA EDITTING
IMPORT
VARIABLES

The command IMPORT allows the user to read data from an existing file. The name of this file (including the complete path, if necessary) is entered after selecting the item FILE, in the lower part of the screen. If the user wishes to read the entire text file, he may directly select the command IMPORT FILE, which begins this process (ASCII is the pre-defined file type for the IMPORT command). It is also possible to read only some of the cases (records) - the needed data is entered after selecting NUMBER OF CASES.

By using the method described above, it is also possible to read incomplete data (as was mentioned), if it is written in aligned columns. From the menu which is displayed after selecting the command IMPORT, the user must select the item TYPE and select FORMAT (which enables the reading of files written in 'fixed format', as is used by the programming language FORTRAN) from the offered choices. Then the data will be read into variables by its exact position in the rows of the file.

If the user wishes to create a new data matrix directly using PC-GUHA (without using any intermediate text file), it is necessary to use the command CREATE A NEW DATA MATRIX. After assigning the matrix a name (the system automatically goes to the procedure for describing variables), the user must enter names for all variables which will be input into the file (it is not possible to add new variables to the data matrix once data has been entered). Direct input of data is started by selecting the command DATA EDITING.

The values for each of the variables are then entered into the highlighted field (whose size is determined by the previously defined format). It is possible to move through the matrix by using the normal cursor (arrow) keys, entry (or modification) of values can be started after the user presses any alpha-numerical character on the keyboard, or the ⟨Enter⟩ key. Completing the entry of a value is also accomplished

using the ⟨ Enter ⟩ key. Each session of entering or modifying data is ended by pressing the ⟨ Esc ⟩ key, after which the system returns to the previous menu.

The command DATA EDITING is suitable for viewing and modifying data matrices. Any row of a matrix (containing all the data for one case, or record) can be deleted by pressing the ⟨ Delete ⟩ key, holding down the ⟨ Ctrl ⟩ key and pressing ⟨ F8 ⟩ clears the entire matrix (all values will be erased, but the variable descriptions will remain).

Checking a large data file may be more convenient to do on paper than on a computer screen. Data matrices from PC-GUHA are possible to "export" to an ASCII file using the command EXPORT ASCII FILE from the DATA PROCESSING menu.

5 Problem formulation for PC-ASSOC

One of the inputs necessary for a run of the PC-ASSOC procedure is a defined set of relevant queries.

The number of literals which each antecedent and succedent may contain, which variables are allowed to occur in the antecedent, and which in the succedent, respectively, and which method will be used to create binary properties from multi-valued variables must be determined by the user.

Other information which is used to define a class of relevant queries is pre-defined (default values in the system), and may be changed. It contains, mainly, a quantifier and its parameters, a method for reducing the number of resulting hypotheses, and also a method for handling missing information, as in the case of incomplete data.

All this input information, combined, will be called a problem formulation. It is possible to save a finished problem formulation in a disk file, which is known as a problem definition, and to use it later for computation, or for the creation of other problem formulations.

After selecting the menu item PC-ASSOC from the main menu, and determining a data matrix, the user is asked if he would like to use a stored problem definition. If not, the system goes directly to problem formulation.

5.1 Defining variables

One thing which must be prepared by the user, even if he wishes to use the pre-defined (default) values, is the definition of the variables. The template for defining variables appears after the user selects the item VARIABLES. The screen is divided into a few fields; the upper part contains a field for defining variables which allowed to occur in the antecedent, and similar fields for the succedent and the condition (the user can switch between these fields by pressing the ⟨ Tab ⟩ key), the lower part displays information about the selected variable.

Definition of variables begins by pressing the ⟨ Enter ⟩ key. A list of all variables present in the data matrix will appear on the screen, from which the user may select a variable to work with. Variables which were defined as antecedent variables are shown in the list marked by an 'A', succedent variables by the letter 'S', and the variable for the condition is marked by the letter 'C'. (Warning! The variable for the condition may

not occur as an antecedent or succedent variable!) Then it is possible to determine (this requirement is not mandatory) the maximum and minimum admissible values for the variables. (In this way it is possible to exclude outlying and probably erroneous values, which will be treated as missing values during processing.)

What follows next is a very important basic element - categorization of the variable. There are three types of categorization available:

YES/NO: The variable will be categorized as a property, which will have the value 1 if its input variable has the value 1, and will have the value 0 for all other valid values of the input variable.

CODES: Each defined category is determined by one or several values of the input variable. The user can enter the admissible values into the Value column in the displayed table, or he can use the ⟨ F9 ⟩ key, which selects all valid values of the variable and displays them (the first 99, if there are more) in the Value column, in increasing order. In the Frequency column, next to each value, the number of its occurrences in the data matrix is displayed. If nothing else is specified, then each value will have one corresponding category, and ordering of the values will be taken into account during the generation of relevant queries.

INTERVALS: Each defined category is determined by one or several values of the input variable. The boundaries of the intervals are entered by the user into the Intervals column, by entering only the upper-bounds of these intervals, in increasing order. The upper-bound of an interval is interpreted as the lower-bound of the following interval. (Values which are exactly equal to the one of these bounds are considered as belonging to the interval for which it is the upper-bound.) If nothing else is specified, then each interval will have one corresponding category.

Ordering of categories, for the last two types of categorization, may be changed by entering the number of each category in the Number column. By attaching the same number to different items, it is possible to connect several different categories together, so they will be treated as one category. The number of a category is not allowed to be greater than the amount of defined categories. Individual categories may be assigned a name (no more than eight characters long) in the Name column. Different categories must have different names. If a category is created by the connection of several items, as described above, then its name will be taken from the first one of them. If no name is assigned to a category, then it will be given a name starting with '*' followed by its value. If such a name would be longer than eight characters, or if it is a category created from the combination of several categories, then it will be given a name of its category number between two asterisks (*1*, for example).

After finishing defining of categories, the system allows the user to determine whether the category may be included in the relevant queries only in its original form, or in its opposite form ("the value of a variable does not fall into a given category"), or both forms.

(From this point of the problem definition, the system suggests, for a given type of category, generally acceptable choices, which the user may accept by pressing ⟨ Enter ⟩ where necessary.)

The next choice which is given to the user is to decide if a defined variable should belong to the group of basic or of remaining variables. (For the group of basic variables, it must be true that each relevant query contains at least one of them.) Basic variables are marked by the letter B, remaining ones by the letter R. If no variable is marked as basic, then the occurrence of variables in relevant queries has no constraints.

The choice which can be made regarding a given variable, is to decide if this variable will, along with some other variables, constitute a common class. For variables in a particular class, only one of them may be contained in a relevant query. Variables which make up a class must directly follow each other when they are defined.

5.2 Remaining items

Besides defining admissible properties, the user is recommended to determine admissible **lengths of the antecedent and succedent** (if he is not satisfied with the pre-determined length, which is 1). Length of the antecedent or succedent is understood to be the number of literals (which is the number of variables) from which it is composed.

The next possible choice is that of a **quantifier**. If the user wishes to change the quantifier, the system offers them a choice of six quantifiers, listed in the chapter 2. Once a quantifier is selected, the system allows the user to change the pre-defined values of corresponding parameters (parameters are further explained in the on-line help system). It is possible to change the value of a BASE, which is the minimal amount of cases (records) for which the antecedent and the succedent must both be valid, in order for a relevant query to be considered "interesting" (as described in the introduction). Its pre-defined value is 5% of all cases in the data matrix (no less than 3).

If there are some cases (records) in the data matrix which have incomplete data, it is possible to choose a method for handling missing information. The user may select from the following three possibilities:

secured - using the secured completion method, a relevant query is considered to be a relevant statement if it holds true for all possible completions.

deleting - the deleting method disregards all cases which have incomplete information in the antecedent or succedent, or possibly the condition.

optimistic - using the optimistic completion method, a relevant query is considered to be a relevant statement if it holds true for some of the possible completions.

The last available option for problem formulation is the selection of a method for reducing the number of resulting hypotheses. It is understandable that too great an amount of outputted statements decreases the meaningfulness of the results. The user can decrease the length of the resulting solution file by selecting a means for improving:

if it is possible to add some literals to the antecedent of a relevant statement, without decreasing its validity, then these literals will be written with these hypotheses and the corresponding hypotheses will not be placed into the output file. Details about this can be found in the on-line help system.

A completed problem formulation may be saved on a disk. If it is desirable to repeat calculation using a modification of this problem formulation, it is possible to use the saved problem definition file and make only the necessary changes. This is possible to do by using the menu items PROBLEM DEFINITION and PC-ASSOC from the main menu. In the first case, the used problem definition determines the data matrix on which these calculations will be performed. The item PC-ASSOC allows the user to perform calculations on a data matrix using a problem definition created for another, similar data matrix, not having more variables than the current matrix. For problem formulation, input variables will be taken by corresponding order numbers. The user must carefully judge whether this problem definition is usable, and make the modifications accordingly. This modified problem definition can also, of course, be saved.

6 Working with results

If the results of computations from PC-ASSOC contain a large amount of offered hypotheses, it can be very difficult to evaluate them. PC-GUHA offers the user a means for sorting these hypotheses, and for creating sub-files according to given criteria. The INTERPRETATION mode can be accessed from the main menu, and if the user doesn't exit from it, it is offered in all cases when the solution of a problem is completed.

Work always begins (even immediately after the completion of calculations) with the reading of the solution file (the file with the extension .SOL, and its name is displayed on the first line of the screen). Because it is often not very easy to work with a very large file, the system offers the possibility of reading only part of a file by displaying, which the user can change, the number of hypotheses and the starting position of the first hypotheses to be read from the file. Pre-defined values are 500 hypotheses (or all, if 500 or less are contained in the file) and starting from hypothesis number 1. Basic information about the file to be read are displayed on the screen.

After the file is read, the user can select options from the interpretation menu.

6.1 Summary information about the solution file

The item PROBLEM SUMMARY offers facts about the studied file. Here can be found some basic information about the problem formulation (such as the quantifier used, and the method used for reduction of the amount of resulting hypotheses, for example), and other items which can be changed, such as the longest and shortest antecedents in the file, and the range of values of numerical characteristics. Attributes which can change are displayed on the screen in three columns. The first one is related to conditions given for the problem solution, the second is related to the file which was read, and the third to the file which is created during the interpretation process according to

the user's specifications. The use of some values for characteristics may require longer computing time (the calculations can be done during work on the interpretation) - these values will be displayed after the computations are finished.

6.2 Choice of hypotheses for interpretation

The item CHOICE OF STATEMENTS allows the user to select only some hypotheses, using some given criteria. The subset of relevant statements is determined by limiting the lengths of the antecedent and succedent, by limiting the occurrence of variables and categories, and possible by limiting the values of some numerical characteristics. In the upper-right corner of the screen is displayed the number of hypotheses in the input file, and the number of hypotheses which have satisfied, up to that time, the given criteria. Conditions for the occurrence of variables and categories are specified using a special mode which is entered after the item VARIABLES is selected. The other limitations are specified in the usual way, directly from the screen for choice of statement. After an item is selected, a table will be displayed in the lower part of the screen, containing information about number of hypotheses which fit the interval of values of the given characteristics in both the original file and the currently selected subset of hypotheses. Evaluation of newly specified characteristics and calculation of the numbers of corresponding hypotheses in the table sometimes requires more computation time, so the user must wait for this information to be updated.

When entering the mode started from the item VARIABLES, the user has available on the screen a list of the given antecedent and succedent variables, and to right of each is shown, by a plus or minus sign, whether positive, negative, or both types of literals occur in the interpreted solution file. A minus sign is displayed to the left of variables which do not occur in the given solution. The lower part of the screen displays more information about the variable which the user has currently highlighted with the cursor: information concerning whether it belongs to the group of basic variables, the type of categories, if its type is YES/NO, then the respective numbers of "YES" and "NO" values which occur in the data are given, if its type is CODES or INTERVALS, then the number of categories is given, and if the variable contains missing values, then the numbers of valid values and missing values are given, respectively. The cursor can be moved from one variable to the next by using the cursor (arrow) keys, and between antecedent and succedent by using the ⟨ Tab ⟩ key.

The user can use the ⟨ Ins ⟩ key to mark the currently highlighted variable (only those which occur in the solution can be marked), which will then have an asterisk displayed to the left of it. Pressing the ⟨ F8 ⟩ key switches between three possibilities which may be selected of how these marked variables will influence the choice of hypotheses:

all marked variables - a relevant statement will be placed in the interpretation only in the case where the corresponding cedent (including improving literals) contains all marked variables.

only marked variables - a relevant statement will be placed in the interpretation only in the case where the corresponding cedent contains only marked variables,

and no others.

some marked variables - a relevant statement will be placed in the interpretation only in the case where the corresponding cedent (including improving literals) contains at least one of the marked variables.

Marking of variables is done independently for the antecedent and succedent. If the user wishes to switch the marking of all variables (from unmarked to marked, and vice versa), he may do so by pressing the ⟨ F7 ⟩ key.

By using the ⟨ F10 ⟩ key, the user can influence the occurrence of literals in the solution file. When the cursor is placed on a variable for which both positive and negative literals occur in the solution, pressing ⟨ F10 ⟩ and then selecting the displayed plus '+' sign with the cursor will specify that hypotheses containing negative literals will be excluded from the solution file, and the opposite by selecting the minus '-' sign. (The choice of literals will have no effect on the improving literals.)

The ⟨ F9 ⟩ key can be used to influence the occurrence of categories in the selected hypotheses for variables with categories of the type CODES or INTERVALS. After ⟨ F9 ⟩ is pressed, a table showing the categories of the highlighted variable is displayed, which gives the number of occurrence of the variable in those particular categories; categories which do not occur in the given solution file are designated by a minus sign to their left. The ⟨ Ins ⟩ key can be used to mark one or several categories which occur in the solution, and relevant statements which contain literals corresponding to unmarked categories will be removed from the solution file. (This marking of categories will have no effect on the improving literals.)

6.3 Ordering of hypotheses in the output

The item ORDERING OF STATEMENTS enables the user, while examining the output on the screen, to determine the order in which hypotheses will occur in the solution file.

During the solution of a problem, relevant queries are generated and verified in what is referred to as 'basic order'. In this order, shorter relevant queries (the same is true for the antecedent and succedent) precede longer ones. Variables in the antecedent and succedent, respectively, are generated in the order in which they were given during the definition of the antecedent or succedent variables. The order of literals is determined by the given numbers of their categories (if the user did not define these numbers, they are numbered by the program in the order in which they were defined), with the convention that basic properties precede their negation.

During interpretation it is possible to change this ordering. The program offers several criteria for the ordering of hypotheses. If one criterion does not order the hypotheses in an unambiguous manner, which means that there are groups of hypotheses that equivalent in relation to this criterion, then it is possible to add another criterion which can then determine the order for these groups. In this manner, it is possible to select no more than four criteria from the following list: [1]

[1] The given criterion can be removed using the item ERASE THE LAST CRITERION

- to order antecedents
  - by increasing length
  - by decreasing length
  - lexicographically

  By lexicographic ordering is meant ordering in which the order of literals is given, first of all, by the order of variables given in the problem formulation, for each variable this is done by the order of categories given in the same manner, with a basic properties preceding their negations. Elementary conjunctions are ordered by comparing the first literals, if the first literals are the same, then the second literals are compared, and so on.

  Also in both methods of ordering according to length, the order of antecedents is completely determined - by order according to length is understood ordering in which antecedents of the same length are ordered lexicographically.

- to order succedents
  - by increasing length
  - by decreasing length
  - lexicographically

  The same conditions which apply to the ordering of antecedents are also true for the ordering of succedents.

- according to frequencies A & S (which is the number of objects which satisfy both the antecedent and succedent)
  - ascending
  - descending

- according to numerical characteristics

6.4 Examining results on-screen

After selecting the menu item OUTPUT TO THE SCREEN, the user can examine particular hypotheses from the studied solution file. Only hypotheses which satisfy the conditions given using the item CHOICE OF STATEMENTS are displayed, in the order given using the item ORDERING OF STATEMENTS.

One screen is used to display each relevant statement. The heading of this screen contains the name of the solution file, the number of hypotheses in the file, the original number of the displayed hypothesis (which is the order in which the statement was generated by PC-ASSOC), and the new order number assigned by the designated ordering of the file.

In middle part of the screen, information about the antecedent and succedent is displayed, for relevant statements with a condition, the condition is also displayed.

The length of the antecedent and the succedent is given, respectively, along with a list their respective literals. If improving was used, then the improving literals, if there are any, are included in these lists. These improving literals are separated from the others by a row of dots (.....). If there are more literals than can be displayed at once on the screen, the user can use the ⟨ Tab ⟩ key to select either the ANTECEDENT or SUCCEDENT column, and then use the cursor (arrow) keys to move up and down in this list.

The lower part of the screen displays a four-fold table and the statistical characteristics corresponding to the quantifier used, and any characteristics used for selecting and ordering the relevant statements.

While in examination mode, it is possible to create a new solution file by using the ⟨ F8 ⟩ key. When ⟨ F8 ⟩ is pressed, the currently displayed hypothesis is saved to a file. The first time this option is used, a new file is created, which is closed when the user moves to another interpretation or another solution file.

The ⟨ F9 ⟩ key allows the user to create a text file. This option saves the currently displayed hypothesis, exactly as it appears on the screen, and one line of comments, if the user enters any. The first time this option is used, a new file is created, which is closed when the item RETURN TO THE MAIN MENU is selected from the Interpretation menu.

Pressing ⟨ F10 ⟩ sends the currently displayed relevant statement, and an optional one line of comments, directly to the printer.

6.5 Other options available from the Interpretation menu

The user can create a new solution file which contains the relevant statements which satisfy the conditions for selection and ordering which were specified during interpretation. The old solution file can then be deleted. (Warning! If the user wishes to save the transformed solution file, he must not delete the old solution file before saving the new one.)

The formulated conditions for ordering and selection of hypotheses can be used for another solution file. Any specifications given to the variables for the selection of relevant statements will not be used.

7 Creating reports for printing

The user can select which information, and in what format, will be included in the text output for a given solution file. Choosing the item TEXT OUTPUT from the main menu will display a table containing the pre-defined (default) parameters for text output, which may be changed.

The default value for the heading for an output report is the comment which is given during problem formulation. This can be modified, and must be no more than 80 characters long. This title will be displayed at top of each page.

The number of hypotheses which will included in the output report can be specified by the user. The default value is the actual amount of relevant statements in the given

solution file, or 100, if the amount in the file is greater.

It is further possible to select which of the following four surveys will be used:

survey of variables

The description of variables will consist of the antecedent and succedent variables and the condition variable. Ordering of variables is according to the order in which they occur in the data matrix. Order number, name, category type, and description, is given for each variable. Format of values of the variables, boundaries for valid values, and the code used to designate missing information, will be included only if it was explicitly specified by the user.

survey of categories

This survey will include the antecedent, succedent, and condition variables, which were categorized by codes or intervals. The same information as was given in the previous survey is included for each variable, along with a description of its categorization.

- For variables of the category type CODES, a list of valid values which are included in its respective category is given. Each value is given along with its order number and the name of the category to which it was assigned.
- For variables of the category type INTERVALS, a list of the intervals is given, each along with its order number and the name of the category to which it was assigned.

survey of antecedent literals

the same way as in the problem formulation. For each variable, its name, the number of the order in which it was entered in the data matrix, and its description, if one was completed, is given. Further given is whether the literals created for the variable were allowed to be positive, negative, or both types. The column B/R shows whether the variable belongs to the group BASIC or REMAINING. Classes of the variables are separated by dotted lines. For variables with missing information, the number of cases (records) in which it had an unknown value is given.

For categorized variables, a list of their categories is given, each along with its number, name, and the number of cases in which the value of the variable fits that category. If some frequency is less than the value of the parameter BASE, it is designated by the # character. This means that the corresponding literal can not be included in any relevant statements.

survey of succedent literals

The same applies as for the preceding description of the survey of antecedent literals.

The user can choose whether the output will be created in the form of a table or a list of relevant statements.

In the list, the individual hypotheses are separated by lines. For each hypothesis, the number of cases which the antecedent and succedent are fulfilled, and the values of the numerical characteristics corresponding to the used quantifier, are given. Antecedent literals are listed on the left side of the output, and succedent literals are listed on the right. Literals are expressed by numbers of variables and categories, or by names of categories and names or descriptions of variables.

Please beware that the following method, which is atypical from the others used in PC-GUHA, is used for the selection of numerical characteristics and for the structuring the four-fold table: the marking of desired items is done using the ⟨ Ins ⟩ key, and the ⟨ Enter ⟩ key is used only to confirm that the selection process is completed (if the ⟨ Enter ⟩ key is pressed before desired items are marked, then numerical characteristics and a four-fold table will not be output).

If the user selects the option for output in tabular form, the output will contain one or several tables in which each hypothesis has one corresponding row, and each antecedent and succedent variable will have a corresponding column on the left or right side of the table, respectively. Literals which are present in the relevant statements are written in the column corresponding to the variable from which they were derived, literals which belong to the cedent of the relevant statement are underlined with ***, improving literals are underlined with ..., and negative literals are designated by a minus sign in front of their categories.

The format in which literals and variables appear in the headings of the table columns can be selected by choosing the item **syntax of literals**. When the item NUMBERS is selected, literals are written by category number, and variables are written by the order number of the variable (as determined during the description of the data matrix). When the item NAMES is selected, variables will be written by name and number, and categories will be written only by numbers. When the item NAMES OF CATEGORIES is selected, variables are written by name and number, and literals are written by names of categories. Classes of cedent variables are separated in the heading by a comma. Numbers of variables selected as BASIC are underlined by ===.

The first column of the table contains the number of the order in which the relevant statement was generated by ASSOC, the second, and possibly third, columns contain values of the numerical characteristics corresponding to the used quantifier. The column labeled A & S contains the number of cases for which the antecedent and succedent are valid.

If the selected output in tabular form requires more than the pre-defined 72 characters per line, the system will tell the user what the necessary length of a row should be. Then it is possible to change either the selected form of output of literals, or change the specified length of a row. If the users doesn't choose to make any of these suggested modifications, the literals will by written by category number, and names of variables in the column headings will output across the necessary number of rows. If after these modifications the determined row length is not sufficient for output in tabular form, the output report will be given as a list of hypotheses.

Text output in tabular form is not possible if it requires a row length greater than 160 characters.

References

Hájek, P. (1984). The new version of the GUHA procedure ASSOC (generating hypotheses on associations) - Mathematical Foundations, COMPSTAT 1984 - Proceedings in computational statistics, PHYSICA-VERLAG, WIEN, 360-365.

Hájek, P. & Havránek, T. (1978A). Mechanizing hypothesis formation - Mathematical foundations for a general theory. SPRINGER VERLAG, HEIDELBERG.

Hájek, P. & Havránek, T. (1978B). The GUHA method - its aims and techniques (with bibliography). INT. J. MAN-MACHINE STUDIES 10, 3-22.