

GOLDVARB 2001

A Multivariate Analysis Application for Windows.

John Robinson, Helen Lawrence & Sali Tagliamonte

October 2001

USERS' MANUAL

Table of Contents

[Introduction](#)

[Getting Started](#)

[Looking at data](#)

[Creating a new Token File](#)

[Factor Groups](#)

[Opening an existing token file](#)

Using the data

[No Recode](#)

[Recoding](#)

[Excluding](#)

[The Conditions \(.Cnd\) file](#)

[Exclusion from a single group](#)

[Results](#)

[Cross-tabulations](#)

Multivariate analysis

Other functions

[Find and Replace](#)

[Finding text](#)

[Finding Factors](#)

[Replacing factors](#)

[And/Or](#)

['And' function](#)

['Or' function](#)

[Elsewhere](#)

[Further information](#)

[Hints](#)

[Transferring files from GoldVarb 2.0](#)

[Error/warning messages](#)

References

Table of Figures

Figure 1: GOLDVARB 2001 window

[Figure 2: New Tokens window](#)

[Figure 3: First token](#)

[Figure 4: Comments in token file](#)

[Figure 5: Factor specification \('Group'\) window](#)

[Figure 6: Factor specifications](#)

[Figure 7: GOLDVARB 2001 window showing the log of a coding error](#)

[Figure 8: Existing token file](#)

[Figure 9: Token file with factor specifications](#)

[Figure 10: New Results window](#)

[Figure 11: No Recode results](#)

[Figure 12: Recode Definition window](#)

[Figure 13: Recoding a factor group](#)

[Figure 14: Condition file](#)

[Figure 15: Edited condition file](#)

[Figure 16: Cross-tabulation dialog box](#)

[Figure 17: Cross-tabulation \(Text\)](#)

[Figure 18: Cross-tabulation \(Grid\)](#)

[Figure 19: "Knockout" in a results file](#)

[Figure 20: Binomial Varbrul window](#)

[Figure 21: End of Binomial Varbrul](#)

[Figure 22: Best stepping-up run](#)

[Figure 23: Find text](#)

[Figure 24: Find factors](#)

[Figure 25: Finding factors in more than one factor group](#)

[Figure 26: Token identified through 'Find' function](#)

[Figure 27: Find and Replace function](#)

[Figure 28: Recode definition - 'And' function](#)

[Figure 29: 'And' function, final operation](#)

[Figure 30: Completed 'And' function](#)

[Figure 31: Recode definition - 'Or' function](#)

[Figure 32: 'Elsewhere' function in Conditions window](#)

Introduction.

GOLDVARB 2001 is an application for multivariate analysis, based on the previously circulated program GoldVarb 2.0 (Rand & Sankoff 1990). While GoldVarb 2.0 is run on Macintosh computers, GOLDVARB 2001 reflects the need experienced by many for a similar program for Windows. GOLDVARB 2001 is a stand-alone application, contained within a single .exe file and requiring no other specialist software.

The literature on variable rules is extensive (see for example Cedergren & Sankoff 1974; Sankoff 1978, 1988; Sankoff & Labov 1979, Sankoff & Rousseau 1979, Guy 1988,1993; Wolfram 1991). Further discussion of procedure and methods can be found in the manual for GoldVarb 2.0 (Rand & Sankoff 1990). Therefore the purpose of this manual/booklet is to provide a brief guide to the workings of GOLDVARB 2001 itself. There is an assumption throughout this manual that the user will have some experience of Windows.

GOLDVARB 2001 was developed at the University of York, as a collaborative project between the Department of Language and Linguistic Science and Computer Services. Many thanks are due to Steve Harlow who had the idea to create the Windows version and made the necessary funds available, to David Rand and David Sankoff for generous provision of the GoldVarb 2.0 source codes and to John Robinson of Quantic Computing (formerly of Computer Services, University of York) for his programming expertise and inexhaustible patience.

Helen Lawrence and Sali Tagliamonte

Department of Language and Linguistic Science

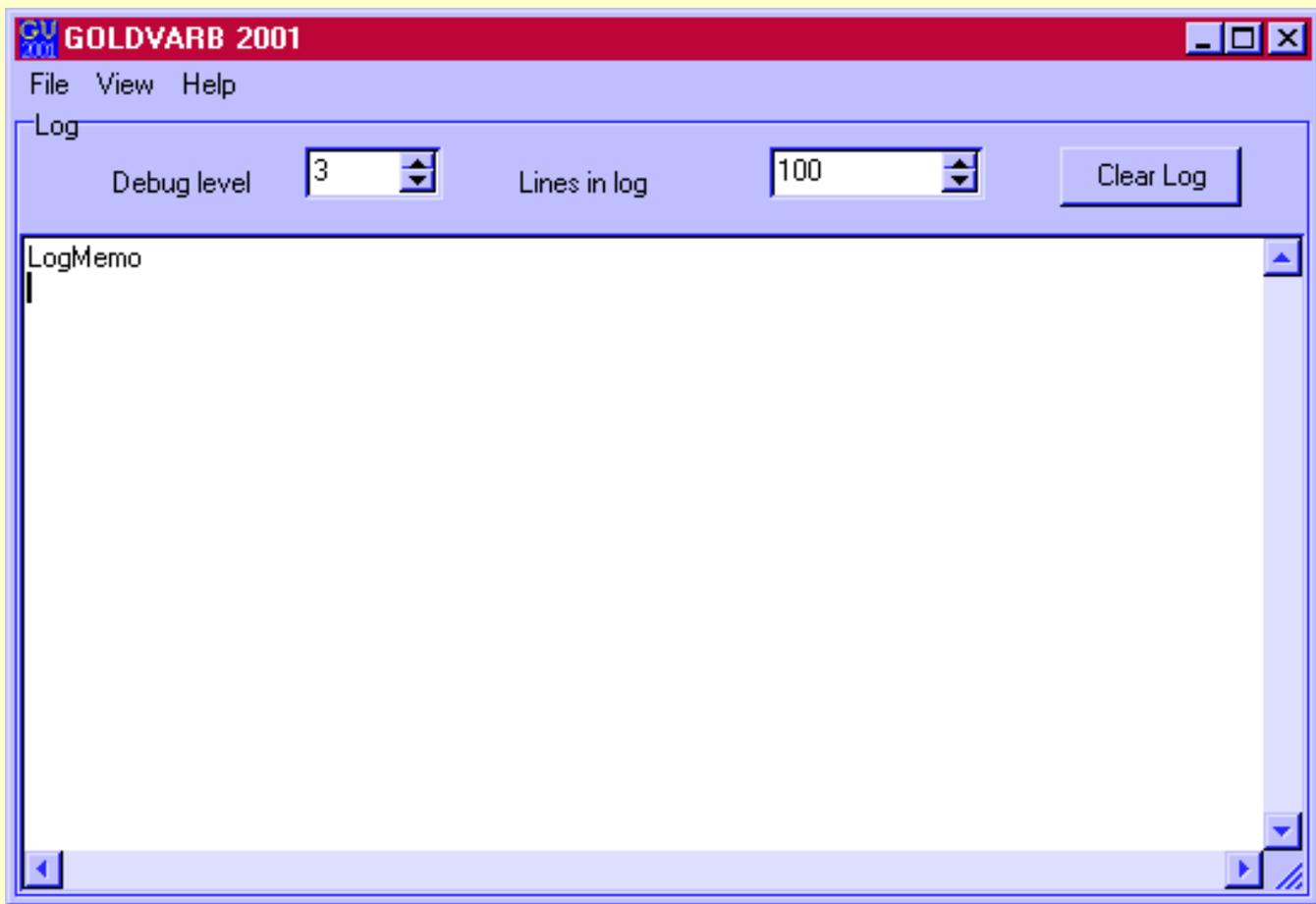
University of York

[Back to table of contents](#)

Getting started

When you open the application, you will see a window that looks like Figure 1:

Figure 1: GOLDVARB 2001 window



This is the main 'centre of operations' and will hereafter be called 'the GOLDVARB 2001 window'.

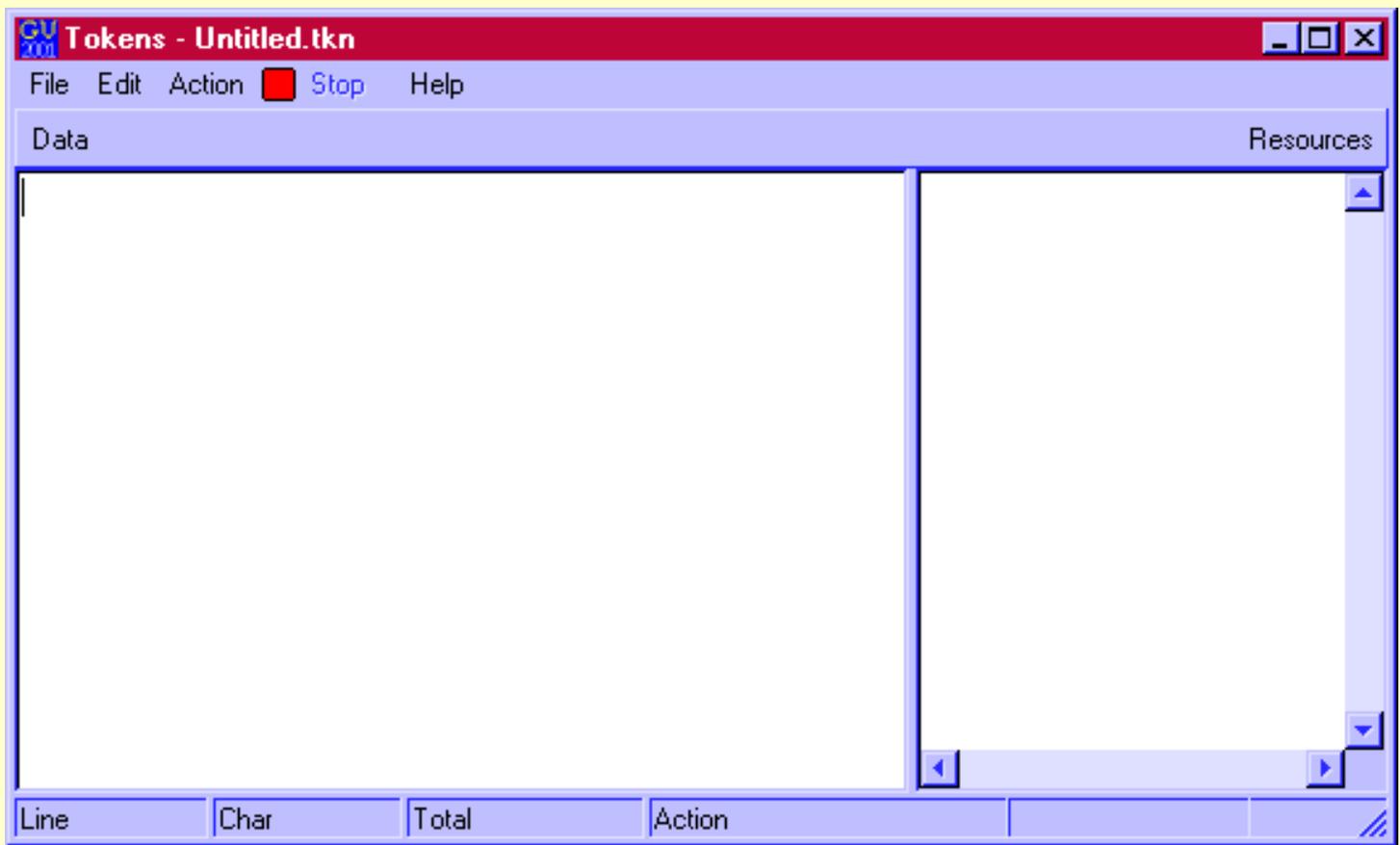
[Back to table of contents](#)

Looking at data

The data for quantitative linguistic analysis is made up of individual instances of a linguistic variable (see Wolfram 1991). These individual instances are known as **tokens** and are stored within a **token file**.

★ To access a token file, select Tokens from the View menu in the GOLDVARB 2001 window. This will open a window which looks like Figure 2:

Figure 2: New Tokens window



[Back to table of contents](#)

Creating a new Token file

From the File menu in this new Tokens window, select New.

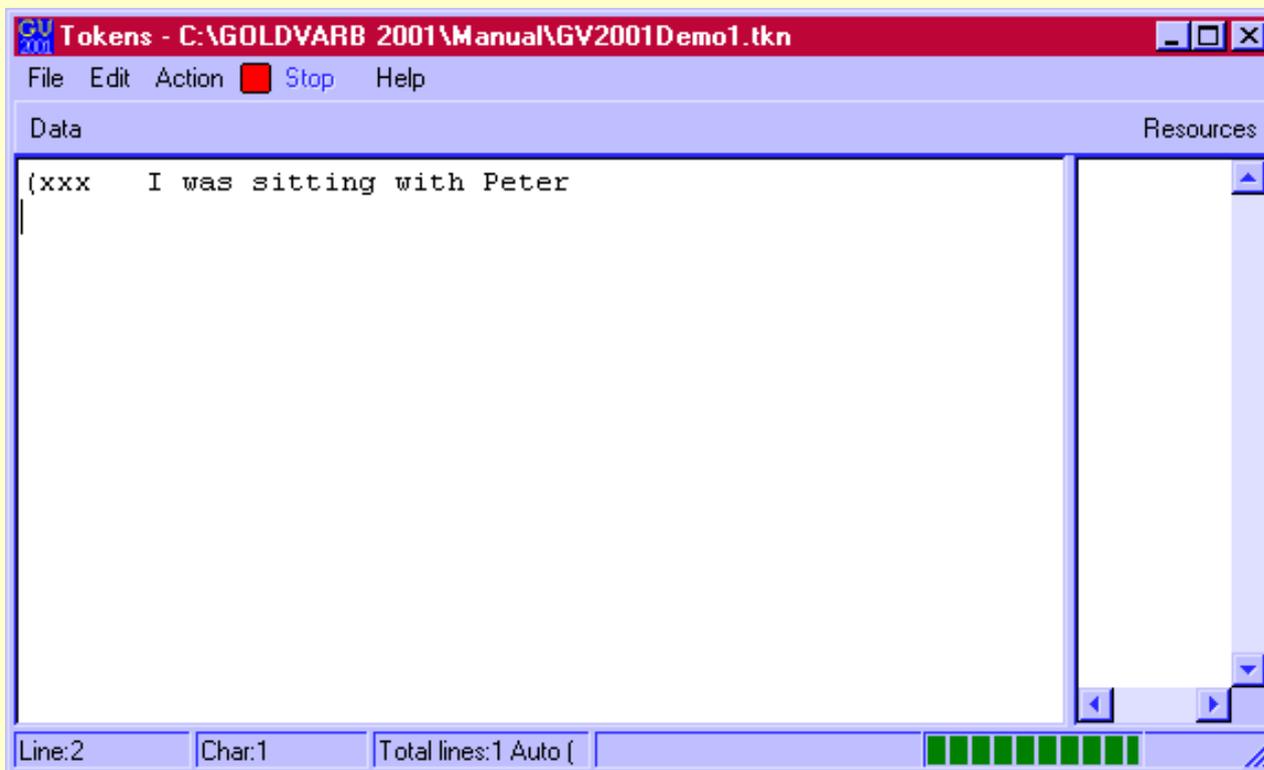
There are several issues to be aware of when you are creating a new token file.

The program will read as valid data any line which begins with a left parenthesis and will disregard any line which begins with a semi-colon. Therefore, when entering a token, ensure that it begins with an open parenthesis, "(", so that the token is recognised. If you wish to insert comments which are not part of the data, ensure that the line begins with a semi-colon, ";".

If you choose to record the token's content as well as its coding within the token file, leave at least one space (although 3 spaces makes the file easier to read) between the end of a coding string and the beginning of the data itself. This will instruct the program (which only reads the coding string) to proceed to the next line. **Do not** use tabs. They look the same, but the program does not recognise tabs as spaces.

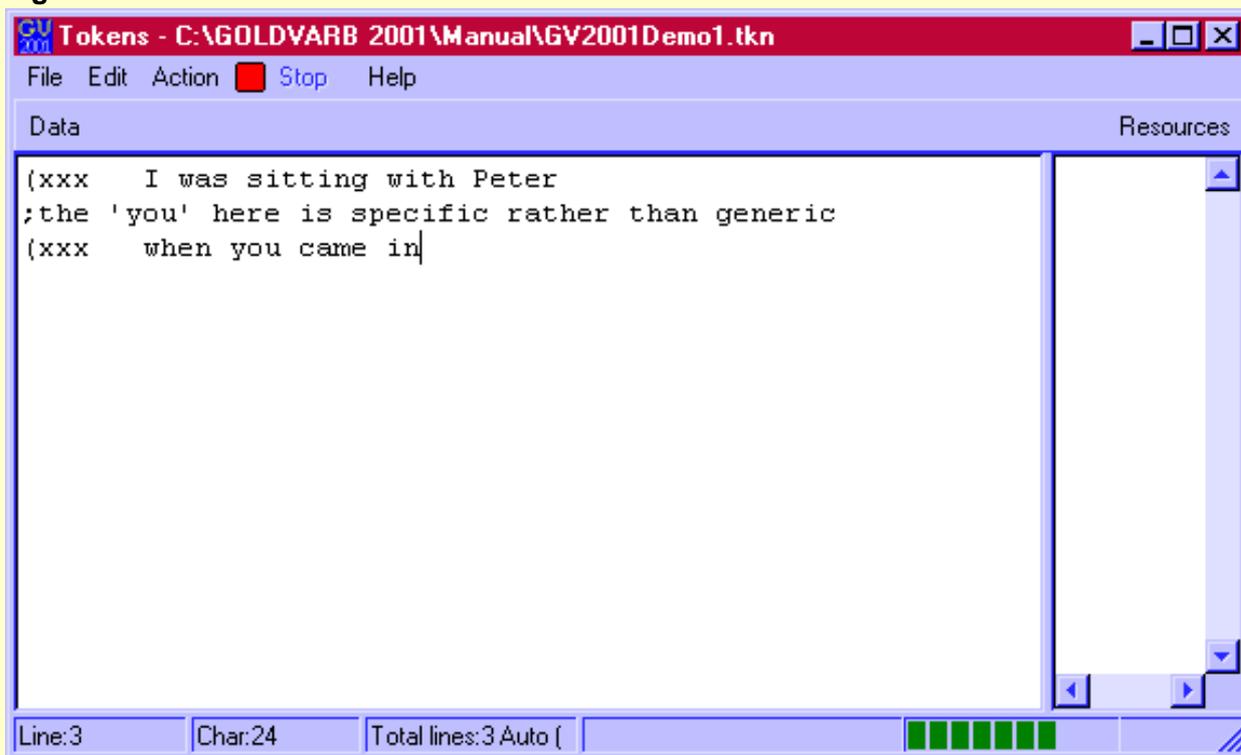
For example, suppose the sentence "*I was sitting with Peter*" contains an instance of the variable you are interested in. What you would enter into the token file would be as shown in Figure 3, where "xxx" represents the coding string of factors which will be discussed later:

Figure 3: First token



If you wish to make a comment about a particular token which may be useful later, enter it either above or below the token, ensuring that it starts with a semi-colon.

Figure 4: Comments in token file



[Back to table of contents](#)

Factor groups

In order for GOLDVARB 2001 to analyse the data, a selection of factor groups must be specified. These allow the linguist to code the data for such variables as verb type, subject type, vowel, presence or absence of a particular feature, speaker - in fact, any factor which may be considered to have an effect on the occurrence of a given variable. These groups are understood by the program as a set of codes - single characters which can be entered into the token file. The program allows any character in the Courier keyboard to appear as codes for a given token, each representing a specified variable.

For example, it may be necessary to code each token for the grammatical person of the subject. This will be entered as one factor group. The factor group will consist of as many factors as there are relevant distinctions within that factor group.

	Code	Example
1st person singular	1	I was sitting with PeterÖ
2nd person singular	2	when you came in
3rd person singular pronoun	3	He was delighted
Noun phrase singular	N	"When did you get here?" Peter said.
1st person plural	4	We all laughed.
2nd person plural	5	You lot always do that
3rd person plural pronoun	6	They never really understoodÖ
Noun phrase plural	n	how clocks work.

Thus to code for the grammatical person of the subject, the above coding scheme is apt. In order for GOLDVARB 2001 to perform analyses, it is first necessary to specify the factors for each factor group - i.e. to tell the program which characters are to be used as factors. This is particularly important to do whilst in the coding stage, as it is only by doing this that the program can identify mistakes you may make in the process of coding your tokens Ö which will undoubtedly happen!

★ In the GOLDVARB 2001 window, select Groups from the View menu, as shown in Figure 5.

Figure 5: Factor specification ('Groups') window

CU 2001 Groups

Groups

Number of Groups: 1 [New Group] [Delete]

Factors

New factor: abc in group: 1 [Add]

[Edit1] [Delete highlighted factor]

Group	Default	No. factors	actc
1	.	0	

[OK] [Save to Token file]

Click New Group. Then highlight the 'Factors' box by clicking on it, so that it is surrounded by dotted lines.

Type the characters you have chosen to represent your factors into the New Factor box, then click Add. When you are satisfied that a factor group is complete, click New Group.

Ensure that a 'default' factor is entered for each factor group - this must be selected from the factors specified for that group. Highlight the default factor box by clicking on it, and type in the desired factor.

Repeat this process for all the factor groups you wish to specify.

Figure 6: Factor specifications

CU 2001 Groups

Groups
Number of Groups: 4 [New Group] [Delete]

Factors
New factor: =1=4=4 in group: 4 [Add] [Delete highlighted factor]

Group	Default	No. factors	actc							
1	a	3	a b c							
2	1	8	1 2 3 N 4 5 6 n							
3	s	3	s p +							
4	z	8	z c B s l d u w							

[OK] [Save to Token file]

When the process is complete, click Save to Token file, then OK.

Before you exit GOLDVARB 2001, check your coding against the specified factors (Tokens window, Action|Check Tokens). Errors will be identified in the GOLDVARB 2001 window, as shown in Figure 7. Correct these before saving and closing the file.

The **dependent variable** is the factor group which encodes the linguistic variable under investigation. For example, if you have created the token file in order to examine the occurrence of an overt versus deleted copula, your dependent variable might have the characters **S** to indicate an overt copula (e.g. He is clever) and **Ø** to indicate a deleted one (e.g. He clever). The first factor group listed in the condition file is the dependent variable, whether it is group (1), (10), (22) or any other coded group. We will discuss this further with reference to the No Recode function below.

Figure 7: GOLDVARB 2001 window showing the log of a coding error

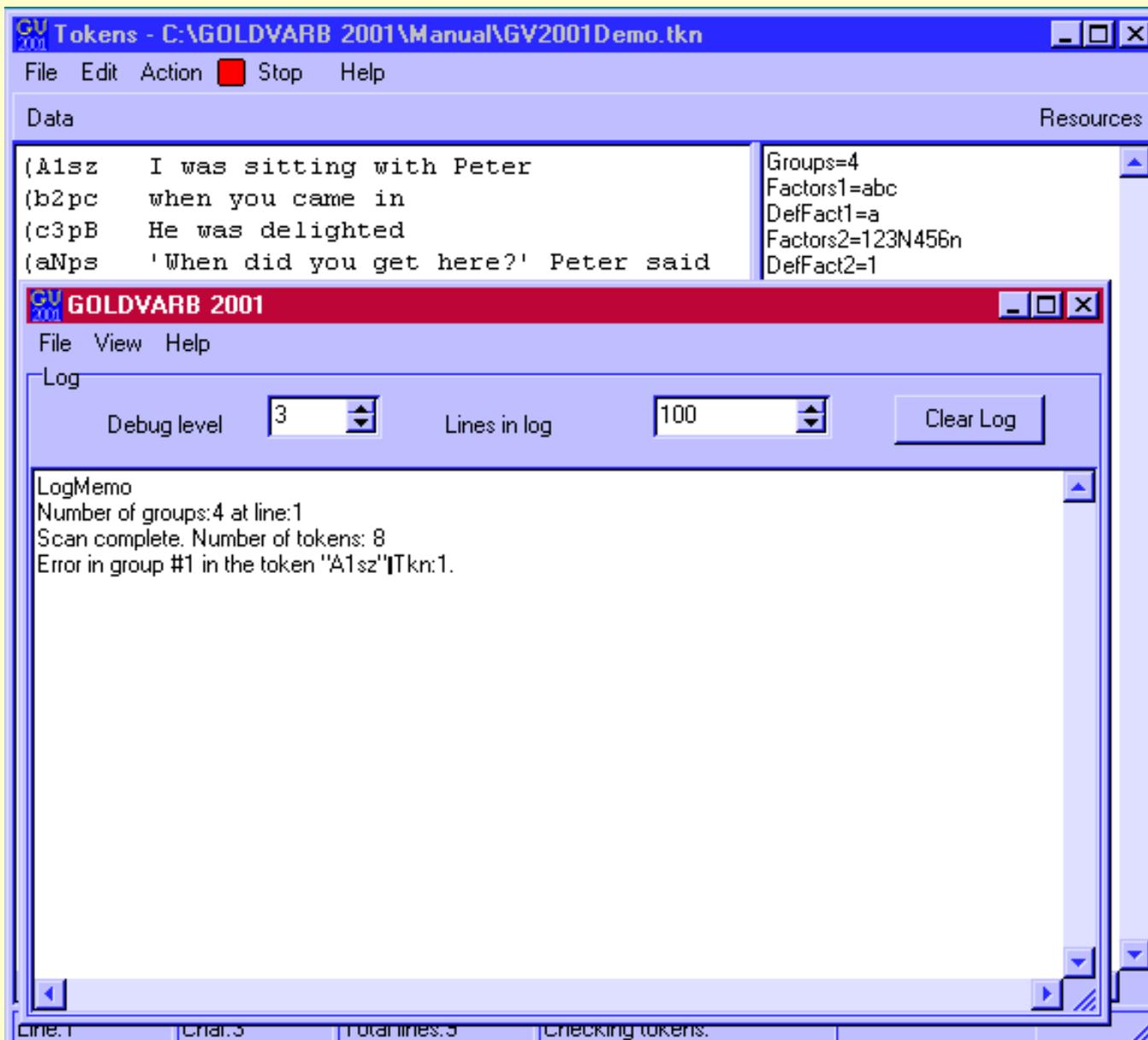


Figure 7 shows the token file, the specified factors in the Resources frame and the error message in the GOLDVARB 2001 window. The figure illustrates a coding error. The Resources window shows that the only characters specified for use in factor group 1 are **a** & **b**. The first token, however, contains an upper case A in group 1. The message in the GOLDVARB 2001 window identifies this error by listing the precise character which is wrong (group #1, Token 1).

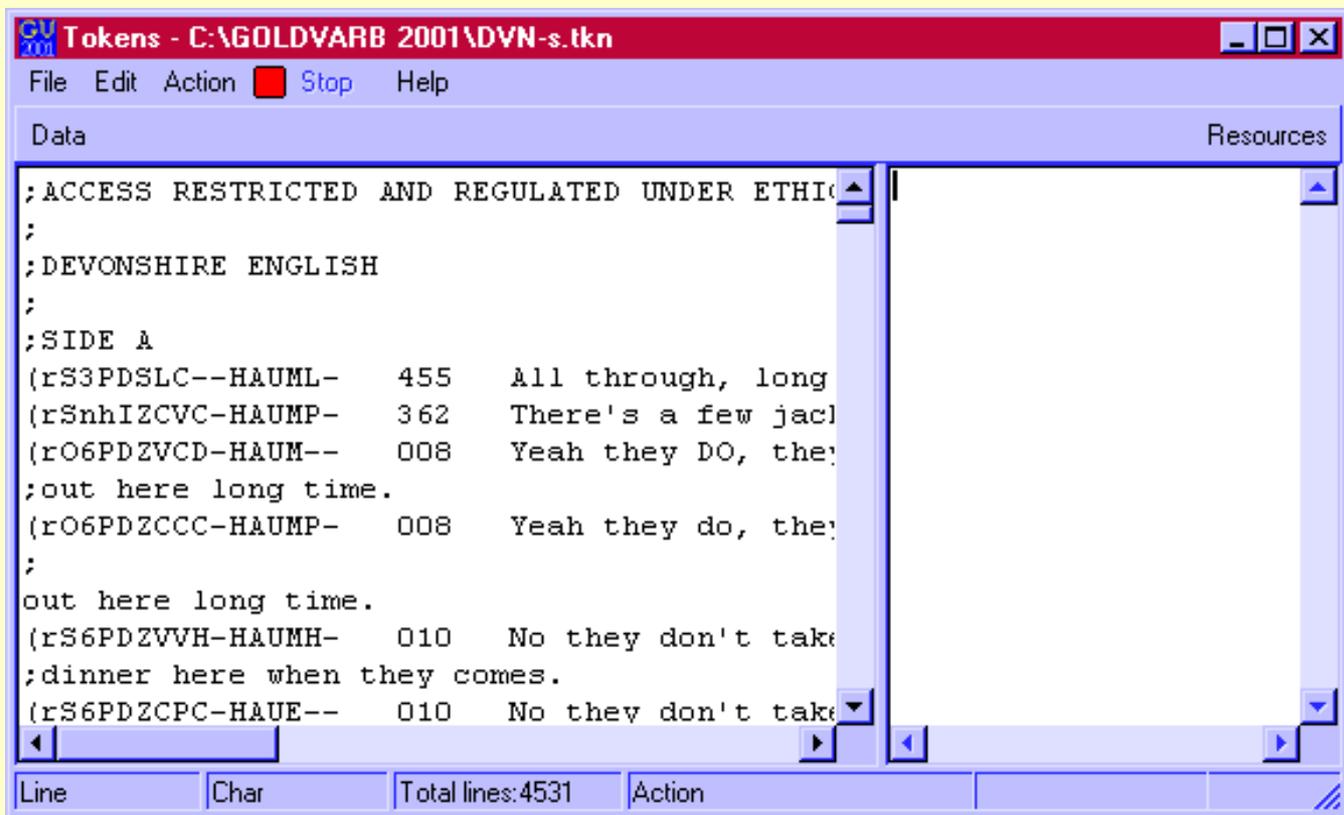
On opening an existing file, GOLDVARB 2001 will regenerate factor specifications (Tokens window, Action|Generate Factor Spec's) by scanning existing factors, therefore it is important to ensure that coding is accurate.

[Back to table of contents](#)

Opening an existing token file.

In the Token window, Click on the File menu. Recently used token files will appear beneath the menu. If the file you want is listed, simply select. If not, select Open and find the file your are interested in through the dialog box. Say the file you chose was called DVN-s. The Tokens window will now look like Figure 8:

Figure 8: Existing token file



[Back to table of contents](#)

Using the data

At this stage you can scroll up and down and side to side to look at the data in the file. You can also alter the size of the window to suit you and increase the frame-size of the Data frame by clicking on the bar dividing the Data from Resources and dragging it to the right.

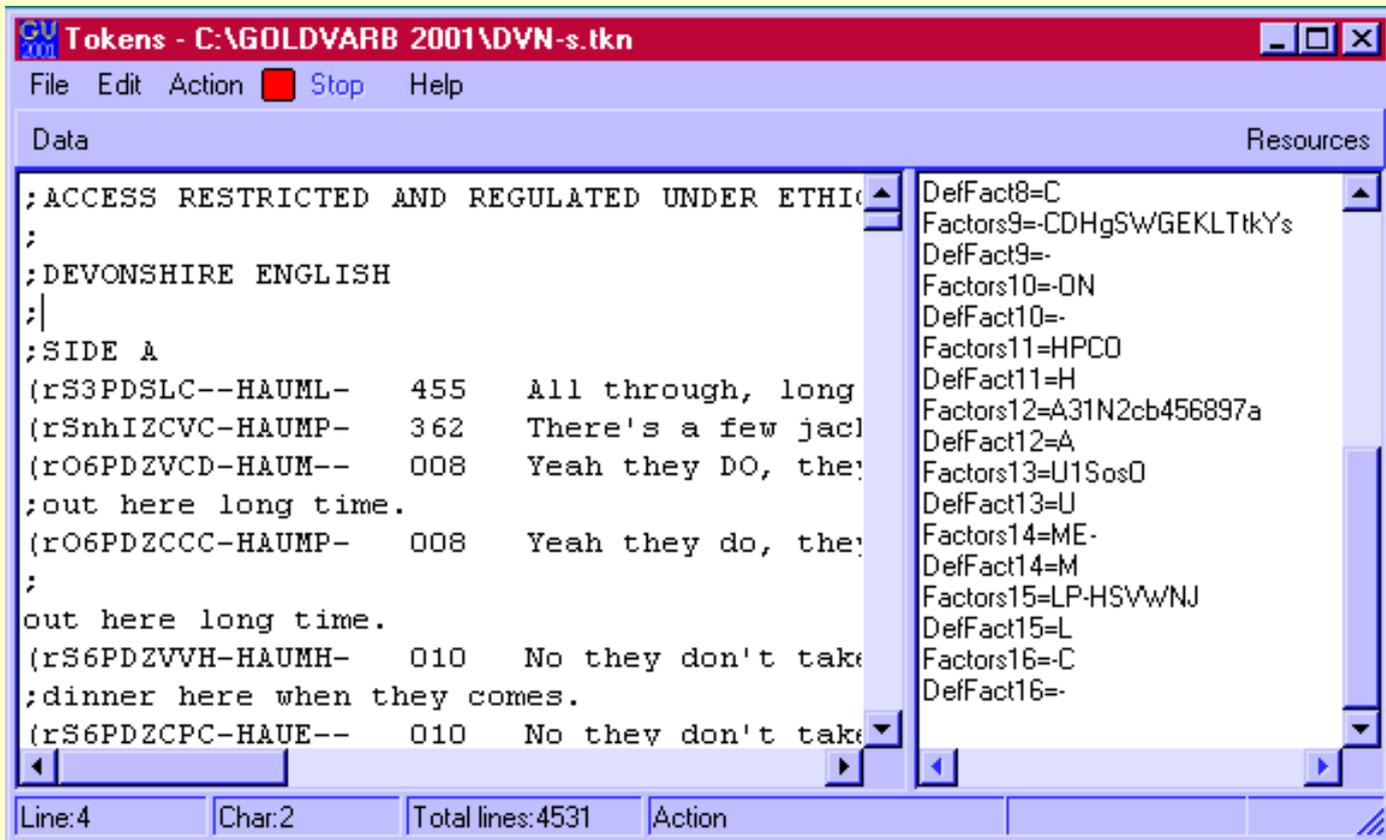
Before anything else can be done, the program must recognise what is in the string of characters to the left of the screen. In order for this to happen, the factors valid for each group must be specified. If this has already been done, the factor specifications will be visible in the Resources frame of the Token window. If not, follow the steps below to specify factors.



From the Action menu in the Tokens window, select Generate Factor Spec's. Then click OK in the dialogue box which appears. The program then scans the factor specifications and a dialog box appears asking for confirmation of the factor specifications. Click OK.

GOLDVARB 2001 then logs the factor specifications in the right hand frame of the Tokens window, as shown in Figure 9. **Save** the file.

Figure 9: Token file with factor specifications



[Back to table of contents](#)

No Recode

In order to perform an overall distributional analysis of each of the factors in each factor group according to the dependent variable, use the **No Recode** function.

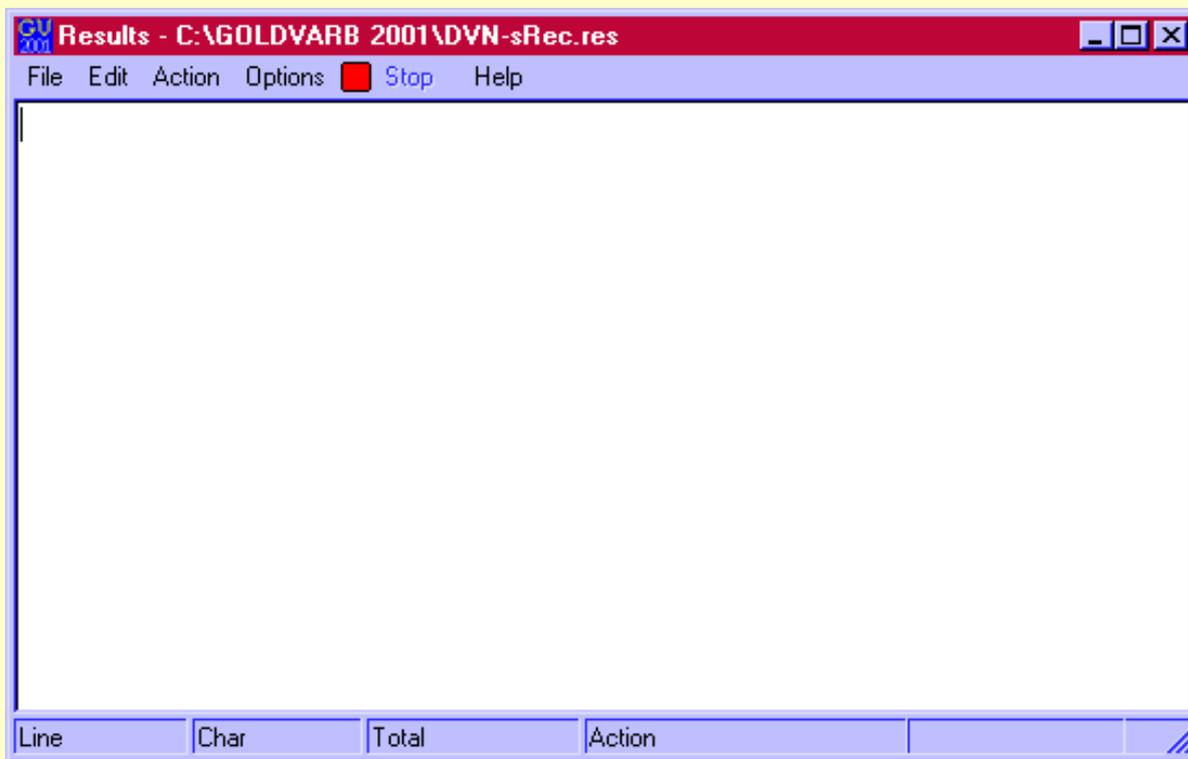


Under the Action menu of the Tokens window, select No Recode.

Then go to the GOLDVARB 2001 window, and under the View menu, select Results.

An empty Results window will appear, which will look like Figure 10:

Figure 10: New Results window



Save the file using the SaveAs from the File menu. DVN-sNoRec has been used here, to indicate the token file to which it relates and the fact that the file has been created with no recoding.

Go to the Action menu of this window, and select Load cells to memory. Click OK on the dialog boxes which come up, which simply ask for confirmation of the factor specifications and the fill character, and then for the **application value**.

The program assumes that the dependent variable appears first in the coding string, so in a No Recode the application value will be the first factor. There are restrictions on the number of factors in the factor group which is the dependent variable - no more than 9 factors for a distributional analysis. Further, GOLDVARB 2001 allows only binomial varbrul analysis, therefore the application value must contain no more than two factors. If, returning to the previous example, you enter \emptyset as the application value, GOLDVARB 2001 will provide the results of the occurrence of \emptyset versus everything else. If you enter \emptyset and **S**, it will provide results of \emptyset versus **S**, to the exclusion of other tokens, say **s** for a contracted copula (e. g. He's clever).

The new results file will look like Figure 11:

Figure 11: No Recode results

Results - C:\GOLDVARB 2001\DVN-sNoRec.res

File Edit Action Options Stop Help

```

CELL CREATION
=====
Name of token file: C:\GOLDVARB 2001\DVN-s.tkn
Name of condition file: C:\GOLDVARB 2001\DVN-sNoRec.cnd
(
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
)
Number of cells: 1089
Application value(s): rbijdpyn
Total no. of factors: 94

```

Group	r	b	i	j	d	p	y	n	Total	%	
1 (2)											
S	N	129	50	132	45	30	36	39	43	504	40
	%	25	9	26	8	5	7	7	8		

Line Char Total lines:375

At the top of the file is the name of the token file from which the marginal results were created and condition file used to create the marginals. Then the factor groups included in the results are listed. These appear in the same format as in the Condition file, excluding any comment lines which may have been inserted.

Next is the total number of cells (the sum of each individual factor being compared against the application values), the application value and the total number of factors in all groups.

Below these notes are the marginal results, with the application values across the top and then a list of all the factors and corresponding numbers and percentages underneath. The total number of each factor (Total) and the proportion (%) it represents of the total number of tokens are shown in the last two columns. The factor groups are indicated by both their original number in parentheses (i.e. the column in which they appear in the token file) and their new number (i.e. the order in which they appear in the condition and results file). The first factor group in the condition file is always treated as the dependent variable. Thus in Figure 13, the first factor group for which results appear is listed as 1 (2) - indicating that it is the first factor group to appear (after the application value) in the condition file, but that it refers to the second factor group (and, therefore, second column) in the token file.

[Back to table of contents](#)

Recoding

When you look at the results provided by the No Recode function, you will often find that there are elements which are either uninteresting, or misleading, or that result in very small numbers in each cell. For whatever reason, most often you will want to recode this configuration, whether to collapse factors together, combine factor groups or remove factors altogether. This can be done in two ways, the first way being the Recode Setup.

Before you begin, you will need to have opened a token file and generated factor specifications as previously described.

★ Select Recode Setup from the Action menu in the Tokens window.

This will provide you with an interactive window looking like Figure 12:

Figure 12: Recode Definition window

Recode definition

Groups in token file

Group							
1	a	b	c				
2	1	2	3	N	4	5	6 n
3	s	p	+				
4	z	c	B	s	l	d	u w

Groups after recoding

Type	Old						

>> Copy >>
Exclude
>> Recode >>
>> And >>
>> Or >>
Remove <<
Modify

Current operation: Idle Finish current operation

Click Ok to write new conditions file from these recode instructions.

Ok - write new conditions from recode instructions Cancel Clear all recoded groups

First, decide which factor group you want to be your dependent variable (and which of its variants the application value). Select this group by clicking on its number on the left-hand frame (e.g. Group 1 or 2 etc.).

Decide if you want the group to appear as it is coded or if you want to make changes. If you want to make changes, see below for further instructions. Otherwise, select Copy from the centre panel.

This will show the group you have selected on the right-hand frame of the Recode definition window (see Figure 13).

For example, having coded each different grammatical person separately, you may want to view 3rd person subjects as opposed to non-3rd person subjects. To collapse factors together, (i.e. consider several factors as one) use the Recode function.

★ Select the group whose factors you wish to change by clicking on the appropriate number on the panel to the left of the window. Click Recode in the centre panel.

You will then see the group that you have selected on the right-hand frame. The original codes appear on the bottom line, recoded values on the top. Select each value individually as shown below and key in the value which you require. This does not have to be one of the original codes, but can be any character you wish.

Figure 13: Recoding a factor group

CV 2001 Recode definition

Groups in token file

Group									
1	a	b	c						
2	1	2	3	N	4	5	6	n	
3	s	p	+						
4	z	c	B	s	l	d	u	w	

>> Copy >>
Exclude
>> Recode >>
>> And >>
>> Or >>
Remove <<
Modify

Groups after recoding

Type	Old								
	group	1							
Recode	2	o	o	3	3	o	o	3	3
	2	1	2	3	N	4	5	6	n
Copy	3	s	p	+					
Recode	4	n	n	s	n	n	n	s	w
	4	z	c	B	s	l	d	u	w

Current operation: Recode Finish current operation

Click Ok to write new conditions file from these recode instructions.
Select and modify factor values in new group.

Ok - write new conditions from recode instructions Cancel Clear all recoded groups

These recoded character can be checked and changed by simply selecting and keying in new character. When you are satisfied with the recoded group, click Finish current operation.

Repeat for other factor groups as desired.

[Back to table of contents](#)

Excluding

There will be times when you wish to exclude a particular group of tokens from your analysis. For example, if you have coded your data for the individual speakers who have produced each token, you may wish to consider the youngest speakers only in your analysis and so need to exclude the others.

This is can be done easily through the Recode definition window.

★ Select the group from which you want to exclude a factor on the left hand side of the window. Click Exclude.

In the left hand frame of the window, select the factor or factors which you want to exclude from the analysis by simply clicking on the boxes in which the characters appear. The selected characters will then appear in the right-hand frame. When you are satisfied that they are all correct, click Finish Current Operation.

When you are satisfied with the recode setup you have created, click the button at the bottom left of the window labelled OK - write new conditions from recode instructions.

There are other functions available in the Recode window, which are discussed later.

[Back to table of contents](#)

The Conditions (.Cnd) file

When you have completed your recode definition, GOLDVARB 2001 will create Conditions file - the instructions to the program which will be used to create the results. These are presented in a form known as LISP syntax. Each item is crucial for the program to interpret your requirements, so take note of every parenthesis, slash and semi-colon!

A Condition file will look something like that shown at Figure 14. Note the recode for Factor Group 1 (at the top of the file). Anything coded 'c' in Factor Group 1 has been excluded from the whole analysis - the program will ignore any token which has **c** in Column 1. Note the recode for Factor Group 2, next in the file. Here, tokens coded '1', '2', '4' and '5' in Column 2 (all non-3rd person subjects) will be shown in the results with the character 'o'. Tokens coded '3', 'N', '6' or 'n' in the same column (3rd person subjects) will be shown as '3'. There is no recoding of Factor Group 3, which will therefore appear in the results file with no changes. Note the recode of Factor Group 4, however. In a similar way to Factor Group 2, the factors have been recoded to appear in the results file signified by different characters. Tokens which are coded 'z', 'c', 's', 'l' and 'd' in Column 4 will appear in the results file represented by an 'n'. The other tokens, those coded 'B', 'u' and 'w' will appear as 's'.

Figure 14: Condition file

The screenshot shows a text editor window titled "Conditions - C:\GOLDVARB 2001\Manual\GV2001Demo.cnd". The window contains the following LISP-style code:

```
(
(1 (NIL (COL 1 c)))
(2 (o (COL 2 1))
   (o (COL 2 2))
   (3 (COL 2 3))
   (3 (COL 2 N))
   (o (COL 2 4))
   (o (COL 2 5))
   (3 (COL 2 6))
   (3 (COL 2 n)))
(3)
(4 (n (COL 4 z))
   (n (COL 4 c))
   (s (COL 4 B))
   (n (COL 4 s))
   (n (COL 4 l))
   (n (COL 4 d))
   (s (COL 4 u))
   (s (COL 4 w)))
)
```

The status bar at the bottom of the window displays "Line:20", "Char:1", "Total lines:21", and "Action".

Exclusion from a single group

At present there is one important function which can **only** be achieved through editing the Condition file directly. There may be times when you do not wish to exclude a factor from your entire analysis, but you do want to exclude it from the results of a given factor group. For example, you may decide that, while you do not want to remove tokens with non-third person subjects from the analysis, you do not want to consider them amongst other types of pronoun in the Subject Type factor group.

★ To do this, go to the factor group from which you want to exclude the factor. Inside the parentheses, insert a left parenthesis (, then forward slash /, followed by **(COL X y)** and finally a right parenthesis - where **X** stands for the number of the factor group, (i.e. the column where the factor to be excluded can be found in the token file) and where **y** represents the factor to be excluded.

So if Factor N is to be excluded from consideration in Factor Group 2, the line in the Condition file will read as

```
(2 (/ (COL 2 N)))
```

Thus, a condition file which has been edited in this way will look something like Figure 15:

Always make sure that, when looking at the instructions for a factor group, that there are the same number of parentheses on the right as there are on the left!

Figure 15: Edited condition file

```
(
(1 (NIL (COL 1 c)))
(2
;remove non-3rd person subjects
(/ (COL 2 1))
(/ (COL 2 2))
(/ (COL 2 4))
(/ (COL 2 5)))
)|
```

Note - **NIL** indicates that a given factor will be excluded from the whole analysis. / indicates that the factor will be excluded from an individual factor group. However, if that factor group is the application value, or dependent variable, the effect is that of exclusion from the whole analysis. In other words, regardless of where **NIL** appears, the factor will be excluded altogether. If / appears in the dependent variable, the effect will be the same. But if / appears anywhere else in the condition file, the factor will merely be excluded from the factor group in which it appears.

[Back to table of contents](#)

Results

★ Look at the results by returning to the GOLDVARB 2001 window, selecting Results from the View menu, then selecting Load cells to memory from the Action menu in the Results window.

If this is the first time you have looked at results since opening the application, the Results window will be blank. However, GOLDVARB 2001 retains results in the file, so each new set of results appears underneath existing ones. So the second time you select View|Results, you will see the Results window with previously loaded cells. This makes no difference to the functioning of the window, simply follow the steps in the same way, and your new results will be appended at the bottom of the file.

Check that you have excluded the right factors from the intended part of your analysis by checking the results. If you have excluded one factor from the whole analysis, each sub-total should be the same as the total at the bottom of the file. There will be no sign of the factor which you excluded. If you have excluded one factor from a single factor group, the remainder of factors from that group will be in evidence, but there will be no sign of the factor you have excluded and the subtotal at the bottom of the results for that factor group will be smaller than the total at the bottom of the file (the difference being the total number of tokens of the factor you have excluded).

[Back to table of contents](#)

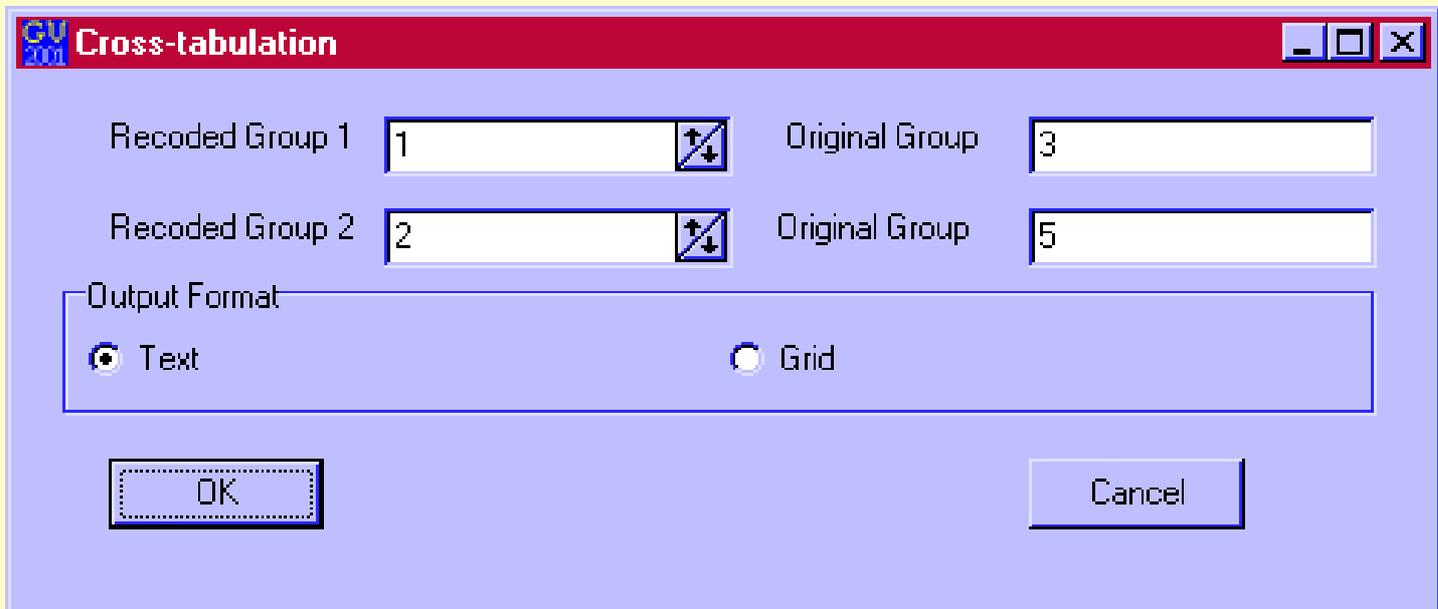
Cross-tabulations

In order to see the intersection of a given factor group with another, use the **cross tabulation** function to set the results against each other.

★ In the Results window, select Cross-tabulation from the Action menu.

A box appears which looks like Figure 16:

Figure 16: Cross-tabulation dialog box

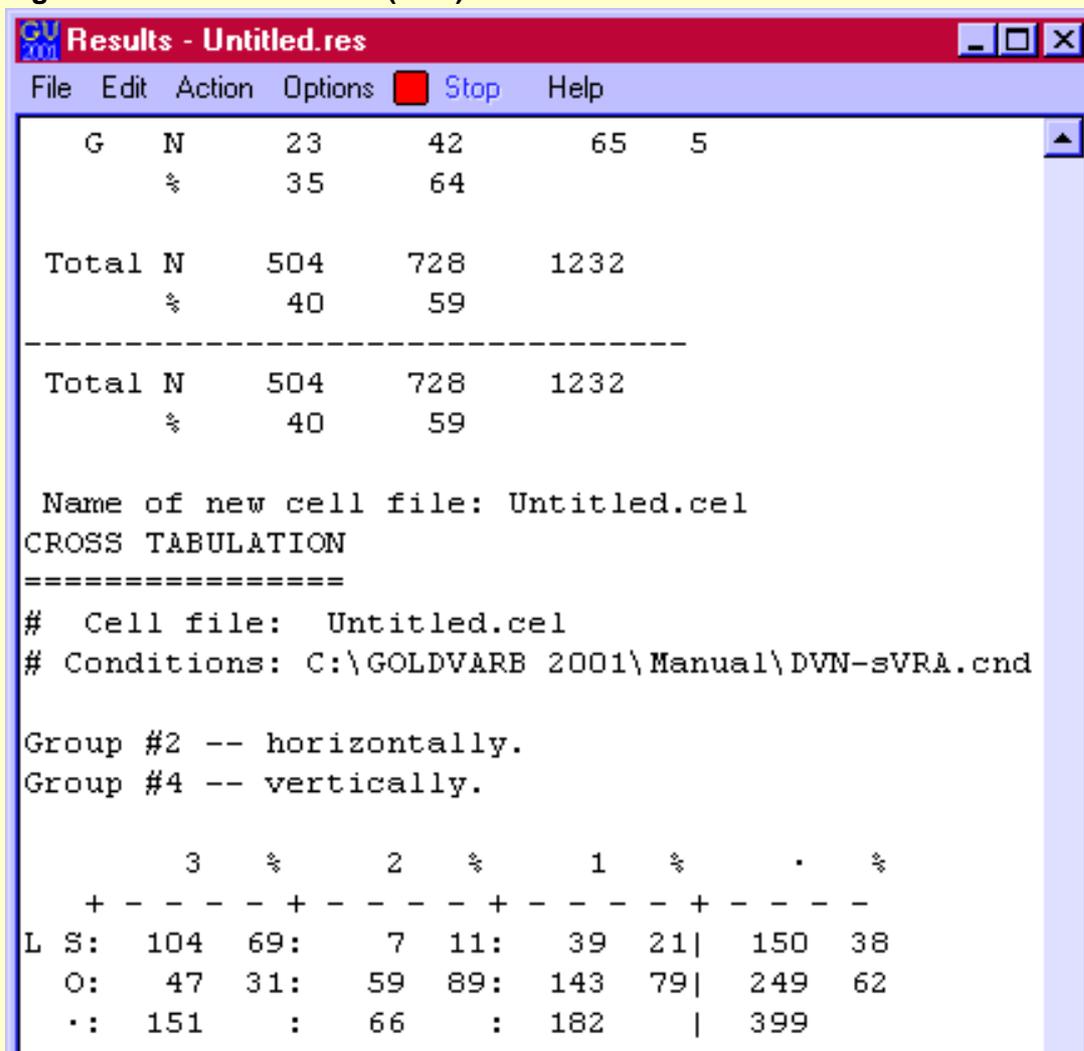


Click on the up/down arrows to the left of the boxes in order to select the factor groups which you wish to set against one another.

There are two options for how you would like to see your cross-tabulation: Text or Grid. Select one of these, then click OK to see the cross-tabulation.

The first option is to select Text, as shown above, so that the cross-tabulation appears at bottom of Results file and looks like Figure 19.

Figure 17: Cross-tabulation (Text)



```

  .: 151      : 66      : 182      | 399
  + - - - - + - - - - + - - - - + - - - -
C S: 117 57:   4 10:   12 16| 133 42
O:  88 43:  35 90:  63 84| 186 58
  .: 205      : 39      : 75      | 319
  + - - - - + - - - - + - - - - + - - - -
V S: 149 63:  18 22:  47 29| 214 45
O:  89 37:  64 78: 113 71| 266 55
  .: 238      : 82      : 160     | 480
  + - - - - + - - - - + - - - - + - - - -
S S:   5 42:   2 14:   0  0|   7 21
O:   7 58:  12 86:   8 100|  27 79
  .:  12      :  14      :   8      |   34
  +-----+-----+-----+-----+
  S: 375 62:  31 15:  98 23| 504 41
O: 231 38: 170 85: 327 77| 728 59
  .: 606      : 201     : 425     | 1232
  
```

Line Char Total lines:135

GOLDVARB 2001 provides an alternative view, Grid, which results in a separate window looking like Figure 18:

Figure 18: Cross-tabulation (Grid)

Cross tabulation results

File Edit View

Source Cells file Conditions file

Group	1	m	m	f	f	Total	Total
2	App Value	Count	%	Count	%	Count	%
D	S	375	47	35	33	410	45
D	O	430	53	71	67	501	55
D	Total	805		106		911	
I	S	84	28	10	62	94	29
I	O	221	72	6	38	227	71
I	Total	305		16		321	
Total	S	459	41	45	37	504	41
Total	O	651	59	77	63	728	59
Total	Total	1110		122		1232	

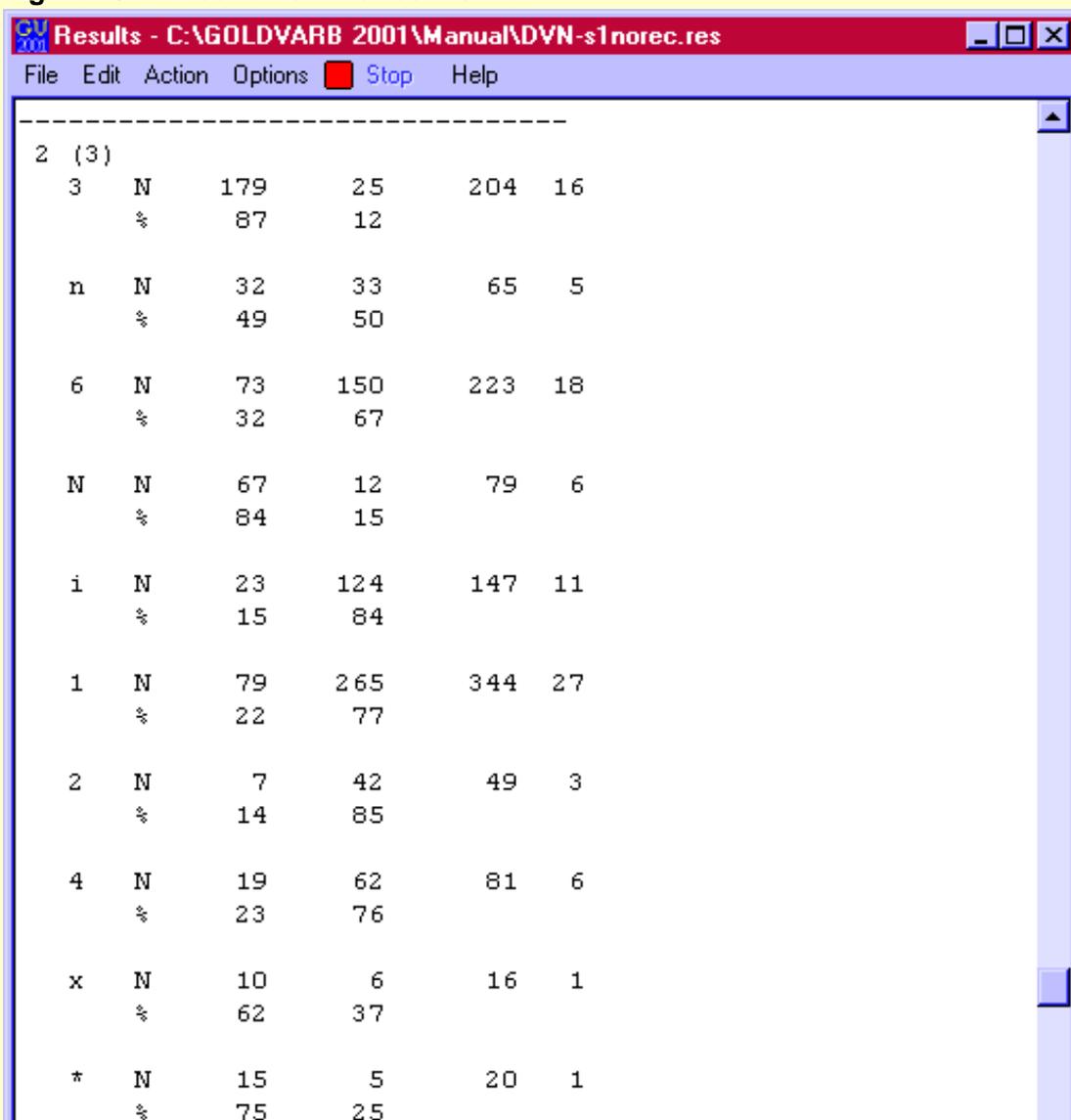
This grid, permits the user to see the patterns of distribution by colour, where darker colours indicate an increased percentage. The closer the colours in opposing cells, the smaller the difference between them; and conversely, the greater the contrast in colour, the greater the distinction. This grid cross-tabulation also has options to copy the data into other applications (e.g. text processing or spreadsheet packages).

[Back to table of contents](#)

Multivariate Analysis

The multiple regression function of GOLDVARB 2001 will run when you have achieved a set of results with a binary application value (i.e. no more than two factors specified for the dependent variable) which have variation in every factor. This means that this function will NOT run if there are any *knockouts* or *singleton groups* in the results file. A knockout is where all the tokens are accounted for by one or the other of the application values. For example, if we are looking at the occurrence of plural -s, it may be that in a given area of the grammar (2nd person plural in the Figure 18) there are no instances of plural -s. The results file shows that occurrence of plural -s is 0% and absence is 100%, there is no variation in this environment and the multiple regression analysis will not run.

Figure 19: "Knockouts" in a results file



The screenshot shows a window titled "Results - C:\GOLDVARB 2001\Manual\DVN-s1norec.res". The window contains a table with the following data:

Application Value	Category	Count	Percentage	Count	Percentage
2 (3)	N	179	25	204	16
	%	87	12		
n	N	32	33	65	5
	%	49	50		
6	N	73	150	223	18
	%	32	67		
N	N	67	12	79	6
	%	84	15		
i	N	23	124	147	11
	%	15	84		
1	N	79	265	344	27
	%	22	77		
2	N	7	42	49	3
	%	14	85		
4	N	19	62	81	6
	%	23	76		
x	N	10	6	16	1
	%	62	37		
*	N	15	5	20	1
	%	75	25		

*	N	15	5	20	1
	%	75	25		
5	N	0	4	4	0
	%	0	100	* KnockOut *	
Total	N	504	728	1232	
	%	40	59		

Line:855 Char:35 Total lines:1147

Once you have honed your analysis by collapsing and/or excluding factors and factor groups, you are ready to perform a multiple regression analysis of your data.

★ In the Results window, select Binomial Up and Down from the Action menu. If you get an error message saying "Warning - bad cells", this simply means that there is at least one 'knockout' in the results file which is preventing the binomial analysis from running.

This will open a new window called Binomial Varbrul, which shows the analysis, which looks like Figure 20:

Figure 20: Binomial Varbrul window

The screenshot shows the Binomial Varbrul window with the following content:

```

Binomial Varbrul
=====
Name of cell file: Untitled.cel

Using fast, less accurate method.
Averaging by weighting factors.
Threshold, step-up/down: 0.050001

# Stepping up:
# Stepping up:

----- Level # 0 -----

Run # 1, 1 cells:
Convergence at Iteration 2
Input 0.409
Log likelihood = -833.480

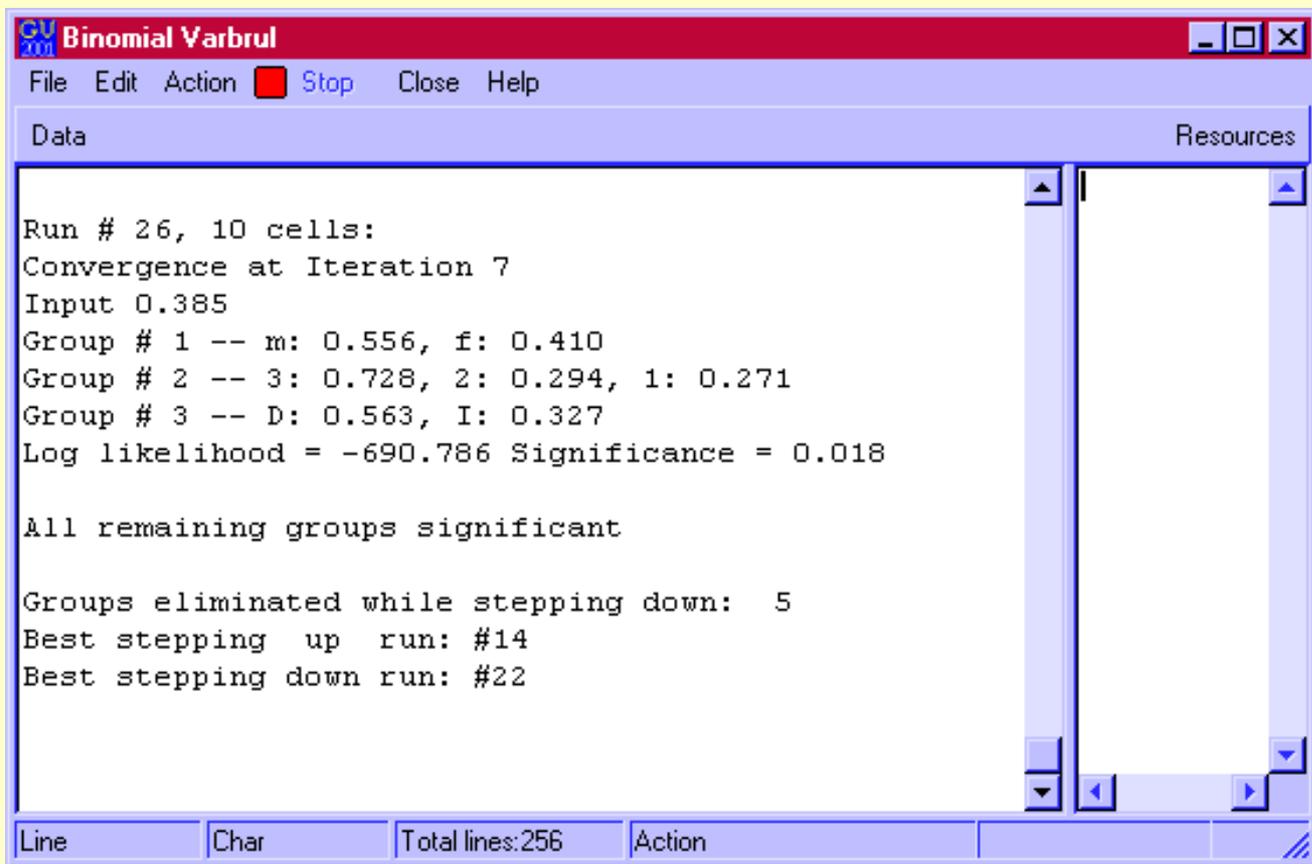
----- Level # 1 -----

Run # 2, 2 cells:
Convergence at Iteration 4
Input 0.409
Group # 1 -- m: 0.505, f: 0.458
Log likelihood = -833.022 Significance = 0.351

```

Line:1 Char:1 Total lines:114 Action

Figure 21: End of Binomial Varbrul



The end of the analysis, shown in Figure 21, shows the best stepping up and down runs. These provide the best fit of the model to the data. The best stepping up and best stepping down runs ideally contain the same factor groups and associated factor weights. If they are not identical, the statistical significance of the factor group that differs from the step-up and step-down analysis is questionable. It is therefore wise to check the results in more detail. Cross-tabulation of factor groups can sometimes reveal "interaction", or areas where a factor in one group is the same (or close to the same) as a factor in another.

The procedures involved here are thoroughly documented in the manual for GoldVarb 2.0 (Rand & Sankoff 1990).

Figure 22: Best stepping-up run

```

Binomial Varbrul
File Edit Action Stop Close Help
Data Resources
Add Group # 1 with factors mf
----- Level # 4 -----
Run # 14, 39 cells:
Convergence at Iteration 7
Input 0.383
Group # 1 -- m: 0.554, f: 0.412
Group # 2 -- 3: 0.733, 2: 0.293, 1: 0.264
Group # 3 -- D: 0.564, I: 0.326
Group # 4 -- L: 0.517, C: 0.439, V: 0.542, S: 0.294
Log likelihood = -685.712 Significance = 0.018

Run # 15, 39 cells:
Convergence at Iteration 8
Input 0.385
Group # 1 -- m: 0.556, f: 0.409
Line:1 Char:1 Total lines:256 Action

```

Figure 22 shows the best stepping-up run for the example analysis, in this case Run #14). The overall probability that the dependent variable will occur, the input value, is .38 (rounded to two decimal places). The run shows that the four factors which were included in the analysis (Groups 1,2,3 and 4) have a statistically significant effect ($p < 0.05$) on the occurrence of the dependent variable.

There is a significant difference between the occurrence of the dependent variable according to whether **m** or **f** appears in factor group 1. The factor weights show that the dependent variable is favoured in tokens coded **m**, at .55, and disfavoured in tokens coded **f** at .41.

The relative strength of each statistically significant effect can be measured by subtracting the lowest value from the highest - in group 1, the range would be 14. The range is 47 for group 2, 23 for group 3 and 25 for group 4, showing that the strength of factors from strongest effect to weakest would be: Group 2 > Group 4 > Group 3 > Group 1.

The figure also shows the log likelihood and statistical significance of the run. The log likelihood figure is a measure of how well the particular analysis fits the data, with figures closer to 0 being better. This is primarily useful for comparing different analyses of the data, i.e. the inclusion of different groups, factors being recoded in different ways and so on.

[Back to table of contents](#)

Other Functions

The instructions above will have been sufficient to take you through the basics for what you need to know about GOLDVARB 2001. However, there are other functions of the application which you may find useful to know about. One of these - find & replace in two or more non-adjacent factor groups at once - is a welcome addition to GoldVarb 2.0.

[Back to table of contents](#)

Find and Replace

GOLDVARB 2001 has three features within the Find function.

Finding text

★ GOLDVARB 2001 allows you to search not only the factors but also the text of your data, so you can search for particular words or phrases in the data. From the Edit menu in the Tokens window, select Goto/Find Text. A dialog box will appear, as shown below.

Simply use Goto to go to a particular line number (shown in the bottom left corner of the window), or enter in the text you wish to search for.

Figure 23: Find text

Finding Factors

★ In the Tokens window, select Find and Replace from the Edit menu. You will see a box which looks like Figure 24.

Figure 24: Find factors

If you want to search for a particular factor simply select the appropriate factor group and the factor you require from the options which will appear. The options include the 'fill' character, so you can also search for tokens which are uncoded in a given factor group. GOLDVARB 2001 also allows you to search for factors from two or more factor groups at once by checking the "and" box and repeating the procedure. So it is possible to find, for example, a token which has a 3rd person

subject (coded as **N** in factor group 3) in a narrative clause (coded as **Z** in factor group 6) simply by selecting the appropriate factors for each of the factor groups. You may also specify a series of characters as shown in Figure 25 for factor group 8.

Click on Next to perform the search.

Figure 25: Finding factors in more than one factor group

Find and Replace in Factors in Groups

Find

In Group 3 any of N and

In Group 6 any of Z and

In Group 8 any of CVP and

Replace

In Group 0 set factor to Confirm

Next All Cancel

GOLDVARB 2001 will then find the first token which matches the criteria you have specified, which will then be highlighted as shown in Figure 26.

Figure 26: Token identified through 'Find' function

Tokens - C:\GOLDVARB 2001\DVN-s.tkn

File Edit Action ■ Stop Help

Data	Resources
;quite a lot.	Groups=16
(rS1PDZCVS-HAURL- 204 Oh she rings me up	Factors1=rbi
;quite a lot.	DefFact1=r
(rO3+DZCCW-C1UEN- 205 That's the only one	Factors2=SC
;you know.	DefFact2=S
(rS1PNDZCVS-C1UMP- 208 The bill soon RUNS	Factors3=3n
(rS3PDZCCW-C1UMS- 211 I don't know, it al	DefFact3=3
;older innit.	Factors4=Pr
(rS1PDSLvg-CA1ES- 211 I don't know, it al	DefFact4=P
	Factors5=Dl
	DefFact5=D

Line:81 Char:1 Total lines:4532 Checking tokens: ■

The highlighted coding string in Figure 26 shows that, as specified by the Find function in Figure 25, the token has been coded as **N** in Group 3, **Z** in Group 6 and has one of the alternatives **C**, **V** or **P** in Group 8.

Replacing Factors

★ Should you wish to replace a factor with another, simply enter the required factor into the box in the centre section at the bottom of the Find and Replace dialog box. If you decide that if a token is coded A in Group 4 and w in Group 6, you want it to be Y in Group 10, your Find and Replace window be as Figure 27:

Figure 27: Find and Replace function

The screenshot shows a dialog box titled "Find and Replace in Factors in Groups". It is divided into two main sections: "Find" and "Replace".

Find Section:

- Row 1: "In Group" dropdown set to "4", "any of" text, a text box containing "P", and a checked "and" checkbox.
- Row 2: "In Group" dropdown set to "6", "any of" text, a text box containing "E", and an unchecked "and" checkbox.

Replace Section:

- "In Group" dropdown set to "10", "set factor to" text, a text box containing "N", and a checked "Confirm" checkbox.

Buttons: "Next", "All", and "Cancel" are located at the bottom of the dialog.

When you are ready to search, click Next to find and replace matching tokens one by one, or All to change all tokens with a single click. Note the 'Confirm' box. This must be selected before the Replace function can be performed.

[Back to table of contents](#)

And/Or

Within the Recode setup, there are two very useful functions which allow you to combine the codes from two factor groups. For example, you may wish to look at distributions of tokens that are **both** 3rd person singular **and** appear in narrative discourse. Alternatively, you may wish to see distributions of tokens which are **either** in a main clause **or** have a pronominal subject. To do this, use the operations And and Or.

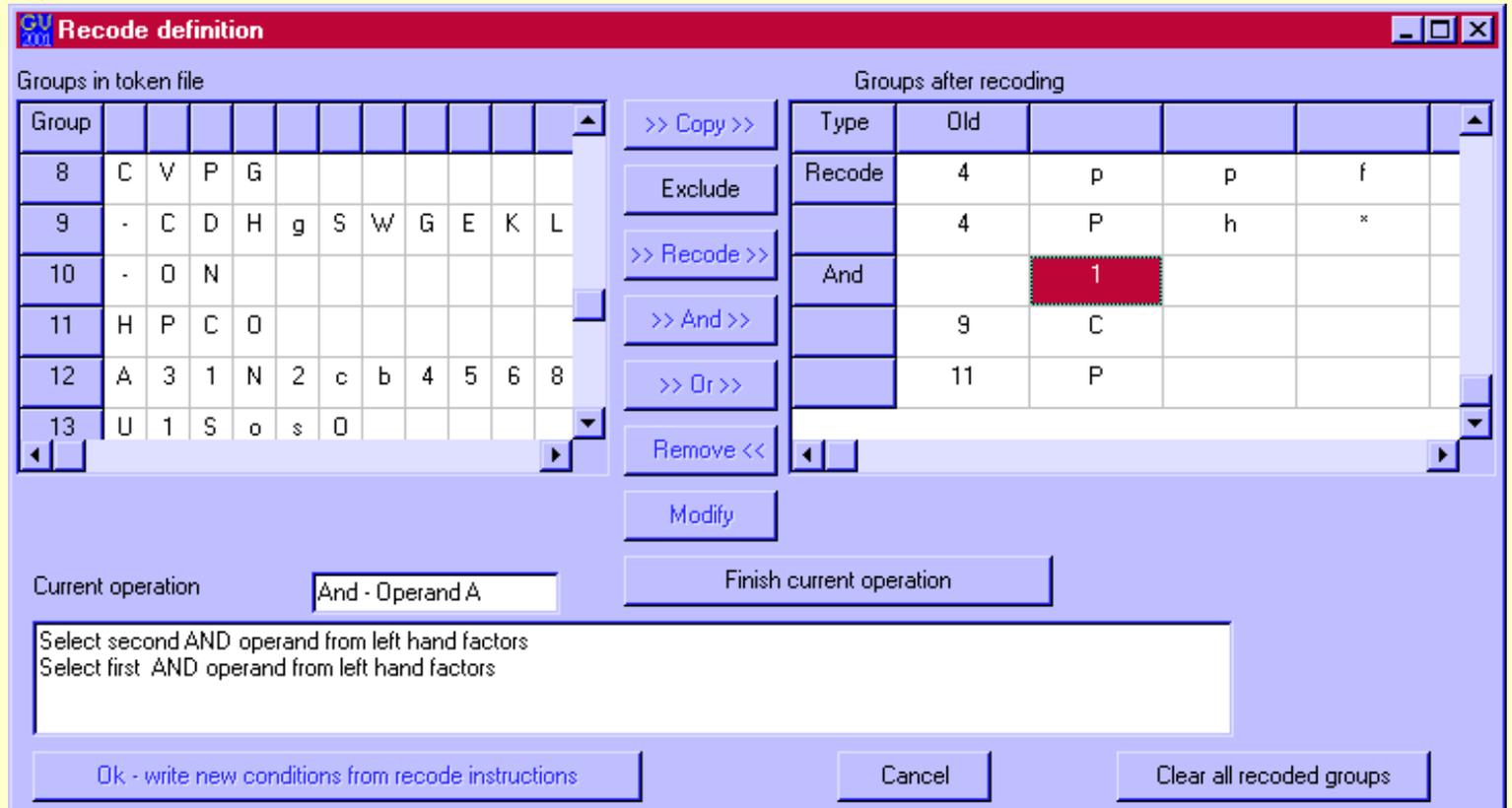
'And' function

★ For example, if you wish to combine **P** from factor group 4 and **Y** from factor group 9. First, open a Recode Definition

window (Tokens window, Action|Recode Setup).

If this is not the first time you have opened this window since entering GOLDVARB 2001, the previously recoded groups will be visible. If you haven't saved the original recode, there will be a warning message indicating this. To proceed without saving, click OK. If you wish to save your previous setup, click Cancel. In the GOLDVARB 2001 window, select View|Conditions. Save this file. Then return to the Tokens window, and open the Recode Definition window again. In either case, the previously recoded group will still be visible. Either edit those groups which are already there using the Modify button, or click Clear all recoded groups to start again.

Figure 28: Recode definition - 'And' function



Select a dependent variable by clicking the number of the required group on the panel on the left of the Recode Definition window. Then click Copy on the centre panel. Select the factor group 9 in the left hand frame. Click And, then select factor group 11.

Select **C** from factor group 9 by clicking on the box in the left-hand frame, then **P** from factor group 11 in the same way. They will appear in the right-hand frame as you select them, as shown in Figure 28.

Select the space above the **C** in the right hand frame, and key in the recoded character - in Figure 28 the recoded character is **1**.

If there are further combined characters you wish to create, repeat the procedure. When you have finished, click Finish Current Operation.

Finally, enter into the box on the right of the window (indicated by the arrow on Figure 29 the character you wish to assign to all tokens from the selected factor groups not included in the specified combinations. This character (**H** in this case) will appear in the far left box, above the selected factor groups as shown in Figure 30.

Figure 29: 'And' function, final operation

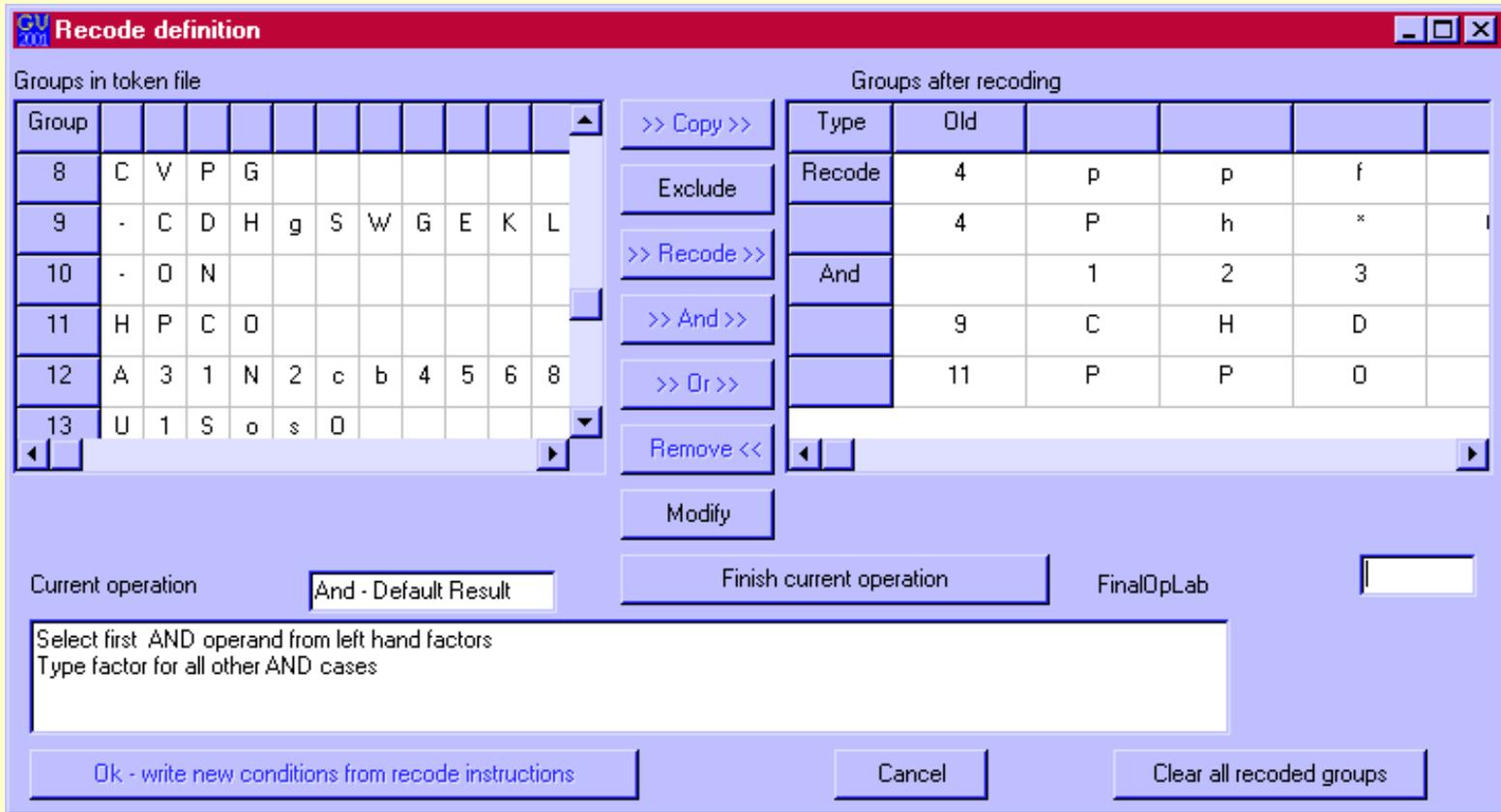
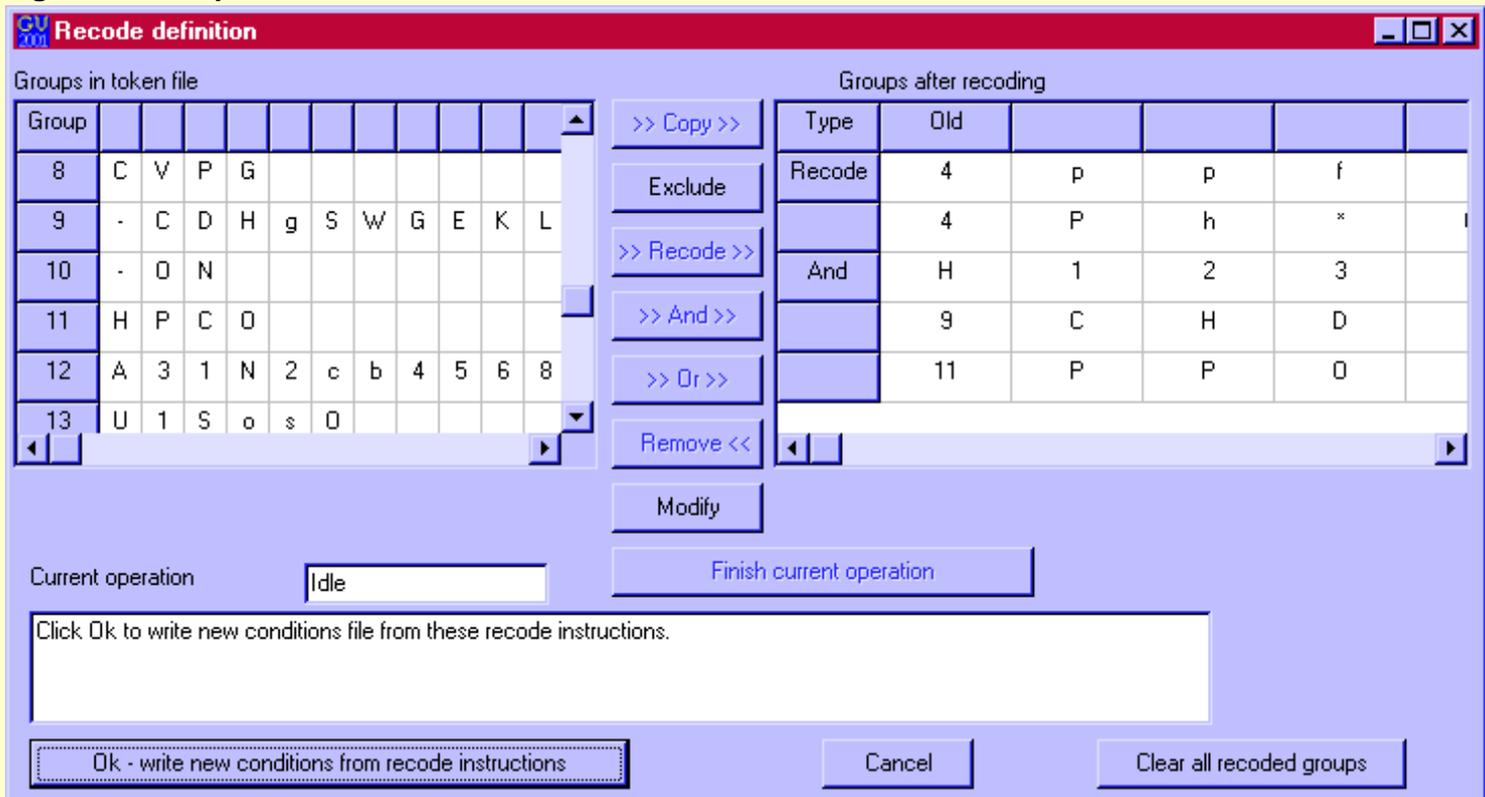


Figure 30: Completed 'And' function

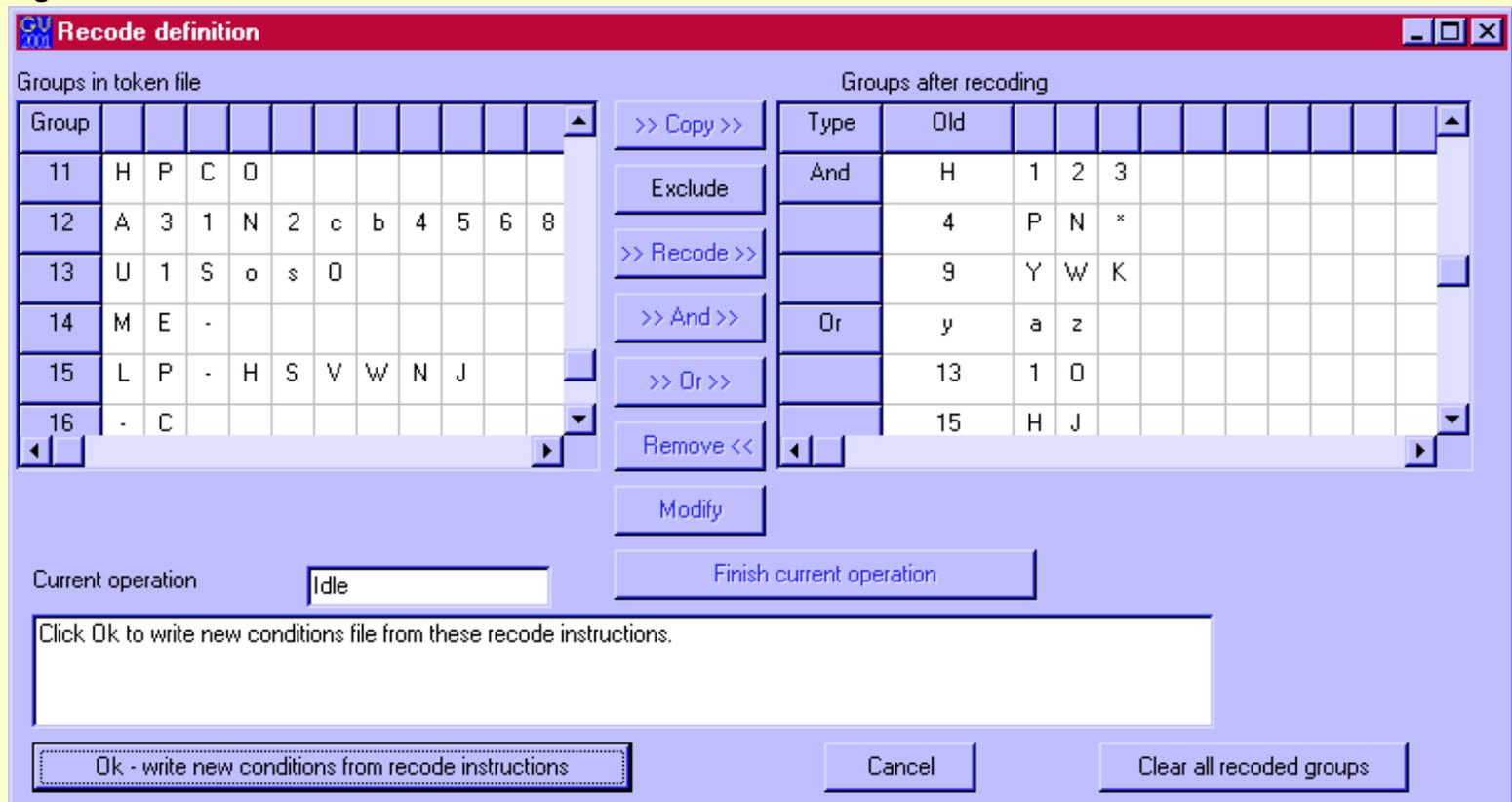


[Back to table of contents](#)

'Or' function.

The 'Or' function works in exactly the same way, simply substituting Or for And. A Recode Definition window with both And and Or functions used, will look like Figure 31:

Figure 31: Recode definition - 'Or' function



[Back to table of contents](#)

Elsewhere

"Elsewhere" is a useful shortcut which can be used in manual recoding (i.e. recoding from within a .cnd file). It means that you can recode to a certain extent, then give a command which says "And anything else that I haven't already declared within this factor group, call X."

Figure 32 shows a Condition file with the Elsewhere function included. This file instructs GOLDVARB 2001 to consider 1, 2, 4, and 5 in Column 2 as o, but then, rather than listing all the other factors to be found in Column 2, it uses the Elsewhere function to instruct the program that anything apart from the listed factors is to be considered as 3, in this case all 3rd person subjects.

Figure 32: 'Elsewhere' function in Conditions window

The screenshot shows a window titled "Conditions - C:\GOLDVARB 2001\Manual\GV2001Demo.cnd". The window has a menu bar with "File", "Edit", "Action", "Stop", and "Help". The main area contains a list of conditions in a text editor. The conditions are as follows:

```
(
(1 (NIL (COL 1 c)))
(2 (o (COL 2 1))
(o (COL 2 2))
(o (COL 2 4))
(o (COL 2 5))
(3 (ELSEWHERE)))
(3)
(4 (n (COL 4 z))
(n (COL 4 c))
(s (COL 4 B))
(n (COL 4 s))
(n (COL 4 l))
(n (COL 4 d))
(s (COL 4 u))
(s (COL 4 w)))
)
```

At the bottom of the window, there is a status bar with the following information: "Line:18", "Char:1", "Total lines:18", and "Action".

[Back to table of contents](#)

Further information

Hints

There is a default setting within GOLDVARB 2001 to show "Hints". These are labels which appear when the mouse rests over any active window (apart from the GOLDVARB 2001 window). To switch this setting off, go to the Help menu in each window. There is an item called Show hints which will have a tick (checkmark) by the side. Select Show hints to remove the tick which will remove the labels.

[Back to table of contents](#)

Transferring files from GoldVarb 2.0 into GOLDVARB 2001

Copy your GoldVarb 2.0 file onto a diskette. Put the diskette into your pc and open the file from within the GOLDVARB 2001 application. Be warned - once a file has been transferred into GOLDVARB 2001, it cannot then be opened in GoldVarb 2.0.

In order to successfully transfer a GoldVarb 2.0 token file into GOLDVARB 2001, there are several points which must be taken into account, largely as a result of the fact that GOLDVARB 2001 scans text as well as coding strings.

Parentheses:

GOLDVARB 2001 recognises left parentheses as starting a token, therefore problems will arise if it finds them in other positions. One solution is to replace 'curved' parentheses with 'square' or 'curly' parentheses. A possible way of doing this is to copy your GoldVarb 2.0 file into a text-processing application, and searching for SPACE (- this will ignore parentheses at the beginning of tokens but will find all others. Then copy the text back into a GoldVarb 2.0 file.

Also ensure that all lines in the token file begin with an open, curved parenthesis, "(", or a semi-colon, ';'.

Factor groups:

Ensure that there are no factor groups with less than two factors. If you have a factor group which has a single character and the fill character (set to "/" as a default), replace the fill character with another symbol before transferring the file to GOLDVARB 2001.

Spaces:

GOLDVARB 2001 reads multiple spaces as distinguishing coding strings from data. Therefore, there will be problems if more than one space appears anywhere in the file apart from between the coding string and the data or in lines beginning with a semi-colon.

There is no quick way of dealing with this. There are, however, two possible methods for dealing with the problem.

1. Copy your GoldVarb 2.0 file into a text-processing application and search for SPACESPACE. Change all occurrences of this which do not separate coding strings from data to SPACE, paying particular attention to those which appear at the end of lines.
2. Once you have opened your file in GOLDVARB 2001, and generated factor groups, go to Action|Check Tokens. The program will then run through the token file, and stop at problematic tokens. Check the log in the GOLDVARB 2001 window. It will indicate which token has caused the problem, and will include a few characters of text which appears directly underneath the offending spaces. Simply delete these spaces, and repeat the procedure until checking is complete.

Both of these methods are laborious, however the second will ensure that anything causing a problem with the token file will be found. This method will also find problems caused by coding strings being separated from data with tabs instead of spaces.

Special Characters:

Finally, and purely as a product of the different platform used by Macintosh computers, when you open your file in GOLDVARB 2001, you may find that any "special characters" (i.e. those characters which require a combination of Alt + another character to produce them) which you have used in your coding will have changed. For this reason, it is wise to use straightforward characters, or Shift + character combinations wherever possible and to spend some time after

transferring the file checking the GOLDVARB 2001 coding against your original coding key to ensure that you know which characters have been changed into which.

Summary of steps:

- Change () in text to [] or {}.
- Ensure lines begin with (or ;.
- Where / = "not-X", change / to Y.
- Change SPACESPACE+ to SPACE except where SPACESPACESPACE distinguishes coding string from data.
- Check special characters against your original coding key after conversion.

[Back to table of contents](#)

Error /warning messages

Within GOLDVARB 2001, there are a few error/warning messages which may appear as you use the program. While most of them are clear, there are a couple which may need some clarification.

Bad group number: Group index out of range 1:0

This may appear when you generate factor specifications for a token file. It simply means that something unexpected is in the token file - for example, it may be that a group has less than two factors. Simply click OK to proceed, then when the factors are generated and the Groups window appears, check that all is as it should be and edit as necessary.

Access violation:Ö

Messages beginning with this rather intimidating message mean that something has occurred which has, in effect, caused the program to stall. It will usually indicate a problem in the Condition file, so even if that is not the active window at the time, this is the best place to check for problems. For example, this message will appear in the Recode Definition window if you instruct the program to recode or exclude a factor before establishing which factor group is the application value.

Warning - bad cells

This will appear if you instruct the program to conduct a binomial analysis when there are knockouts in the Results file. Identify where the knockout is, then go back to the Conditions file and exclude the factor or recode it with other factors in the group.

[Back to table of contents](#)

References

Cedergren, H.J. & Sankoff, D. (1974) "Variable rules: performance as a statistical reflection of competence" *Language* 50:2. 333-55

Guy, G. (1988) "Advanced varbrul analysis" In Ferrara K, Brown, B, Walters, K, and Baugh, J (eds) *Linguistic change and contact* Dept of Linguistics, University of Texas in Austin. Austin, Texas. 124-136

Guy, G. (1993) "The quantitative analysis of linguistic variation." In Preston, D (ed) *American Dialect Research* Amsterdam and Philadelphia: John Benjamin. 223-249

Labov, W. (1969) "Contraction, deletion and inherent variability of the English copula" In *Language* 45:4. 715-762

Rand, D. & Sankoff D. (1990) *GoldVarb: a variable rule application for Macintosh*.

Sankoff, D. & Labov, W. (1979) On the uses of variable rules. *Language in Society*. 8 (2): 189-222

Sankoff, D. & Rousseau, P. (1979) "Categorical contexts and variable rules" In Jacobson, S. *Papers from the Scandinavian Symposium on Syntactic Variation, Stockholm, May 18-19, 1979*. Stockholm: Almqvist and Wiksell. 7-22

Sankoff, D. (1978) *Linguistic variation: Models and methods*. New York: Academic Press

Sankoff, D. (1988) "Variable Rules". In Ammon, U., Dittmar, N. & Mattheier, K.J. (eds) *Sociolinguistics: An International Handbook of the Science of Language and Society* Vol. 2. Berlin: Walter de Gruyter. 984-997

Wolfram (1991) The linguistic variable: fact and fantasy. In *American Speech* 66:1. 22-32