The distrat/askmap suite of programs for cause map analysis
A user's guide

Jeff Goldberg

February 4, 1996

Abstract
This report documents the use of the programs referred to by Markóczy and Goldberg (1995) for causal map elicitation and analysis. It only documents the software and not the method. Only the cause mapping analysis software is discussed and nothing is said about statistical analysis. *This is still incomplete.*

Biographical Note
Jeff Goldberg is the author of the software this report documents, and has contributed to the method that the software is designed to support. He has a bachelor's degree from the University of California, Santa Cruz in Linguistics, and was employed by the Hungarian Academy of Sciences Research Institute of Linguistics and the Theoretical Linguistics Program of the Eötvös University (ELTE), Budapest when most of this software was written. He is currently employed by the Computer Centre at Cranfield University. He can best be reached as J.Goldberg@cranfield.ac.uk. His PGP key finger print, for those who prefer secure Email, is 4E 33 96 71 1E 98 E7 93 20 E2 48 6D 52 A1 71 E0, and the key is available from keyservers and other places.

Acknowledgements
Your name here if you give me feedback. If you do, please specify that your comments are on version 0.74d.

Contents

Chapter 1

Introduction

This manual describes the distrat/askmap set of programs used to perform certain tasks useful for causal map manipulation and analysis. These programs perform many of the tasks described by Markóczy and Goldberg (1995)[1] (henceforth M&G95). This manual does not tell you everything you need to know in order to use the programs. In addition to this manual you will need to read M&G95 (at least pages 305–316). This manual is not a guide, description or tutorial on that method, but only describes the software. For reasons discussed in section 3.1, it can't even be a complete guide to using the software on your system, because I don't know the details of your system well enough to document how you should edit files, create directories, etc. Beyond that it is intended that this will be a complete and definitive guide to the software.

Throughout this guide, you will find that sometimes I use "I" and "me", while at other times I use "we" and "us". The plural refers to myself and Lívia Markóczy as authors of M&G95. The singular refers to myself, Jeffrey Goldberg, the author of the distrat/askmap software and this manual.

1.1 How to use this manual

Read everything in this chapter. Read only those sections of Chapter 2 that you need to in order to install the programs on your computer system. You must read or already know much of the material in Chapter 3 in order to understand the actual descriptions of the programs. If you make the mistake of trying to read the chapters on the particular programs before reading Chapter 3 you

[1] JAI Press has refused to grant the authors permission make a copy of that paper available electronically. Such an action is well within their right, but I do think that granting that permission would not cut into sales and would increase the number of citations of the article. Moreover, their refusal may well affact decisions of where we submit things in future.

will be left struggling. While not every sub-section of Chapter 3 is strictly necessary, you can only determine that by reading the first few paragraphs of each of those sections.

The chapters describing the individual programs then follow, with the large and more useful programs listed first. Do not attempt to skip to those chapters without going through Chapter 3 first. Chapter 4 describes the distrat program which is used to calculate the *distance ratio* between cause maps. Chapter 5 describes the askmap program which is used to assist the researcher in eliciting maps. Chapter 6 describes a program (drsort) that helps prepare the output data from distrat for input to statistical programs. Chapter 7 describes a program which calculates the *density* of cause maps. Chapter 8 describes iodeg which is used of calculated *in-degrees* and *out-degrees* of maps. Chapter 9 describes the program avmap which is used to calculate the *average* or *central* map of a set of maps. Chapter 10 describes a program ross used to get *Ross orderings*. These chapters are then followed by a number of appendices. The only one that I will mention here is Appendix A which you *must* read if you distribute this software or modified versions of this software. Even if you don't intend to distribute the software you might find the preamble of the license agreement of some interest.

1.2 Is this software for you?

The method described in M&G95 is, in my obviously biased opinion, the best method to date for doing solid work with cause maps. I hope that it will be built upon and will be usable. It is, however, a rather narrow method, applicable to a special class of problems as far as I can see. This software, other than its limited intrinsic interest, is not of much use to someone not using the method described. Thus, this software probably doesn't meet your needs. In order to save you some trouble, I list a set of guidelines which will help you determine how useful this software is likely to be. If you are someone who enjoys downloading, installing, and reading about new software, then you don't have to worry about whether it will be useful. The list is a rough set of guidelines which should be taken has indicators of the chance that you will be disappointed instead of as hard and fast rules.

- Have you still not read M&G95?

  The software and this documentation will little sense to you until you read that paper. I have heard rumors that the software and documentation makes more sense if you not only read the paper, but cite it as well.

- Do you expect the programs to tell you *what* the differences are between CMs instead of merely *how large* those differences are?

  If so, you will be disappointed and have probably not read M&G95.[2] Some of the simplier tools, such as iodeg (chapter 8) and density (chapter 7) may be of some minor, almost trivial, assistence in that, but those calculations can easily be done by hand, and play a small role in the eventual analysis.

- Have you collected data before you worked out how to analyize it?

  If so, unless you are extremely lucky and careful in your data collection, your data is probably nearly unusable. Although some more textually based data analysis tools may be of some help to you, distrat/askmap probably is not.

- Do you not really understand the method described in M&G95?

  It is not essential to understand the statistical analysis portion of that paper (if you don't wish to attempt that sort of statistical analysis). Also experimenting with the software can help you understand the method, but obviously you should never use a method you don't understand.

  The equation for the distance ratio there may look frightening, but contains nothing more that addition, multipication, division, and subtraction. Read through the text a few times slowly, maybe even with a couple very small maps as samples and doing the calculation by hand. If you have questions or comments about the method, please get in touch with either of the authors of M&G95. Remember, we want you to use the method, otherwise we wouldn't have gone to such effort describing it. A step by step example of the calculation appears in Markóczy (1995, Appendix D). It really isn't as hard as it may first appear and every effort has been made to make it less mysterious.

- Do you want to impress with numbers? Are you impressed by them?

  Explicit quantitative analyses are benificial when they are used to make things more clear. They tell your readers *exactly* what you did to get your results and lay all of the many assumptions that go into an analysis bare for all the world to see and evaluate. Methods should never be used to obscure or intimidate. Eden and Jones (1984) warned of the "dubious opaqueness" introduced by making analysis more technical than necessary, and it is a warning worth noting.

[2] Or M&G95 isn't written as well as it should be.

However, it is worth noting that the fact that a method does something with a computer can sometimes help "sell" a project to participants or funding bodies. Computer tools can make a method appear more modern or even more "scientific". But don't oversell things on that point; you only insult your participents, discredit yourself, and—which is far worse from my point of view—discredit the author of the software and authors of the method.

- Do you just want to see what the program looks like because it's there?

  Enjoy! But it doesn't look very impressive. The only mildly impressive bits are deep inside anyway, which show up only with the speed at which it does to calculation for large numbers of maps. Most of the programs are rather simple, and could be (and, I suppose, were) written by a casual programmer. The advantage of this programs is that they are here. They are most certainly *not* flashy.

- Do you want to use this software as an alternative to or replacement of other cause map software?

  There is almost no duplication between what these programs do and what other cause mapping software does. This is not due to some great originality on my part; instead it is a consequence of laziness:[3] If there already was software that did what I needed, why should I write something myself? Furthermore, now that these programs exist, there is little reason for anyone else to write new programs that do the same. Of course, someone could make improvements on these programs, and I hope that they will. For the most part, there is very little overlap in utility between these programs and other cause mapping software that I am aware of.

  Information about other cause map analysis software is listed in Appendix B. It may very well be that your disappointment in this package is matched by satisfaction with some of these others.

- Are you unwilling to read computer documentation?

  This is not plug and play take you by hand software. Not only do you need to learn what it does (by reading M&G95) but how to use it. Fortunately, you don't need to read the entire set of instructions. You should read the first bits (this chapter and parts of chapter 3) and then just the parts that you need.

[3] Laziness is one of the three great virtues of a computer programmer. The other two are impatience and hubris (Wall and Schwartz, 1991, p. 426). I hope that my programming reflects these.

> Still don't be put off by the length of this documention. Review section 1.1 to see how to make the most use (with the least time) of reading the manual.
>
> And if you get stuck don't be afraid to ask for help. Both authors of M&G95 want to see the method used and improved.

Now that I have frightened you off, let me try to coax you back. I obviously want to see this software used. Otherwise I would not have written instructions at all. (I didn't need the documentation myself, and I personally trained the person for whom I wrote the software.) So do take the existence of the documentation as a sign that I really do want to help you use it if it will be useful to you.

I have also helped people use it who (according to the list above) are not ideal candidates for my full enthusiastic support. I'm sure that I will continue to do so. Furthermore, it would be foolish of me to state conditions (other than what appears in the License Agreement) on the use or distribution of the software. You may have a creative use for that software which I could not anticipate. In fact, I hope that you do.

Finally, if it does turn out that the software really isn't for you for any of the reasons listed above or any of the countless alternatives, don't be afraid to let me know why. Methods only rarely compete with each other. For the most part they are designed with specific for specific projects and problems, and if one method is not appropriate, it is no shame on the authors of the method.

1.3 Legal matters

I must state that the software and accompanying documentation is distributed *as is* with *no warranty whatsoever*. See the Gnu Public License Agreement in appendix A. Because the software is distributed absolutely freely, I can afford neither to make any commitment to maintaining it, nor can I afford liability insurance.

The software—including the source code—is distributed freely and may be further distributed or including in other software which will be distributed according to the conditions of the GNU General Public License version 2 or later (included in the accompanying file `COPYING` and in appendix A). The preamble to the license agreement contains its rational. I want the software distributed without restriction. However, to make sure that all potential users and programers would have equally unrestricted access, I need to ensure that no one else places restriction on the software. Therefore, I must maintain my copyright in order to impose a restriction that no further restrictions be placed on the material.

Additionally, once a method, including that described by Markóczy and Goldberg (1995), is published it can be used freely by the research community, in which there are proper conventions for crediting those who have developed the method. It is professionally in the interest in the creator of a method to see it as widely used as possible, and if the method can be more easily employed with certain software, it is in the interest of the developers of the method to make that software as freely available to the academic community as the method itself.

I have been able to do that for the programs included. However, there are a number of statistical procedures described by M&G95 for which no accompanying software is provided. In some cases proprietary software was used "off the shelf", while in other cases I modified and extended code which is not in my power to distribute. I can bring my modifications under the terms of the GNU General Public License Agreement, but they are too well integrated in to code copyrighted by Numerical Recipes Software and can only be distributed to those who have a license from Numerical Recipes Software which automatically accompanies the purchase of the outstanding book *Numerical Recipes in C* (Press, Flannery, Teukolsky and Vetterling, 1988).

1.3.1 Trademarks

Throughout this document and others distributed with the programs I make use of various trademarks. In particular, I refer to "DOS", "MS-Windows", "Windows95", which are trademarks of Microsoft; "Unix", which is a trademark of Unix Systems Laboratories[4], but I use it to refer to Unix-like systems, such as linux, FreeBSD, Ultrix, OSF, etc (all of which are trademarks themselves) as well as Unix proper as licensed by USL; "Turbo C", which is a trademark of Borland Inc.; "TEX", which is a trademark of the American Mathematical Society; "LATEX", which, I believe, is a trademark of Addison-Wesley; "PostScript", which may be a trademark of Adobe Systems; "PCL", and "Lasarjet", which may be trademarks of Hewlett-Packard; "PKzip", which is also a registered trademark.

The above list is far exhaustive. Other trademarks may be used in this document; their use here and ommission from the above list should not be taken in any which diminishes their status as trademarks.

[4]USL with all assets (including trademarks) has recently been purchased by Novell and I can not longer keep track of who the trademark is registered to any more: AT&T, USL, Novell, X/Open...I have given up keeping track.

1.4 How to reach the author(s)

1.4.1 Reaching Jeff Goldberg

You can reach Jeff Goldberg by Email at *J.Goldberg@Cranfield.ac.uk*, or by post.

Jeffrey Goldberg
Cranfield University Computer Centre
Cranfield University
Wharley End MK43 0AL, UK

I actively encourge people to use secure email for even mundane transactions and for those who wish to use PGP[5] my public key (which is available via `http://www.cranfield.ac.uk/public/cc/cc047/`, or `finger:goldberg@csli.stanford.edu`, or `finger:goldberg@ny01.nytud.hu` or from any keyserver pgp-public-keys@keys.pgp.net) my key fingerprint is 4E 33 96 71 1E 98 E7 93 20 E2 48 6D 52 A1 71 E0.

1.4.2 Reaching Lívia Markóczy

Lívia Markóczy can be reached by email at L.Markoczy@Cranfield.ac.uk. Or

Dr. Lívia Markóczy
Cranfield School of Management
Cranfield University
Wharley End MK43 0AL, UK

FAX +44 (0)1234 751-???; Tel: +44 (0)1234 751122

Her PGP key finger print is F9 81 94 D7 AE 06 1D C7 5A F6 F1 E2 39 12 7D 8F

[5] [I will need to add one pointer to where to get information on PGP].

Chapter 2

Obtaining and installing distrat/askmap

2.1 Obtaining distrat/askmap

This reviews various ways of obtaining the software. There is some difficulty in working out the appropriate level of detail for this chapter. For those who are familiar with getting things over the network, I can just tell you the short answer which is

- The Management archive `gopher://ursus.jun.alaska.edu` in the *Management Related Software* directory. Or
- `ftp://ftp.cranfiled.ac.uk/public/cc/cc047/cause-mapping` which will always have the most up to date version. Or
- `http://www.cranfiled.ac.uk/public/cc/cc047/cause-mapping` which is just another way of getting to the same files.

In each case look for the file called `read.me` and read it for further instructions.

For those of you who find the above cryptic, the rest of this section should should help. But keep in mind that I most certainly can't tell you how to call up various internet tools at your site (ask for help locally) nor even tell you which tool is best or what the best location to go to is. That depends on what tools you have available (and what your preferences are) and where you are.

It is worth learning how to get things from the internet, even if the distrat/askmap package is of very little utility to you, since so many other more valuable things are available freely often under the terms of the GNU license agreement in Appendix A. You should ask your local computer support people for help, since they can show you how best to do this. Have this chapter ready and make an appointment with some computer support people (or, better yet, with a colleage who knows about these things) and do it.

2.1.1 How *not* to get distrat/askmap

Do *not* ask me to send you diskette. Do *not* send me a diskette.

With that said, let me list exceptions. I may be persuaded to arrange to have a disk sent to you if

- Your site doesn't have any internet access. You should complain *loadly* to the university administration if this is really the case.
- You have given this a try, but failed.
- You can persuade me that you have a project in mind which requires this software and is likely to lead to publication. That is, the software really "is for you" (see section 1.2).

2.1.2 How to get distrat/askmap using gopher

If it is easier for you to reach the UK than Alaska from your site, please look at section 2.1.4 (Getting the files in Europe) as well as this section.

If you are familiar with the Management Archive (`gopher://ursus.jun.alaska.edu`)

1. Get the files from the Management Archive. The programs are in a clearly labeled directory/menu in the "Management-Related Software" directory.
2. Look for a "Read Me" file which will tell you which files you need to get. Which files you need depends more on your what kind of archiving and uncompressing software you have, and whether you want the DOS executables. Look at the sections 2.2.1 and 2.2.2 for some information. Note that some gopher, ftp, or www programs which you might run may decompress some of the files for you automatically.

If you are not familiar with the Management Archive

1. Jim Goes maintains an electronic archive of management related material. It is freely available over the Internet. It includes many useful things including working papers. (You can also put your own working papers there.)

2. The archive is accessible via the gopher program (or any WWW browser). Your department's, school's or university's computer will have the program. You may need to ask your computer support staff about this. But it should be on some machine that has internet access.

   If you have World Wide Web (WWW) software (such as Mosaic, Lynx, or NetScape) you will not need a specific gopher program, since gopher is subsumed by WWW.

   If your cite really doesn't have gopher, ftp or a WWW browser at all, then your system administrator will have to install one. All it requires is that you have internet access. If you don't have internet access you will have to find somoene who does. You should cetainly make sure that your institute gets internet access.[1]

3. Do feel free to ask your computer support people for help with some steps in the processes. You may be able to figure things out after spending a lot of time that could be saved just by asking.

4. You will need to point gopher to the management archive which is at `ursus.jun.alaska.edu`.

   I cannot tell you how to do that, since gopher programs differ.

5. Once connected you should get and read the information available there by the menu. Note that menu items that end with the character / have submenu's. Also note that you are reading information off of a machine in Alaska. Depending on where you are you will require various amounts of patience.

6. The programs are in the "Management-Related Software Archive" section of the Management-Archive. You will see the menu item clearly.

7. Once you get into the menu with the distrat files you will need to save the files to your machine. On many gophers you just type 's' on a menu item and it will copy that file to your machine.

8. Once you have learned how to reach the Management Archive, you naturally may wish to do so again. You are encouraged to do so. In the main

[1] If your institution is based in a region where internet access is limited (usually due, not to poverty, but to the obstinence of national telecoms monopolies and governments which protect them) then I will send you a disk if you promise to make some statement in writing to the head of your instituton or to a relevant government ministy requesting that internet access be expanded.

directory of the Archive there are files that tell you more about it, and which tell you how to have your papers placed on the archive. It is too good a tool to pass up.

2.1.3 How to get distrat/askmap with FTP

You only need to read this section if there was no gopher or WWW to use, or if you prefer ftp in which case you probably don't need to read this section either.

First you will need a machine with Internet access and the ftp (File Transfer Protocol) program. Any machine that has an internet connection will have ftp. If you need help finding ftp, ask your computer support staff.

Connect with ftp to ursus.jun.alaska.edu with the command

```
ftp ursus.jun.alaska.edu
```

Of course the way that you call up ftp on your machine may differ.

You will get some welcoming message and will then be asked something like:

```
username (yourname):
```

You should type in `anonymous` Next you will be told "Anonymous login okay. Use complete E-mail address as password"

It will ask `Password` Do not type in your real password, do type in your E-mail address.

After this you will get some more welcoming messages. You may have to wait a few seconds (or not so few if you are connecting from a bad subnetwork) for responses from `ursus.jun.alaska.edu`.

Your prompt from ftp may look like

```
ftp>
```

But this depends on what system you are using. But to change directories with ftp on the Management Archive you use the command `cd` to ftp and the directory separator is `/` and not `\`.

You will need to change directory to `/pub/distrat`. You can do this in stages by

```
ftp> cd pub
```

and

```
ftp> cd distrat
```

At any point you can type either `dir` or `ls` to get a list of files and directories in your current directory. You can type `pwd` to to see what your current directory on the Archive is.

Once you are there you should get the files with the get command. All of the files, with the exception of `read.me` should be transferred in *binary* mode. Type the command `bin` to ftp to set up binary mode transfer. (Type `ascii` to set up ASCII mode transfer). So you might do something like the following (You only type what is after the `ftp>`. The rest is produced by ftp).

```
ftp> ascii
Type set to A
ftp> get read.me
local: read.me  remote: read.me
Opening connection for ...
... Transfer complete   (... Kbytes/Sec)
ftp> bin
Type set to I
ftp> get drdist.zip
...
```

You leave ftp with the either quit or bye. Of course you can use the opportunity of your connection to look around and see what other good things you may wish to get from the Management Archive.

2.1.4 Getting the files from Cranfield, UK

The files are also archived at Cranfield University in the UK. They can be retrieved either with ftp to ftp.cranfield.ac.uk in the directory `/public/cc/cc047/cause-mapping/` or by using a World Wide Web browser to `http://www.cranfield.ac.uk/public/cc/cc047/cause-mapping/`

2.1.5 Getting the files from others

Just because I (Jeff Goldberg) am too lazy to send you a diskette with the software (see section 2.1.1), doesn't mean that somebody else won't. If you for some reason can't get access to gopher or ftp or just don't want to, you can try to persuade a friend to do that for you and give you (or send you) as diskette.

There is nothing barring someone from going into the business of selling that kind of activity as a service as long as they give you all of the files that you have a right to as part of the system, and inform you of your rights under the terms of the General Public Software License Agreement in Appendix A.

2.1.6 Which files

There is a text file called "read.me". Get and read that for further instructions. It will more accurately reflect the situation on the site that you are getting the files from than this documentation can.

If you do not need DOS executables, then you will only need the source and documentation compressed (tar-ed and gzip-ed) archive called something like `drdist.tgz` for "Distance Ratio Distribution (tar gzip)"

If you need DOS executables you need the file `drdist.zip`.

The ASCII files archived in `drdist.zip` have DOS style newlines (0x0d0a) while the ASCII files archived in `drdist.tgz` have Unix style linefeeds (0x0a).

MSDOS users may also need the two programs `tar.exe` and `gzip.exe` or `unzip.exe` to uncompress and extract from the archives. These are either immediately present or the readme file will state where they are. The developers of those programs have placed them under the terms of the GNU software license agreement (Appendix A). The shareware program PKunzip can be used instead of unzip if, for some reason, you prefer paying for software instead of getting it for free.[2]

2.2 Unpacking the software

How the software is installed depends on the system that you wish to install it on. The programs are designed to work on a variety of systems, but require some very specific procedures for the very specific system you have. If you are not comfortable with the steps that follow, you should ask your computer support people for help. Show them the installation section of this document and ask for their assistance. You could even give them advance warning by sending them a copy of this chapter prior to meeting, so they know exactly what tools will be needed.

2.2.1 Unpacking the software from the .zip files

If you got this far then you have probably unpacked the things already, since this document, when it is completed, will be in the distribution. However, what you probably did to unpack the distribution is

```
c:
cd \
a:unzip a:drdist.zip
```

[2] PKunzip is an excellent program, but unzip is better, free, available with source, and works on many platforms.

which created a lot of files in the directory `C:\distrat`. Note that if you use the `.zip` file on Unix you should use the `-a` option to `unzip`.

2.2.2 Unpacking the software from the .tgz files

Note that different archives may use different names depending on what they can handle and when they think users can handle. A "gzipped tar file" really should have the extension `.tar.gz`, but some systems don't take that, so this is alternatively called `.tgz` or `.taz`. After un-gzipping, the file on all systems should be `.tar`.

If you have gotten far enough to read this, you have successfully unpacked the files. However, as a reminder of what you did, you copied the files `drdist.taz` (or `drdist.tar.gz`) to some place on your machine, and ran

```
gzip -d drdist.taz
```

or

```
gunzip drdist.taz
```

which uncompressed the file `drdist.taz` and replaced it with a larger file called `drdist.tar`. Your next step was to issue the command

```
tar -xvf drdist.tar
```

which extracted all the files from the archive, building several directories.

If you do this on DOS, you may have to give special options to the `tar` to make sure that ASCII files come out okay.

2.2.3 Directory structure after unpacking

After unziping (or untaring) you should have the directory structure listed below. Note that I use / instead of \ to list directory separators here.

`distrat`	The main directory, containing a readme, and some addministrative files.
`distrat/source`	The source files (C programs, makefiles, etc).
`distrat/bin`	DOS AT executables. For 286 and higher (they work fine on 386s, 486s, an Pentiums).
`distrat/386bin`	DOS 386 (or higher) executables. These can make use of all of your machine's free memory.

`drdoc.ps`	PostScript file (300dpi)
`drdoc.hp`	PCL (HPlaserJet) file
`drdoc.dvi`	If you know that this is, use it, otherwise, you can very safely delete it.
`drdoc.tex`	LaTeX source (not really usable since many specialized macros are not included)

Table 2.1: Forms of the documentation files in order of decreasing utility

2.3 The documentation

Save the documentation (the files in the `doc` directory). Even though you may not need certain programs at the moment, your needs may change. This documentation is you are now reading is intended to replace most of the separate files in the `doc` directory, but if they are still included in your distribution, then they are meant to be.

2.3.1 Printing the new documentation

The full documentation you are now reading can be printed out on either a PostScript printer or a PCL (HP Laserjet compatible) printer without requiring any text processing software at all. All you need to do is send either `drdoc.ps` or `drdoc.hp` directly to your printer, without loading and "printer drivers". The full documentation is more than pages long.[3]

Because I don't know the details of your system, it is difficult for me to specify exactly what steps you need to take to print. However, the documentation is in such a form that you need no special software (e.g., "Word", "WordPerfect", "TEX", etc), nor printer drivers to print out the documentation. You do need one of two very common types of laser printers. The documenation is distributed in four forms listed in order of decreasing utility in table 2.1

Since I have received few requests for the software from North America, these documents are produced for A4 paper. North Americans can still print them out on letter paper, but the margins will be funny. Also, I can easily produce letter sized versions if you would rather have that, but I don't want to clutter up the distribution files with many versions of the same file.

There are (a bit more than) three things you need to figure out at your site for printing:

[3] My apologies for being so long winded, but it is difficult to write a document that contains all the information that each person needs.

1. What Operating system are you using?

   MS-Windows and Windows95 are DOS as far as this documentation is concerned, but the commands listed are still why you type to a DOS prompt, which you can get at by opening a DOS window.

   (I will discuss only DOS and Unix below. If someone can provide information for Macs, I would be more than happy to include that here. Also, I will be assuming that Novell users will be able to more or less use the DOS instructions.

2. What kind of printer do you have access to?

   You will need either a printer which understands PostScript (A printer language) or which understands PCL (another printer language), which is the language used by HP LaserJets and compatibles. If you have a choice between the two, use the PostScript printer. Note that some printers accept both and automatically detect which kind of data it is getting.

   If you only have a postscript printer, then you can remove the .hp file, and if you only have access to a PCL printer you can remove the .ps file.

3. What is the name (port) to print on to get to that printer?

   For DOS users, this will be something like `lpt1:` or `lpt2:`. For Unix the names are far more arbitrary, like, `fred_ps`, or `Bldg63_P1`.

   For the rest of this section, I will use the following for my examples

`lpt1:`	DOS PCL printer
`lpt2:`	DOS PostScript printer
`escher_ps`	Unix PostScript printer
`durer_ps`	Unix PCL printer

   You will have to substitute these with the correct values (ports or names) for your printer and system.

4. DOS users may encounter other problems which I can only half anticipate, since there are too many ways things can be set up at your end.

2.3.1.1 *Printing `drdoc` on DOS*

First, let me remind you that you should *not* try to load either `drdoc.hp` or `drdoc.ps` into some sort of text processor or "printer driver". These files have already been prepared for common types of printers and need no further preparation. They are already in the form that the printer expects them in.

First go into the `doc` subdirectory of the `distrat` directory. If you did everything as above with unpacking that will be `C:\DISTRAT\DOC\`. The command `cd \distrat\doc` ought to get you there.

To print to a PCL printer on `lpt1:`

```
C:> copy/b drdoc.hp lpt1:
```

Note the '`/b`' option for the copy command. This is necessary for printing PCL files. Do *not* use the DOS `print` command; it will not work.

To print to a PostScript printer on `lpt2:`

```
C:> copy drdoc.ps lpt2:
```

Special Problems with DOS Depending on how your system is designed to talk to the printer, you may get an error message every few pages (depending on the size of the printer's memory). Keep hitting "retry" when prompted to. If this doesn't work, please ask someone local for help.

2.3.1.2 Printing `drdoc` on Unix

To print on a PostScript printer called `esher_ps` use

```
% lpr -Pescher_ps drdoc.ps
```

If you have a PCL printer called `durer_laser` use

```
% lpr -Pdurer_laser drdoc.hp
```

2.3.1.3 If you have problems

If you have problems printing this, take the files, and a printed of what you are now reading to someone who knows your system for more help. If that person needs to contact me with more questions they should feel free. I may be able to help you remotely, but too much of how to print is specific to how your system is set up. If you do wish to contact me for help, please be able to answer the three questions listed above about your printing system. There is no way that I can answer those questions remotely.

2.4 Installation

After you have unpacked things and taken a look at the documentation, you just need to put files in the correct directories. If you don't wish to (or need to) compiling things yourself, installation is easy.

2.4.1 Installing on DOS

If you use the DOS executables from the `drdist.zip` file then your installation is easy. Just place the executable files in some directory which is in your **PATH**.[4] If you plan to use the language features of the askmap program, be sure to place copy the `.aml` files from the `SOURCE` directory in a safe place. See the discussion in section 5.4 for more details.

If you do wish to compile the programs yourself see section 2.4.2.1 for instructions. If you don't know what this means then it very unlikely that you should recompile.

The distributed executables in `bin` are for 80286 or higher, running DOS 3 or higher. If you need executables for an XT (8088), please contact the author. If you are running on an 80386 or higher and you are finding yourself running up against memory limitations or need an extra bit of speed, you can use the executables in `386bin`. Note that to run those programs, the file `go32.exe` must be in your path. Also note that the `.aml` files are not duplicated in the `386bin` directory, so you will need to copy them from the `bin` directory or the `source` directory. Once you have decided which you want to use, you may wish to delete the other executables to save space.

2.4.2 Installing on Unix

Make sure that if you get the `.zip` file that you use the `-a` option to `unzip` when you unzip the archive, because files in that archive have DOS style newlines. Alternatively, use the `.tar.gz` which has Unix style newlines in the first place. For Unix, executables are not distributed, and you need to compile the files yourself as described in the next section.

2.4.2.1 Compiling the files

In the `source` directory you will find a number of files called `makefile.`*`xxx`*. The "*xxx*" indicates what system it was desigend for. You should identify which is the closest to your system. (`makefile.tcc` and `makefile.djp` are for DOS, so it is unlikely that you need those.)

If you have `gcc`, try `makefile.gcc` first:

```
cp makefile.gcc Makefile
make all
```

[4] The **PATH** environment variable specifies a list of directories in which programs are searched for when you type a command name. It is usually set in the `autoexec.bat` file with the command `path`.

2.4.2.2 Why aren't Unix executables distributed?

The form of a binary executable is a function of two things: The hardware of the machine it runs on, and operating system it runs under. Since Unix runs on so many different kinds of hardware, and for the same hardware there are often a variety of different Unices which have to be considered different operating systems as far as binaries are concerned, it would be difficult to produce binaries for so many different compinations of systems. Furthermore, you have better access to the hardware and operating system you will be using than I do; so, naturally, it is actually easier for you to compile then for me. If you encounter difficulties on you system, you should let me know.

2.4.2.3 Placing files

If the make went okay, you should have a number of binary executables. You can place them in a directory in your path (say `$HOME/bin`), run the shell command `rehash` if your shell needs that, and you are ready to go. You should also see the section 5.4 about the placement of the `.aml` files.

The binaries to install are askmap, avmap, density, distrat, drsort, iodeg, ross.

The program cmtran is utterly useless as a program, so should not be installed, but is provided as a template.

2.4.3 More on compiling the software

This section provides more details about compiling the software. Chances are you don't the material in this section, but it is provided in case you run into difficulties. If might be useful to someone who is helping you compile things.

If you are not installing executables already created for your system, either because the distribution does not have such executables, or because you would like your own specially configured version, or for safety reasons (A quick glance through the source files will convince you there are no viruses, etc), or any number of these reasons, you will need to compile the sources.

If you are familier with make and C then you should just need to look at the section 2.4.3.2, and in the event of errors section 2.4.4.1.

2.4.3.1 What you need

In addition to the files in source directory for this distribution you will need a

- C compiler that does ANSI for your type of machine. gcc is recommended.
- A linker/loader. (If you have a C compiler you will probably have a loader).
- The make utility. (Not actually necessary, but this documentation assumes that you have it.)

`Makefile.tcc`	TurboC on DOS. Use this as a model for other compilers on DOS.
`Makefile.dgp`	For DOS gcc (djgpp). This file has an extra layer for the coff2exe stage needed by djgpp.
`Makefile.gcc`	Standard Unix with gcc (Best and cheapest C compiler around)
`Makefile.unx`	Standard Unix, but using cc
`Makefile.dst`	Template `makefile`. Will *not work* on any system.

Table 2.2: Makefiles to choose from

Note you won't need these tools to run the executable programs, only to make them. So, if you don't have a C compilier you could compile the code on a machine that does (as long as it produces the right kind of executables for your machine).

2.4.3.2 Picking a Makefile

There are several makefiles provided, and with luck, one will work for your system without modification. The file `makefile.dst` is the general one that all of the others are modelled from. It will *not* work on any system. The make files provided are listed in table 2.2

What you should do, copy the file that seems the most reasonable to you to the file `Makefile`:

```
cp Makefile.xxx Makefile
```

or

```
COPY makefile.xxx makefile
```

And just try it the command `make all`

If all goes well, you should have the eight executable programs: askmap, avmap, cmtran, density, distrat, drsort, iodeg, ross. On DOS these will have the extension `.exe` on their names.

You can copy these to where every you want them, or you can use `make install`

But, before using `make install`, you should look at the `Makefile` and see that `DESTDIR` is set up the way you want it. (and that the directory exists on the system). So you might have to edit the `makefile` for `make install`. So either use your "copy" command to copy the files by hand, or read the next section to learn more about make.

2.4.3.3 Can't find `unistd.h`?

If, when trying to compile on Unix, you get an error about not being able to find a file called `unistd.h`, then you will need to modify the makefile that you are using. Look for `HAVE_UNISTD_H` in the file, and put a `#` character at the begining of the line. Also, let me know, so that I can list in this documentation what systems appear not to have `unistd.h`. More information about how make works, and what a `makefile` really is is listed in section 2.4.4.

2.4.3.4 Lib warnings from tcc

The first time you compile using tcc, you will probably get warning messages from tlib about modules not being in the library. Don't worry about this at first. Those warnings are just tlib's way of saying that the modules are being added to the library for the first time.

2.4.3.5 Other problems compiling

The number and range of difficulties that you could encounter when compiling makes it difficult for me to list out everything. But in addition to those listed above, you may need to refer to section 2.4.4.1 for other half anticipated problems. If my programs are written correctly, then the only thing that needs to be modified are the `makefiles` or things about your system.[5] If you do have problems compiling, please let me know. In many cases, someone at your site is more likely to be able to help then I will, but you can it would still be useful for me to know (especially if there is something in the source files that need to be changed).

2.4.4 About make

If you are unfamiliar with make, it is worth understanding basically what make does. If you do know about make, then just skip this section. If you are just installing binaries or the quick installation instructions worked for you, then you certainly don't need to read this little introduction to make, and you probably have a better use of your time. This section is to give you some conceptual background if you actually do need to modify the makefiles.

Make reads a special file (called a `Makefile`) which lists what files depend on what, and how to make the former. For example, suppose that you worked in a text editor that created files in a particular form (lets say .XXX), but for other purposes you sometimes needed the file in RTF form (rich text format). Suppose that your editor didn't have a way of saving in RTF, but that you had a program, called xxx2rtf which did the translation. If this case, the .rtf file (the target) depends on the .xxx file, and the makefile would have a rule like

[5] If my programs are not written correctly (beyond the "limits" and "bugs" listed in most chapters), then if I could tell you how to fix it, I already would have fixed it.

```
foo.rtf:        foo.xxx
                xxx2rtf foo
```

If this were the case, then when you typed the command, `make foo.rtf`, then make would check to see if the file `foo.rtf` already exists, if it does it would check to see whether the file `foo.rtf` as been modified more recently then the file `foo.xxx`. If so, make would do nothing because foo.rtf is up to date.

If, on the otherhand, `foo.xxx` has been modified more recently (or is not up to date), then make would run the command `xxx2rtf foo`. If `foo.xxx` depends on other things (as possibly stated elsewhere in the makefile), then make would first try to "make foo.xxx" and only after `foo.xxx` is up to date would it run the command to make `foo.rft` from `foo.xxx`.

An almost real life example: I use the typesetting system TEX. The program TEX reads in a file which is usually created by a person and produces something called a `dvi` (DeVice Independent) file. To print the `.dvi` file, one needs to translate from that to the kind of file that the printer wants. Let's say an `.hp` file for printers that use PCL (Hewlett-Packard compatible Printer Control Language). In this situation one could have a makefile with the following information:

```
foo.dvi:        foo.tex
                tex foo.tex

foo.hp:         foo.dvi
                dvi2hp foo
```

So if I edit `foo.tex` and want to produce the `.hp` file, I would just type `make foo.hp`.

Things can get more complicated. Sometimes I don't what to type in the `tex` file directly, but have it produced by a program that reads something that I do type in and translates that to TEX. There is a program that will automatically put in the TEX commands for Hungarian hypenation so the user doesn't have to use a special version of TEX for Hungarian. This program, **hion**, takes files `.hun` and makes `.tex` files. In this case it is the `.hun` file that the human edits. So the makefile would look like:

```
foo.dvi:        foo.tex
                tex foo.tex
```

```
foo.hp:         foo.dvi
                dvi2hp foo

foo.tex:        foo.hun
                hion -TEX foo.hun
```

Of course A TEX file can input other files, so sometimes the `.dvi` file depends on more than one file.

```
foo.dvi:        foo.tex table1.tex figure1.ps appendix1.tex
                tex foo.tex
```

And there could be all rules about making the things that `foo.dvi` depends on.

Two more things about `make`. First it is possible to teach make how to make things based on the file extensions instead of the full file names. So for example, it would be possible to tell make that any file *name*`.dvi` can be made from any file *name*`.tex` by running `tex` *name*`.tex`. I won't explain how. Some of these sorts of rules are "builtin" to make.

The second other thing about `make`, is that you can define variables

```
RENAME=ren
...

             $(RENAME) foomain.dvi foo.dvi
```

Lines in a make file that begin with a `#` are not processed by make; So a makefile could have

```
#### use the ren for DOS
RENAME=ren
#### or mv for unix
#RENAME=mv
...
        $(RENAME) foomain.dvi foo.dvi
```

If none of the Makefile templates that are provided with this distribution work exactly for your system, then you will have to change some of these sorts of things for your system, eg, from the above to

```
#### use the ren for DOS
#RENAME=ren
#### or mv for unix
RENAME=mv
```

See the difference? (the second uses `mv`, while the former will use `ren`).

2.4.4.1 If there were problems with make

What you need to do depends on the program, so you will need to read the output of make carefully. Note that make also displays the commands that it calls. This way you can see how far make got.

- Error type: Out of memory

  You are obviously using DOS. See the section on special instructions for DOS (section 2.4.4.3).

- Error type: blah blah Command not found

  If this is something like gcc command not found, then it is likely that the named C compiler from the makefile does not exist on your system. The makefile has a section for selecting the compiler. You will need to make changes in that section.

  If you are on Unix you can safely use `CC=cc`. On DOS you will need to know what the name of your compilier is. You may need to change many things in the DOS version if you are not using tcc.

- Error type: file `limits.h` not found

  This error might happen during a compile (most likely when compiling `rossify.c`). It means that you don't have a properly configured C compiling environment. You can get around this particular problem by setting `-DNOLIMITS` in the "Predefineds" section of the Makefile. But chances are, if this has gone wrong other things will go wrong as well.

  You should safely be able to restart the make, without having to remake the other things already made.

- Error type: parse error, or syntax error, or bad cast

  If you follow the output of make carefully you should be able you should be able to see what file the compiler was working on. The error message from the compiler will say what line number the error occurred on (or near). It may also give a number for the error, so be sure to guess that you are looking at the line number and not the error number.

Look at the line in the appropriate file. Don't worry, you won't have to change anything in the source code. This is just to see what kind of error it is.

If on or near that line there is `ANYPTR`, then you will should edit the predefines in the Makefile to use `-DANYCHARPTR`. Chances are that if you have encountered this error you will also encounter the more serious errors described next. Anyway, it won't hurt to make the change to the makefile and try again.

If on or near that line was something like

```
blah blah
function(blah blah, blah blah)
{
```

It doesn't matter what the 'function' or 'blah' is, what matters is that in the '`(`' and '`)`' there are (at least) two 'blah's between any commas or the the parenthesis stuff looks like `(void)`. Also this will always be at the begining of a line.

If that is what you found, it means that your C compiler doesn't understand prototypes in function definitions. If this is what happened then, at the moment there is nothing to do, but try a more modern C compiler.

In particular if you are compiling on a Sun workstation. Do not use `cc`, but use `gcc` or `acc`. There is a tool called `unproto` which will fix modern programs to work with obsolete C compilers.

If the syntax error was in the file `distrat.h` and near a line that looks like

```
extern blah function(blah blah)
```

then your compiler doesn't even understand prototypes in function declarations. Again, if you have this problem then you probably have the above problems as well. Try another C compiler.

2.4.4.2 Reporting problems

Naturally, I would be interested in knowing about problems you encounter, but my curiosity does not entail a commitment to fix anything. But it is certainly more likely that I will be able to fix things if you give me as complete information as you can. Tell me about your system. (on Unix the `uname -a` command will tell you. About your compiler (for gcc the command `gcc -v` will tell you). Most importantly, I need the error messages. Please run make putting all make out put into a file, say `make.out`. On DOS do this with

```
make all > make.out
```

On Unix, if you are using csh or some derivative of it like tcsh use

```
make all >& make.out
```

Use the & to get the error messages there as well. If you are using the Bourne (standard) shell (sh) or the things like it (bash, Bourne Again SHell) or the Korn Shell (ksh)

```
make all >make.out  2>&1
```

If you don't know which type of shell you are using, type

```
csh
make all >& make.out
exit
```

This way, you will use the csh.

2.4.4.3 Special DOS Compiling difficulties
If you have never used make or a compiler on your machine, it is really best for you to use the assistance of someone who is familiar with these tools and DOS, since you often need to mess around with memory managers to get make and the C compiler to run. (The distrat/askmap programs do not need any special memory manager to run, it is make, the linker and the compiler that need extended or expanded memory or use special memory managers.)

As an warning, I once attempted making this with exactly the same compiler and make on Livia's machine which has sufficient RAM, but it turns out that the DOS utility Smartdrv was installed in such a way it took too much memory, so make wasn't getting enough to be able to it on the the linker. It took me several hours one day, a night's sleep, and an hour the next to identify the problem. Once identified all worked well. (A DOS expert would probably figure it out much faster).

The novice should feel free to try to compile under DOS, but if you get an out-of-memory error during compilation, consult a local DOS programer. And when you start cursing, that cursing should be directed not at me, but toward Seattle. The problems aren't my fault nor the fault of the developers of the C compiler (unless, of course you are using mcc, but that's just guilt by association.)

Chapter 3

Using the programs: General issues

Before providing documentation for each of the programs in the package, there are some general issues which are common to all of the programs.

Although the command-line syntax of each command is different, there is a commonality of style and conversions, which, once explained in one place only, will make the description of individual programs easier. So general properties of command line conventions are explained in section 3.2. If you are very comfortable with Unix style commands and command lines, you can probably skip that section.

Many of the programs read or write files which contain causal maps in a particular format. The format of the causal map files is then also described separately from the specific documentation of each program in section 3.3.

Several programs take lists of map file names. The format of these files is discussed in section 3.4.

Many of the programs exit with a particular exit states on different sorts of error conditions. Many of these are common across these programs. These are listed in section 3.5. If you don't know what an exit status is, then you don't need the information in that section.

3.1 Modularity and design philosophy

Over the past ten years it has become increasingly popular to design software packages which are large integrated packages that perform all the required computer tasks for some project. In such a system—and distrat/askmap is not such a system—the user calls up an "environment" on the computer and does everything within that environment. distrat/askmap, on the otherhand, instead of being a closed integrated system, is an open modular system. Modularity is well worth the price of making the documentation more difficult to write. With an integrated system you need to know nothing about your particular computer or how the program interacts with your computer (ie, you need know nothing about files and directories if that is what your computer uses), but you need to learn the system of the integrated package fully.

With a modular set of programs, you need to learn less about things specific to the programs, but you need to know a little more about the your computer system (ie, how to create and view files, etc). Since every computer system has at least one—and maybe up to a dozen—ways of viewing simple character files there is no need for me to write a file viewer into distrat/askmap, nor is there a need for you to learn a new one. Modularity means that the programer does less work and the user has less to learn about particular programs. There are other advantages of modularity, which should become clear below; but the chief drawback is that since you have the freedom to use, for example, whatever file viewer you wish to use, I cannot provide full documentation. This is because while *you* probably know how to view simple files on *your* computer, I don't know how to do so on *your* machine.

Modularity also allows the system to grow easily without requiring central control: Anyone can add a module. If some user needs some other form of analysis for maps of the same type as are used here, it would just be a matter of writing a module that does that and only that. Ideally, the new module would be constructed so that it fit in with the existing modules, but that is up to the creator of the any new module.

3.2 General command line syntax

You can skip the rest of this section if you understand fully this statement:

> The programs use traditional Unix style options and arguments. The non-interactive programs write to standard output. The program drsort is the only program that reads from standard input.

If you didn't understand that, then read this section and you will by the end.

3.2.1 Command line arguments

Many of the commands take command line arguments. For these programs they are often names of files which contain causal maps. For example, the program density which can be used to calculate the ratio of arcs in a map to total places there could be arcs.

```
density filename
```

where 'filename' is the name of a file which contains a map. In this example 'filename' is the command line argument. Naturally it should be the name of an actual file on your system. Some commands take no command line arguments, some take one, some take more, and some take a variable number of arguments.

3.2.1.1 Command line options

In addition to command line arguments provide on the command line, most of the programs take a number of options (which are optional as the name suggests). These change the behavior of the programs in certain ways. Options, sometimes called 'flags', are single characters and are immediately preceeded by a hypen, '`-`', character.

For example, with this program (and almost all of the others as well) there is an option `h` which displays a very terse summary of the command line syntax for the particular command. So typing the command

```
density -h
```

will display the command line syntax information instead of trying to calculate the density of anything. Many of the commands also have a `v` option which stands for 'verbose'. When this option is used to program runs as normal, but may write out additional information about what it is doing to your computer screen while it is doing it.

```
density -v filename
```

will calculate the density of the map in the file called 'filename', but will write out additional information as well. Note that the option precedes the command line arguments.

Some options take arguments themselves. In such cases the arguments to the options are obligatory and must follow the option (but there may be whitespace between the option and its argument). The option `A` for the density can be given a numerical argument (actually only 0 or 1) which specifies whether arcs can reasonably occur from a node to itself.[1] If the `A` option is given the argument '0' then such reflexive arcs (or the lack of them) will be calculated as part of the density, while if it is set to '1' then reflexive arcs (or places where they could be) will be ignored. (See chapter 7 for more explanation.) So,

```
density -A 0 filename
```

[1] One would think that an option that only has two possible values would be a simple option instead of an argument taking one. Normally that would be the case, but the reason why this option is they way that it is will become clear from the discussion of the `distrat` program in chapter 4.

will calculate the density of the map in 'filename' assuming that arcs can occur between nodes and themselves. The command could have also been typed as

```
density -A0 filename
```

If we wanted to make the same calculation verbosely, this could be done with

```
density -v -A0 filenane
```

or

```
density -A0 -v filename
```

Options which don't take arguments can be grouped together (with no space) behind a single hyphen:

```
density -hv filename
```

although this particular example is a bit silly, since the `h` option means that nothing really gets calculated.

Here are some general points to note about options:

- options are always single characters
- options, if used, must appear on the command line before any command line arguments.
- option names are case senstive: '`-A`' is not '`-a`'.
- if an option takes an argument it, the argument is obligatory and the options arguments may or may not be separated from the option by space.
- only options which do not take arguments may be clustered together behind a single hyphen.
- There is no guarantee that an option with a particular name, say 'x', will have the same meaning for every program. However, when a group of programs are written a part of a package, it is usually the case that the program author will keep option names as consistant as possible across programs.

Many of the programs which read in maps have a special option, `-f`, which takes as its argument the name of a file which contains a list of filenames, which in turn contain maps. Typically when this option is used the normal command line arguments are no longer needed. So, the command

```
density -f listfile
```

will read in a list of file names from a file named 'listfile'. Note that the file name 'listfile' is the argument to the 'f' option, and is not a command line argument. It would also be ok to say

```
density -flistfile -A 0 -v
```

If you were to actually run

```
density -h
```

the result would be something like

```
CM Density. Copyright 1993, 1994 by Jeff Goldberg
Usage: density [options] file
   or  density [options] -f listfile
        files contains a map
        listfile contains a list of files
        Options:
                -h      display this message
                -v      set verbose flag [off]
                -d num  set debug flag to num [0]
                -n num  set max. distinct nodes [300]
                -m num  set max. maps in list [200]
                -t num  set type of map file [1]
                -A num  set alpha to num [1]
```

Most of these options haven't been explaned yet, but by looking at the option that you already know can learn how to read that rather terse usage message. The information in the square brackets, displays the value value which the program is executed with if the option is not stated. It should also be noted that not even I remember which options are available with which programs and what they are called. I make heavy use of the 'h' option.

3.2.2 Saving program output

With the exception of askmap, none of the programs write their output to a file. Furthermore there are no options you can specify which will tell them to do so. Yet, if there were not some easy way to save the output of these programs, they would be useless. Your command interpreter provides the mechanism that is needed.

With all Unix shells and DOS command interpreters that I know about one can tell the operating system to *redirect* to screen output of a program to a file by using on the command line the '>' symbol followed by the name of the file you want to output to go to.[2]

Suppose that you wanted to save the output of the distrat program to a file called `dr.out`. Suppose that you had a list of map files in a file called `maplist`, then the command

```
distrat -f maplist
```

would display a great deal of information to the screen. Programs are said to write to the *standard output* when they write information to the screen in such a way that the output can be easily redirected to a file. The information produced by the distrat program for example, is written to standard output (normally the screen). If you use the command line

```
distrat -f maplist > dr.out
```

the program still writes to standard output, but your computer system will treat standard output as the file `dr.out`. If the file `dr.out` existed before this command, it will be overwritten.

Not everything that a program writes to the screen is written to standard output. For example, the copyright notice and various error messages are written to the screen, but are not written to standard output. This means that when you redirect standard output from the screen to a file using > this other information (error messages, copyright notice) and so on, are still displayed on the screen and not sent to the redirected standard output file with the other data.

Again, let me remind you that this redirection tool, '>', is not something written into the program. Both when you use it and when you don't the program simply writes data to its standard output. However, when you don't redirect standard output your computer's operating system will treat the screen as standard output; and when you do use output redirection, your computer treats the named file as standard output. distrat won't even know if it is writting to the screen or to a file; all it knows is that it is writting to standard output.

3.2.3 Redirecting standard input

There is a similar notion, *standard input*, which only matters in the case of one of the programs provided: drsort. drsort expects data from its standard input, which, unless the operating system is told otherwise, will be the computer

[2] I know of one shell, jsh, which uses Japanese word order and treats the redirect symbol '>' as a *post*position instead of a *preposition* so it actually must appear after the file name and not before.

keyboard. It is highly unlikely that anyone will want to give that program keyboard input; so, in order to tell the operating system that the standard input of drsort should come from a file, say dr.out, the command would be given as

```
drsort < dr.out
```

Since drsort not only reads from standard input, but also writes to standard ouput, the output of drsort could be redirected to a file with something like

```
drsort < dr.out > outfile
```

There are other things that Unix users can do with standard input and standard output that introductory material to Unix covers better than I can here. In most practical terms, DOS users are limitted to the sorts of examples shown above.[3]

3.3 Map file format

The files (both the files containing maps and lists) can be created and edited by any ASCII editor such as Emacs, vi, Elvis, dosedit, the Norton editor, or any other of scores of such editors. The files can also be edited using word-processors such as Word Perfect making sure that they are saved in ASCII or Text form. In what follows 'line feed', 'new-line', and 'return' refers to what is sometimes referred by such word processors as 'hard returns' or 'hard line-feeds'. The files can also be created by the askmap program which is the subject of chapter 5.

The files containing maps must be in a specific form. In many respects the form is very strict, while in other respects it is very flexible. The best way to understand what the file can be is to look at the samples (`tun01` `tun02` `tun03`)

In some of the examples below, I use the symbol '␣' to indicate a single space. The file *must* begin with the string

```
Map␣id:
```

The capitalization there matters as does the space between the 'p' and the 'i'. The colon ':' is also required. This must be followed by a space and the an identifier. For example:

```
Map␣id:␣Joan-Smith-1
```

[3] DOS command interpreters do provide a pipe simulation, but since the operating system itself does not actually handle pipes, the simulation is hardly worth the risk of the mess left behind a broken pipe on DOS.

Here the identifier is 'Joan-Smith-1'. The identifier must contain no spaces or tabs. The identifier must be less than 32 characters long.

The next line of the map file must contain the number of nodes in the particular map. So the first few lines of a file may look like:

```
Map␣id:␣Joan-Smith-1
12
```

The number of nodes in a map must be at least two, otherwise some of the calculations based on the maps may be undefined. The programs are currently written so as to not accept maps that have fewer than two nodes. See section 9.2 for a discussion of how such maps might arise.

After the first two lines the rules about newlines and spaces become much more flexible. Anytime after the second line where a space, newline, or tab character can be, you can use any number (greater than zero) of spaces, newlines, tab characters in any combination.

If the number of nodes, as in the example, is 12, then the next 12 numbers must be the numbers of the nodes in the particular map.

```
Map␣id:␣Joan-Smith-1
12
35␣22␣31␣45␣␣2␣29␣17␣19␣63␣53␣54␣␣9
```

The node numbers can appear in any order, but the order determines the layout of the association matrix. That is, the first node number specified will determine what node the first row and column of the associate matrix will correspond to. So for the example we are working through, the number in row 1 column 4 of the association matrix indicated the influence of node 35 on node 45. A number in row 6 column 2 would indicate the influence of node 29 on node 22.

Following this, the array of influences appears:

```
Map id: Joan-Smith-1
12
35 22 31 45  2 29 17 19 63 53 54  9
 0  3  3  1  0  3  3  2  3  3 -1  0
 3  0  3  2  0  2  3  2  0  2  2 -2
 3  2  0  2  0  3  3  2  2  3 -3  2
 2  2  2  0 -2  2  2  0  2  0  0 -1
 2  0  0  0  0  2  0  0  2  0  1  1
 3  2  2  0  2  0  3  3  2  3  2 -1
```

```
 3  3  2  0  0  2  0  2  2  0  3  0
 2  3  0  0  0  2  3  0  0  3  0  1
 0  0  2  0  0  2  2  0  0  3  1  1
 3  2  3  0  0  3  0  0  3  0 -1  0
 1 -2  1  0  2  1  1 -3  1 -2  0 -1
 2  3 -1  0  0  2  0  2  2  0  3  0
```

Again note that anywhere after the line indicating the construct numbers, all tabs, spaces, and newlines are equivalent. The same map could have been written as

```
Map id: Joan-Smith-1
12
35 22 31 45 2 29   17 19   63 53 54 9

0 3   3 1 0 3 3  2 3 3 -1 0 3
          0 3 2 0 2 3 2 0 2 2 -2 3 2 0 2 0 3 3 2 2
3 -3 2 2 2 2 0 -2 2

 2 0 2 0 0 -1 2 0 0 0 0 2 0 0 2 0 1 1 3 2 2 0 2 0
3 3 2 3 2 -1 3 3 2 0 0 2 0 2 2 0 3 0 2 3 0 0 0 2
 3 0 0 3 0 1 0 0 2 0 0 2 2 0 0 3 1 1 3 2 3 0 0 3 0 0 3 0 -1
0 1 -2 1 0 2 1 1 -3 1 -2 0 -1 2 3 -1 0 0 2 0 2 2 0 3 0
```

although using such a form is not recommended for obvious reasons. But what users should note here is that the program may not detect errors in your input while it is reading the file. This lack of syntax checking was a deliberate design decision, to make the reading of the file quicker. The other way to make a reading quicker would have been to impose an extremely rigid file form.

Finally, the program allows any material to appear after the map. So maps could look like:

```
Map id: Joan-Smith-1
12
35 22 31 45  2 29 17 19 63 53 54  9
 0  3  3  1  0  3  3  2  3  3 -1  0
 3  0  3  2  0  2  3  2  0  2  2 -2
 3  2  0  2  0  3  3  2  2  3 -3  2
 2  2  2  0 -2  2  2  0  2  0  0 -1
 2  0  0  0  0  2  0  0  2  0  1  1
 3  2  2  0  2  0  3  3  2  3  2 -1
```

```
 3  3  2  0  0  2  0  2  2  0  3  0
 2  3  0  0  0  2  3  0  0  3  0  1
 0  0  2  0  0  2  2  0  0  3  1  1
 3  2  3  0  0  3  0  0  3  0 -1  0
 1 -2  1  0  2  1  1 -3  1 -2  0 -1
 2  3 -1  0  0  2  0  2  2  0  3  0
Nationality: Latvian
Age: 34
Job-Title: Marketing Directory
Node-35: Market share
Node-22: blah blah blah
```

The distrat/askmap set of programs will not make use of that information, but other programs might. The distrat/askmap programs will not read beyond the last element of the association matrix.[4] Note that if something is missing from the association matrix (a line, or even a single element) the program will try to read past the end of the array. Its behavior then is very hard to predict, but it will most certainly lead to either erroneous results, or, if you are lucky, an error message.

One more note about the files: The influences can be floating point numbers:

```
Map id: test-2
4
   3      5      8     12
   0  -1.54      2   2.40
2.00   0.00   -3.2      2
   1  -2.25      0    2.3
   0   -1.1     -2      0
```

This will allow files produced by the avmap program to be usable by other programs in the package.

Currently, the programs assume that all input maps have at least one node. No calculation are performed for empty maps. However, the output of avmap may be an empty map.

3.4 The form of the list files

The list file (the file used with the `-f` option) is simply the name of each map file. In the list file, each map file should be on a complete line by itself and

[4]The program askmap under the rare circumstance that it reads a map file does expect a particular bit of extra information.

there should be no blank line, nor extraneous spaces in the file.

A list file might look like:

```
hun01.map
hun02.map
hun03.map
```

3.5 Exit states

Many of the programs exit with various exist states depending on error conditions. If you don't know what a exit status is or what to do with one, then you most certainly don't need to read this section. Here they are:

Status	**Meaning**
0	Normal exit
1	Usage Error
2	File opening error
3	Impossible error (inform author if you get this)
4	Node out of range
5	Memory error (could be out of memory)
6	Error with input data

Chapter 4

The **distrat** program

The distrat program calculates the 'distance ratio' between causal maps. For definitions of the distance ratio (and of causal maps) see Markóczy and Goldberg (1995). The program is well suited for the cross comparison a large number of maps and should run reasonably well on even the machines slowest machines.

4.1 Calling distrat

As with many of the other programs that read map files, there are two forms for calling distrat. One is where two file name arguments are provided, and each file contains a causal map, eg,

```
distrat mapfile othermap
```

In this case distrat will calculate the distance ratios between the maps stored in files *mapfile* and *othermap*. These files are in the form described in section 3.3. Of course, as described in section 3.2.1.1, the various options to this command can appear before the command line arguments.

In the other usage, the '`-f`' option is provided and its argument is the file name of a file containing a list of other file names (see section 3.4). Those other files contain the maps to be compared. For example if you wish to calculate the distance ratio between all of the files `map1`, `map2`, `map3`, `map4`, `map5`, `map6`, `map7`, and `map8` you can create a file that just contains the names of the map files, one file name per line. Suppose the contents of the file `maplist` is

```
map1
map2
map3
map4
```

```
map5
map6
map7
map8
```

then you can calculate the distances between the maps in all of the map files with a command like:

```
distrat -f maplist
```

Of course all of the other options can also be specified, e.g.:

```
distrat -G2 -d 3 -f maplist
```

4.2 Options available with distrat

In addition to the `-f` option described above, there are a number of other options. The options `-A` through `-E` can be used to set the values for the parameters α through ϵ. The 'factory default values' uses settings $\alpha = 1$, $\beta = 3$, $\gamma = 1$, $\delta = 0$, and $\epsilon = 2$, which have the formula reduce to Langfield-Smith and Wirth's (1992) formula 12. The settings used by Markóczy and Goldberg (1995) and Markóczy (1995) are the same except that $\gamma = 2$.

The options `-d` and `v` can be used to have more information displayed to the screen while the program is running about, for example, intermediate calculations.

The options `-m` and `-n` can be used to reset some program limits on number and size of maps.

The option `-c` can be used to set the cache size, and is discussed in much more detail below.

Here is an almost complete list of options.

`-h` Lists all command line options

`-d` *N* This sets the debug level to N, the higher the value of N, the more internal information the program will provide about what it is doing. The default value is zero. (This option may not be supported in future versions.)

`-v` Tells the program to run "verbosely" (same as `-d3`) (This option may not be supported in future versions.)

`-f` *fname*
: The file *fname* contains a list of file names all to be compared with each other. If this option is given then no files name arguments may be given on the command line.

`-A` N Set α to N. Default 1

`-B` N Set β to N. Default 3

`-G` N Set γ to N. Default 1

`-D` N Set δ to N. Default 0

`-E` N Set ϵ to N. Default 2

`-n` N Sets the maximum node number to N. All node numbers in maps must be between 1 and N inclusively. By default N is 300. See the discussion on optimizing performance in section 4.2.5 for more about this option.

`-m` N Sets the maximum number of maps to N. This is number maximum number of names in the listfile. Default N is 200. This option is only relevant in conjunction with `-f`. See the discussion on optimizing performance in section 4.2.5 for more about this option.

`-c` N Sets the maximum number of maps to be remembered when comparing maps from a list. If there are more than N maps in the list, maps will sometimes have to be "forgotten" and then re-read from the file. The default size of the map cache (N) is 40. This option is only relevant in conjunction with `-f`. See the discussion on optimizing performance in section 4.2.5 for more about this option.

`-t` N Sets the format type of the map files to be read in. N must be a number in the range 1–9. Currently only type 1 is available, and is discussed in section 3.3. The program has been written so that it is easy for other programers to add new file formats. The notes in the source file `readmap.c` should be consulted for further details. Generally, numbers 7, 8, 9 will be used for file formats specific to your site. Lower numbers, such as 3, 4, 5 and 6 should be used for formats in more general use.

Default is 1.

4.2.1 The basic options

The use of `-f` has already been described. This option is probably the most common one. Also, option `-h` has been described in section 3.2.1.1. The options `-v` and `-d` can be used by the user to get some information about the internal state of the program during its operation. However, exactly what information is provide for different settings of `-d` or for using `-v` is subject to a tremendous amount of variation from one version of the program to the other.

4.2.2 Parameters

The meanings of the options for setting α through ϵ are explained in the paper by Markóczy and Goldberg (1995) and not in the documentation for the software. The same is true ofthe actual meaning of the distance ratio calculation.

4.2.3 Map limit options

The options `-n` and `-m` only need to be reset from the default if you are reading more maps than the default value for `-n`, or your highest node number for a map is greater than the default value for `-m`.

4.2.4 Map cache

The option `-c` can be adjusted when you are reading a large number of maps (or very large maps). The value of the argument of this option is the maximum number of maps which remain in the program's memory (the map cache) when the program is running. It is only relevant when the `-f` option is used. If the value of `-c` is greater then the actual number of files to be read in, then the size of the map cache is automatically set to the lower amount. Generally, if a map remains in the map cache, then it will not have to be reread from the disk file.

4.2.5 Optimaizing speed and memory

You only need to be read this section if you find yourself running into either memory limitations, or need to get the program to run faster.

By far, the most substantial savings in both time and memory can be gained by adjusting the cache size with option `-c`. If, for example, you will be calculating the distance ratios between 120 maps, you can acheive a much faster run by increasing the size of the map cache. If the maps are large (say 100 actual[1] nodes each) then you may well find that even the default map cache size leads to memory problems on some limited machines.[2] To date, I have not heard reports of users needing to adjust the default for their runs, though I do know of one case where running on 111 maps (of 10 nodes each), increasing the cache size to 70 made a dramatic improvement in speed.

[1] That is the actual number of nodes in a map, as distinct from the question of the highest possible node number, is what matters here.

[2] The distributed versions for DOS can only make use of native DOS memory (something I consider a design error of DOS and not of the programs). However, if you have a 32 bit DOS machine, then you can obtain versions of the program that will make use of all free memory on your machine. If you are running on a machine with a real operating system, (ie, one which gives programs the memory they ask for if the memory is available without asking the programs to jump through a few hoops first) then you are less likely to run into memory problems.

Adjusting the options `-m` and `-n` downward to reflect the actual size will have an extremely small improvement on both time and memory, but this really will be so small as to not be worth it. A small, but far greater improvment in memory can be achived by keeping the file names in the list file short. And those files will be read from the disk much faster if they are in the same directory that distrat is called from.

Finally, I would be interested to hear of any cases in which users have felt the effect of machines limits running these programs. Writing a program to perform the calculation was trivial, having it do so making good use of the computer's resources was the interesting (and fun) bit.

4.3 The output

distrat writes its output to the screen (standard output) and might look like

```
DR(hun01, hun02) = 0.1234
DR(hun01, hun03) = 0.5678
DR(hun02, hun03) = 0.9012
```

4.3.1 Order of the output information

Note that the the ordering of distance ratios will be an upper right corner of a distance matrix. So if comparing maps 1 through 6 (as the order in the list file) then the Nth distance ratio will fit according to the following table

	1	2	3	4	5	6
1		1st	2nd	3rd	4th	5th
2			6th	7th	8th	9th
3				10th	11th	12th
4					13th	14th
5						15th
6						

A more general statement of the output order is

$$\begin{array}{cccccc}
(1,2) & (1,3) & (1,4) & \ldots & (1,n-1) & (1,n) \\
 & (2,3) & (2,4) & \ldots & (2,n-1) & (2,n) \\
 & & (3,4) & \ldots & (3,n-1) & (3,n) \\
 & & & \ddots & \vdots & \vdots \\
 & & & & (n-2,n-1) & (n-2,n) \\
 & & & & & (n-1,n)
\end{array}$$

That is, it does the first with the second, then the first with the third, first with the fourth and so on until it does the first with the last, then it does the second with the third, the second with the forth and so on until it does the second with the last, then the third with the forth, and so on.

The notion of 'first', 'second' and so on has to do with the order in which the files are listed in the `listfile`.

4.3.1.1 Getting the ouput in order

Most software packages which input distance data seem to prefer input as lower left instead of upper right. Also, they like to have the zero's down the diagonal. If you wish to sort the output in this way, there are a number of things you can do.

- One way is to use the drsort program which is discussed in more detail in chapter 6. It has been designed exactly for this purpose.
- Another is to use existing sorting tools. This requires some planning ahead, in that it means that the order of the maps in list file should match the sorting order of the map id's. That is, if the map in file xxx has the id Jim-Jones and the map in file yyy has the id John-Smith, then you can convert the file from an upper right triangle by using your sorting program to sort the lines in your output file with the primary sort on the second field and the secondary sort of the first field. If your sorting software can't do this, it is time to get a new sorting program.

Chapter 5

The askmap program

The askmap program is probably the most difficult to document, and the documentation that you see for it here is not as complete as the documentation for the other programs. However, askmap is an interactive program which prompts the user for input and will do its job as required. The askmap program is inflexible in a number of ways, but for normal use it can be extremely helpful. It is the program what will should be changing the most radically from release to release.

It must be reiterated here that these programs along with this documentation do not substitute in any way an understanding of the relevant sections of Markóczy and Goldberg (1995). The askmap program assists with the elicitation of the causal map from the subject, after the subject has selected her/his top constructs.[1] *The program is not designed to for self-administration!* It is designed to save the researcher the trouble finding the next pair of questions and of writing down the answer.

The program does no automatic back-up of data entered. Always use audio recording (with the subject's permission) of the interview to recover data in case of power failure, computer problem or problem with askmap.

5.1 What it does

Askmap will prompt the interviewer for a file name for the created map to be stored in, for the `Map id` (see section 3.3), for the number of constructs (nodes) in the map, and then for the number of each node, and, optionally, a brief text for each node.

[1] If you don't know what this means, then stop now and read the paper.

Once it has all of that information, it will then prompt for pairs of influencing relations, which should be entered as integers in the range of possible values[2]. It uses so-called Ross ordering to determine which pair of pairs should be asked next. When it is done, it will write out a file in the form described in section 3.3. That file can be used as a data file for other programs (such as distrat) without any other processing of the file being needed.

The best way to see who the program works is to give it a small try now. First select a small number of constructs, say 5, from some list or other, and then simply call up the program,

```
askmap
```

and follow the instructions.

5.2 Correcting input errors

If something is input by error, or if the subject changes an opinion, there is nothing you can do with the program after the information has been entered. The program is *not* a cause map editor. In most cases you can note the error on a sheet of paper (have it ready before you start askmap) and correct it by editing the created file after all is done. Even with askmap you will not get away with failing to read section 3.3.

5.3 Saving work in progress and quiting

When askmap prompts for an influence between a pair of nodes A and B it will ask a pair of questions. The first will be the influence of A on B and the second will be an influence of B on A. When it asks for the first pair, you may, at your option enter the the letter s.

```
...
4th pair 30% done
"three" ---> "two"
Influence of node 3 on 2 [-3 to 3]: s
Saving map in file askmap.tmp
4th pair 30% done
...
```

[2] The current version of the program prompts for input in the range $[-3, +3]$ although it will accept any small integer value. Something may be done to make the this prompt more flexible in future versions, but at the moment the moment ignore the values in the prompt if you wish to use another range of numbers.

which saves the work so far in a file called `askmap.tmp`. If you have a file already by that name, it will be overwritten; so you should never start askmap if you have important information only in that file.

It is highly recommended that at any pause or interruption during the interview you do use the `s` command.

5.3.1 Quitting

At exactly the same time as the `s` command can be used, the `q` command, for "quit" can also be used. This will first ask you whether you want to same the file, and then it will quit. Note that here too, the information will be saved to a file called `askmap.tmp`. You should immediately copy that file to a different name.

At the moment the program does not allow you to save or quit at any other time. It is only at the first influence of a pair of nodes.

If you need to quit after the first of a pair of questions has been answered, then provide a fake answer to the second of the pair, and quit at the begining of the next pair. You will have to make a change to the file `askmap.tmp` described below.

5.3.2 Restarting with a partial map

If you have saved a map in `askmap.tmp` and copied it to, say `saved.tmp` you can restart askmap with

```
askmap -r saved.map
```

Askmap will ask you, again, for the name of the file where the completed map is to be saved, and it will ask again for the optional text that goes with each node since none of this information is actually saved in the created map or `askmap.tmp`. It will then restart asking pairs from the point at which you last saved.

It knows where to restart because the last line of `askmap.tmp` has a line that looks like

```
AM-Recover-from-pair: N
```

where N is the N^{th} pair to in the Ross ordering, and should be the last pair actually completed. So, if you save and quit on the eigth pair, N will be 7.

If you had to add a fake answer in order to quit, you should edit the file with the partial map in it to reduce the point of recovery by one. So, if you have completed the first half of pair 10, and had to provide a fake answer to get to the next pair so that you could save and quit; then you should edit `askmap.tmp` (or whatever name you copied it to) so that the last line will have `9` instead of `10` as the pair to recover from.[3]

5.4 Using a different language

The messages which askmap writes to the user can appear in any language in addition to English for which there is a *`language`*`.aml` file, where *`language`* is the name of the language. With this distribution, the files `english.aml` and `magyar.aml` (Hungarian) are included. The file `english.aml` is never needed since English responses are built into the program, but is provided as a basis for translation if you wish to create your own `.aml` file.

One would call askmap to run in Hungarian with the following command:

```
askmap -L magyar
```

In the current version, the file `magyar.aml` must be copied to the directory you call askmap from. This is an unfortunate limitation, but it will remain until some further release.

5.4.1 How to create your own .aml file

Recall that this program is not intended for self-administration of the causal map elicitation technique; but it is still useful to have the prompts from the program in the same language that you are using with the subject. Creating an `.aml` file for use with askmap will require a fair bit of technical expertise, and—as with all translations—more time than you might expect.

[This section not written yet. Contact the authoer if you wish to create your own `.aml` file.]

5.5 Limits and things to improve

5.5.1 Limits

- Askmap does no checkpointing or intermediate back-ups. A power failure, program bug or whatever could very easily lead to the loss of data. Always audio record data elicitation sessions as back-up.
- Askmap can query maps with no more than 181 elements due to limits in the Ross ordering routine. See chapter 10 for more information about this.
- Most strings (map id, texts which go with nodes, etc) are limited in length to 40 characters each.

[3] It would probably have been easier for me to fix the program than to document this.

5.5.2 Things to improve

This is a list of things that I know should be improved and may be in future versions (but I need to hear from users exactly how annoying the current limits are).
Regarding basic operations:

- More error checking of responses to make sure that reasonable responses are given. (Currently, sanity checks on input are sporadic.)
- The Ross ordering routine is wasteful of memory because I don't have a nice definition of the appropriate function in a computer friendly way. It should be rewritten entirely.
- Signal processing (so things like Ctrl-C would work) should be added.
- Add option for suppressing page clears between querying pairs.

There are things to improve regarding language facilities.

- Set things up so that the `.aml` files do not have to be in the current directory.
- Allow environment variable for language.
- Switch from octal coding of extended ASCII to decimal.
- Character-set dependence/assumptions stated in `.aml` files instead of being built in.
- Solution to identical first letter for "yes" and "no" in some language should be found.

Things need to be done regarding the way partial files are saved and askmap is restarted.

- Allow saves (and quits) half way through pairs.
- Save (and restore) node text information.
- Check file overwrite for `askmap.tmp`
- Allow user specified save file.

Chapter 6

The drsort program

The drsort program is designed to transform the output of the distrat program discussed in chapter 4, so that that output could be used as input to a variety of programs for further (usually statistical) analysis.

The distrat program produces its output one distance ratio per line. The details are discussed in section 4.3 which should be reviewed before proceeding with this chapter. That is, the problem that drsort was create to solve is described there and will not be repeated here.

For the example that follow, we will work with a rather contrived output file from distrat. This distrat output is listed in figure 6.1. Note that this sample file is contrived so that the distance ratio between test-N and test-M bares a striking relation to N and M.

The `-h` option will provide a brief usage message about the command line options

```
drsort: Copyright Jeff Goldberg 1994
Usage:
        drsort [-lufezh] [-d N]
                -h      this message
                -l      Lower left corner [ON]
                -u      Upper right corner [OFF]
                -f      Full matrix (implies -z) [OFF]
                -z      Zero's in diagonal [ON]
                -e      Empty diagonals [OFF]
                -n      Print variable names [OFF]
                -c      Print number of variables [OFF]
                -d N    Debug info level [0]
```

Figure 6.1 Contrived distrat output and distrat input

```
DR(test-01, test-02) = 0.0102
DR(test-01, test-03) = 0.0103
DR(test-01, test-04) = 0.0104
```
[...4 lines omitted...]
```
DR(test-01, test-09) = 0.0109
DR(test-01, test-10) = 0.0110
DR(test-02, test-03) = 0.0203
DR(test-02, test-04) = 0.0204
```
[...5 lines omitted...]
```
DR(test-02, test-10) = 0.0210
DR(test-03, test-04) = 0.0304
DR(test-03, test-05) = 0.0305
```
[...23 lines omitted...]
```
DR(test-08, test-09) = 0.0809
DR(test-08, test-10) = 0.0810
DR(test-09, test-10) = 0.0910
```

As you can see there are many option, but you won't need most of them.

The program reads from standard input (You should review section 3.2 if you don't understand this or what the '<' and '>' symbols mean in the following examples.) For example if we take figure 6.1 to be the contents of the file `shortdr.out` then a command like

```
drsort < shortdr.out
```

will write to standard output:

```
0.0000
0.0102 0.0000
0.0103 0.0203 0.0000
0.0104 0.0204 0.0304 0.0000
0.0105 0.0205 0.0305 0.0405 0.0000
0.0106 0.0206 0.0306 0.0406 0.0506 0.0000
0.0107 0.0207 0.0307 0.0407 0.0507 0.0607 0.0000
0.0108 0.0208 0.0308 0.0408 0.0508 0.0608 0.0708 0.0000
0.0109 0.0209 0.0309 0.0409 0.0509 0.0609 0.0709 0.0809 0.0000
0.0110 0.0210 0.0310 0.0410 0.0510 0.0610 0.0710 0.0810 0.0910 0.0000
```

As you can see, by default the output is a lower left corner with zeros in the diagonal. If you wanted to save this output, let's say into a file called `statdist.dat` you could do something like:

```
drsort < shortdr.out > statdist.dat
```

And now the file statdist.dat will contain the output which you can examine with with `cat` or `more` (or `TYPE` on DOS).

On systems where using pipes makes sense you can pipe the out put of distrat directly to drsort

```
distrat -f list | drsort
```

which would perform the distance ratio calculation and present it neatly arranged to standard output. Of course that output could also be redirected

```
distrat -f list | drsort > statin.dat
```

And now `statin.dat` contains the matrix.

The program will only accept input which is in the form of the output of the distrat program.

6.1 Different output forms

By default, drsort gives a lower left triangle with zeros in the diagonal and no other information. If you wish to change that you need to use one of the options.

6.1.1 Upper right corner: option -u

If you want an upper right triangle, use the `-u` option

```
drsort -u < shortdr.out
```

yields

```
0.0000 0.0102 0.0103 0.0104 0.0105 0.0106 0.0107 0.0108 0.0109 0.0110
       0.0000 0.0203 0.0204 0.0205 0.0206 0.0207 0.0208 0.0209 0.0210
              0.0000 0.0304 0.0305 0.0306 0.0307 0.0308 0.0309 0.0310
                     0.0000 0.0405 0.0406 0.0407 0.0408 0.0409 0.0410
                            0.0000 0.0506 0.0507 0.0508 0.0509 0.0510
                                   0.0000 0.0607 0.0608 0.0609 0.0610
                                          0.0000 0.0708 0.0709 0.0710
                                                 0.0000 0.0809 0.0810
                                                        0.0000 0.0910
                                                               0.0000
```

6.1.2 Full matrix: option -f

If you want a full symmetrical matrix, use the -f option

```
drsort -f < shortdr.out
```

produces on the standard output:

```
0.0000 0.0102 0.0103 0.0104 0.0105 0.0106 0.0107 0.0108 0.0109 0.0110
0.0102 0.0000 0.0203 0.0204 0.0205 0.0206 0.0207 0.0208 0.0209 0.0210
0.0103 0.0203 0.0000 0.0304 0.0305 0.0306 0.0307 0.0308 0.0309 0.0310
0.0104 0.0204 0.0304 0.0000 0.0405 0.0406 0.0407 0.0408 0.0409 0.0410
0.0105 0.0205 0.0305 0.0405 0.0000 0.0506 0.0507 0.0508 0.0509 0.0510
0.0106 0.0206 0.0306 0.0406 0.0506 0.0000 0.0607 0.0608 0.0609 0.0610
0.0107 0.0207 0.0307 0.0407 0.0507 0.0607 0.0000 0.0708 0.0709 0.0710
0.0108 0.0208 0.0308 0.0408 0.0508 0.0608 0.0708 0.0000 0.0809 0.0810
0.0109 0.0209 0.0309 0.0409 0.0509 0.0609 0.0709 0.0809 0.0000 0.0910
0.0110 0.0210 0.0310 0.0410 0.0510 0.0610 0.0710 0.0810 0.0910 0.0000
```

6.1.3 Empty diagonals: option -e

If you want the diagonal to be empty, use the `-e` option

```
drsort -e < shortdr.out
```

produces

```
0.0102
0.0103 0.0203
0.0104 0.0204 0.0304
0.0105 0.0205 0.0305 0.0405
0.0106 0.0206 0.0306 0.0406 0.0506
0.0107 0.0207 0.0307 0.0407 0.0507 0.0607
0.0108 0.0208 0.0308 0.0408 0.0508 0.0608 0.0708
0.0109 0.0209 0.0309 0.0409 0.0509 0.0609 0.0709 0.0809
0.0110 0.0210 0.0310 0.0410 0.0510 0.0610 0.0710 0.0810 0.0910
```

The `-e` and the `-u` option can be used together to get an upper right corner with empty diagonals.

6.1.4 Map names: option -n

The option `-n` provides the map ids across the top:

```
drsort -n < shortdr.out
```

produces something like (with bits left out)

```
test-01 test-02 test-03 test-04 test-05 test-06 test-07...
 0.0000
 0.0102 0.0000
 0.0103 0.0203 0.0000
 0.0104 0.0204 0.0304 0.0000
 ...
```

6.1.5 Map count: option `-c`

If you wish to have the number of maps also produced, then use the `-c` option.

```
10
 0.0000
 0.0102 0.0000
 0.0103 0.0203 0.0000
 0.0104 0.0204 0.0304 0.0000
 ...
```

It is particularly useful in combination with the `-n` option. When they are both used, the count (number of maps) is always written before the names of the maps.

6.1.6 Conflicting options

If you try to use conflicting options (`-f` and `-u` for example) the result will be unpredictable. Actually the results are fully predictable; it does the same thing each time. What "unpredictable" really means in software documentation is that the result might change in future releases, so you shouldn't count on any particular behavior here.

6.2 Time and Space

If you are sorting a large number of ratios, drsort may run out of space. Here's why:

As someone pointed out, time and space are the Yin and Yang of programming. This program tries to find its way through this in a reasonable manner. The amount of space for floating point numbers (floating point units, F) reserved by the program at its point of maximal memory usage is: $F = p + m - 1$ where m is the number of maps, and p is the number of distance ratios between maps (ie, $p = (m^2 - m)/2$); so F works out to be $(m^2 + m)/2 - 1$.

The number of memory pointers, P, at this maximum time will actually be nearly the same: $P = p + m$.

So the total maximum memory usage M of the program will be

$$M = S_r + (S_f + S_p) \times (m^2 + m)/2)$$

where S_r is the actual size of the program, S_f is the size of a floating point unit, and S_p is the size of a memory pointer on the machine. A typical size of these things on many machines would be for S_f to equal 4 bytes, and S_p equal 4 bytes and for the size of the program, S_r, to be about 30 thousand bytes. point number to require 4 bytes and for each pointer to require 4 bytes and for the program to be about 30 thousand bytes. So, under such a configuration memory usage is

$$M = 30000 + 4(m^2 + m)$$

in bytes.

So for 100 maps about 70K of memory is needed. While for 350 maps about 522K of memory would be needed. This strikes me as a sufficiently large number, since on systems with restricted memory you would probably not be able to get your statistical package to work on that much data anyway. Yet you should be aware that the memory requirements go up with the sqaure of the number of maps. If you do need to use drsort on output from large numbers of maps, then do get in touch with the author who can provide you with versions of the program that will use all available memory on your machine.

It would have been possible to write the program to use less memory, but would have either required the user to tell the program how many maps were used or have the program make two passes over the input (the first time through to count its length).

6.3 Notes

This program will not be built in the the distrat program since it would force distrat to store all its output. The use of the pipe (`distrat` *args* `| drsort`) should be sufficient.

Chapter 7

The density program

The density of a map is the ratio of arcs to places where there could be arcs. For example, in a 5 node map in which we do not consider reflexive arcs (arcs from a node to the same node) there are 10 places were there could be arcs. If there are ten arcs within such a map, then the map has a density of 1.0, if there are none, then the density is 0.0. If the map has, say, six arcs then the density is 6/10, or 0.6. Density is not used or discussed by M&G95, but the program was trivial to write once the others were written.

Usage and input files are as will other programs in the package. So run it with the `-h` option of a list of command line options:

```
Usage: C:\BIN\DENSITY.EXE [options] file
   or  C:\BIN\DENSITY.EXE [options] -f listfile
        files contains a map
        listfile contains a list of files
        Options:
                -h      display this message
                -v      set verbose flag [off]
                -d num  set debug flag to num [0]
                -n num  set max. distinct nodes [300]
                -m num  set max. maps in list [200]
                -t num  set type of map file [1]
                -A num  set alpha to num [1]
```

All of the command line options here have exactly the same meaning that they have for the distrat program. So review the options described in 4.2 on page 39 for a complete description.

The output of the program is fairly straight forward.

```
  density tun01
```

produces

```
Density(TUN-0792-01) = 0.7333
```

Chapter 8

The iodeg program

The iodeg program is used to calculate the *in degrees* and *out degrees* for various nodes in a map.

These concepts are defined and used in several cause mapping papers (e.g., Bougon, Weick and Binkhorst, 1977; Ford and Hegarty, 1984). In addition to the standard senses of these terms, the iodeg calculates not only the number of arcs to each node, but also sums up the absolute values of their strengths. This additional type of information was used by Ford and Hegarty (1984) and by Markóczy (1995). In and out degree calculations are not described in M&G95.

For example, if the map `tun01.map` looks like the map in figure 8.1 then the output of `iodeg tun01.map` will look like:

```
In-degrees for map TUN-0792-01:
    2:     7  (+17.000)
    6:     9  (+23.000)
   13:     6  (+15.000)
   27:     5  (+13.000)
   36:     9  (+22.000)
   37:     9  (+26.000)
   41:     8  (+18.000)
   43:     3  (+6.000)
   47:     7  (+16.000)
   49:     3  (+7.000)
Out-degrees for map TUN-0792-01
    2:     6  (+12.000)
    6:     8  (+23.000)
   13:     7  (+17.000)
   27:     6  (+13.000)
```

Figure 8.1 Map in file `tun01.map`

```
Map id: TUN-0792-01
10
 2  6 13 27 36 37 41 43 47 49

 0  2  0  0  3  3  1  1  2  0
 0  0  3  3  3  3  3  2  3  3
 3  3  0  0  1  3  3  3  1  0
 2  2  0  0  2  3  2  0  2  0
 3  3  2  3  0  3  1  0  0  0
 3  3  0  3  2  0  3  0  0  2
 3  2  3  0  3  3  0  0  3  2
 0  2  3  0  2  3  3  0  3  0
 1  3  3  2  3  3  2  0  0  0
 2  3  1  2  3  2  0  0  2  0
```

```
   36:     6  (+15.000)
   37:     6  (+16.000)
   41:     7  (+19.000)
   43:     6  (+16.000)
   47:     7  (+17.000)
   49:     7  (+15.000)
```

The first column is the node number, the second is the in-degree or out-degree of the node (number of arcs leading to it, number of arcs leading from it). The third number is the sum of the absolute values of the weights of the arcs leading in (or out).

The options for the program can be found by using the `-h` option, which will tell you the following.

```
Usage: iodeg [options] file
   or  iodeg [options] -f listfile
        files contains a map
        listfile contains a list of files
        Options:
                -h      display this message
                -v      set verbose flag [off]
                -d num  set debug flag to num [0]
                -n num  set max. distinct nodes [300]
```

```
-m num  set max. maps in list [200]
-t num  set type of map file [1]
-A num  set alpha to num [1]
```

The meaning of all of the options here are identical to the options for the distrat program described fully in section 4.2 on page 39.

lodeg does not sort the output. If you want it sorted or transformed in some systematic way, that is up to you. Any good text editor will be able to perform most transformations as will programs designed for such purposes such as perl, awk, and sed, which are standardly available on Unix systems and are also freely available for DOS from standard internet archives.

Chapter 9

The avmap program

The avmap program performs the calculation described in M&G95 [pp. 320–321] as the way of creating so-called *central maps.*

To get the average (central) map for the three maps `tun01`, `tun02` and `tun03` use the command:

```
avmap tun01 tun02 tun03
```

which should result in

```
Map id: Average-map
7
     8     27     36     37     40     41     47

 0.000 -2.000 -1.000  3.000 -3.000  0.500  1.500
 1.000  0.000  2.000  3.000  0.000  1.500  2.500
 0.000  3.000  0.000  3.000  0.000  0.500  0.500
 1.000  3.000  2.000  0.000 -1.000  3.000  1.500
-2.500  0.000  1.000  3.000  0.000  0.500  2.000
-0.500  1.000  1.000  3.000  1.000  0.000  3.000
-0.500  2.000  2.000  3.000  0.500  2.333  0.000
```

As with distrat and most of the programs here, the `-f` option can be used so that the file names come from a list. So `avmap -f list` yields,

```
Map id: Average-map
3
    36     41     47

 0.000  0.143  1.200
```

```
 0.000  0.000  3.000
 2.200  2.600  0.000
```

You can use normal command line redirection (see section 3.2 begining on page 28) to save the results in a file:

```
 avmap tun01 tun02 tun03 > tun123.avr
```

places the output map into the file `tun123.avr`, which can then be used as a map file as input for other programs.

For the usual usage message use the `-h` option:

```
Usage: C:\BIN\AVMAP.EXE [options] file [files]
   or  C:\BIN\AVMAP.EXE [options] -f listfile
        file and files contain maps
        listfile contains a list of files
        Options:
                -h      display this message
                -v      set verbose flag [off]
                -d N    set debug level flag [0]
                -p num  set cutoff percent to num [50]
                -n name set map id to name [Average-map]
```

9.1 The options

9.1.1 Changing the output map-id

One of the most important options are the `-n` option which allows you to give the created map an id distinct from "Average-map". For example

```
 avmap -n TUN-averages -f list
```

yields

```
Map id: TUN-averages
3
    36     41     47

 0.000  0.143  1.200
 0.000  0.000  3.000
 2.200  2.600  0.000
```

Note that the here the map id of the output corresponds to what was specified as the argument to the `-n` option on the command line.

Naturally you should make the name something which is a legal string for a map id (see the map file format section, section 3.3 on page 33). Also, if you use characters that may mean something special to your command interperter, then you will need to quote the name or otherwise escape the special meanings. I can't tell you exactly what these characters are or what to do because that depends on your command interperter.

9.1.2 Changing the cut-off percentage

The other important option is `-p` which sets the minimum percentage of maps that a node must be in to appear in average. The default is 50. By using a lower percentation we can get larger maps.

```
avmap -p40 -flist
```

produces

```
Map id: Average-map
4
   27     36     41     47

 0.000  1.800  1.000  1.750
 2.600  0.000  0.143  1.200
 0.750  0.000  0.000  3.000
 1.500  2.200  2.600  0.000
```

9.1.3 The `-v` and `-d` options

The options `-v` and `-d` are there to provide you with debugging information about intermediate calculations.

9.2 Empty and single node maps

It should be noted that avmap can produce maps with zero nodes or one node as output. Such maps are not actually acceptible as input maps, since using them could lead to division by zero in certain cases. When avmap produces such a map it writes a warning message to the screen.

Chapter 10

The **ross** program

The ross program simply displays the Ross ordering of on N elements. The ordering was developed by Ross (1934). It is an ordering of pairs of elements such as to maximize the distance between pairs involving the same elements.

The program, as an independent program, is not needed directly for causal map analysis or elicitation. The basic calculation is needed for the askmap program (chapter 5), and once that was written, it took no more that a few minutes to write a stand alone program.

It's usage is simple:

```
ross 7
```

will produce the Ross ordering for seven elements:

```
1: 1 2
2: 7 3
3: 6 4
4: 5 1
5: 3 2
6: 4 7
7: 5 6
8: 1 3
9: 2 4
10: 7 5
11: 6 1
12: 4 3
13: 5 2
14: 6 7
15: 1 4
```

```
16: 3 5
17: 2 6
18: 7 1
19: 4 5
20: 3 6
21: 2 7
```

10.1 Limitations

10.1.1 Limitations on N

Running

```
 ross 200
```

produces

```
rossify(): must be no more than 181 elements
Error getting ross array
```

The 181 is not an arbitrary number (It is the greatest integer N such that $(N^2 - N)/2 \leq 65536/4$. On some computers the largest single array you can request is 65536 bytes without writing machine and compiler specific code, and 4 is the maximum size of the element of such an array on such a machine.) Due to the fact that the essential routine used for generating the Ross ordering is actually pretty stupid (and stores the whole array in memory) this limit exists.

Unless you have found a use of Ross ordering that is new to me, then it is unlikely that you will need it for more than 30 elements. For the purposes of Ross ordering, if you really want more that 181 elements, you could safely give up on the Ross ordering, and just order your pairs randomly.

10.1.2 No `-h` option

The ross program doesn't offer a usage message.

Appendix A

GNU General Public License

A.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

A.2 GNU General Public License Terms and Conditions for Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

(a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

(b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

A.3 No Warranty

11. **Because the program is licensed free of charge, there is *no warranty* for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.**

12. **In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a**

failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

End of Terms and Conditions

A.4 How to apply these terms to your new programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what
it does.> Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version
2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public
License along with this program; if not, write to the Free
Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139,
USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome to
redistribute it under certain conditions; type `show c' for
details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in
the program 'Gnomovision' (which makes passes at compilers)
written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of
Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix B

Other software tools

This appendix, when it is written, will list contact information for Mauri Laukkanen's CMAP2 package (Laukkanen, 1990; Laukkanen, 1992), and Colin Eden and associates Graphics Cope (Ackermann, Cropper and Eden, 1992; Eden, Ackermann and Cropper, 1992). Some other information will listed to point to influence diagramming software as well.

Bibliography

Ackermann, Fran, Steve Cropper and Colin Eden (1992). 'Graphics COPE users guide', *Report*, Department of Management Science, University of Strathclyde, Glasgow.

Bougon, Michel G., Karl E. Weick and Din Binkhorst (1977). 'Cognition in organizations: An analysis of the Utrecht jazz orchestra', ***Academy of Management Review*** **22**: 606–639.

Eden, Colin, Fran Ackermann and Steve Cropper (1992). 'The analysis of cause maps', ***Journal of Management Studies*** **29**(3): 309–324.

Eden, Colin and Sue Jones (1984). 'Using repertory grids for problem construction', ***Journal of the Operational Research Society*** **35**: 779–790.

Ford, Jeffrey D. and W. Harvey Hegarty (1984). 'Decision makers' beliefs about the causes and effects of structure: an exploratory study', ***Academy of Management Journal*** **27**: 271–291.

Langfield-Smith, Kim M. and Andrew Wirth (1992). 'Measuring differences between cognitive maps', ***Journal of the Operational Research Society*** **43**(12): 1135–1150.

Laukkanen, Mauri (1990). 'Describing management cognition: The cause mapping approach', ***Scandinavian Journal of Management*** **6**(3): 197–216.

Laukkanen, Mauri (1992). ***Comparative cause mapping of management cognitions: A computer database method for natural data***, Helsinki School of Economics and Business Administration Publications, Helsinki.

Markóczy, Lívia (1995). ***Barriers to shared belief: The role of strategic interest, managerial characteristics and organisational factors***, Phd thesis, University of Cambridge, Cambridge.

Markóczy, Lívia and Jeff Goldberg (1995). 'A method for eliciting and comparing causal maps', ***Journal of Management*** **21**(2): 305–333.

Press, William H., Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling (1988). ***Numerical Recipes in C: The art of scientific computing***, Cambridge University Press, Cambridge, UK.

Ross, Robert T. (1934). 'Optimum orders for the presentation of pairs in the method of paired comparison', ***Journal of Educational Psychology*** **25**: 375–382.

Wall, Larry and Randel L. Schwartz (1991). ***Programming Perl***, Nutshell Handbook Series, O'Reilly & Associates, Sebastopol, Calif.