Open Cluster Data Base

Version 2.0

(User's Guide)

J.-C. Mermilliod

Institut d'Astronomie de l'Université de Lausanne
CH - 1290 Chavannes-des-Bois / Switzerland

E-mail: `mermio@scsun.unige.ch`

July 15, 1994

Abstract

The database for stars in galactic open clusters (BDA) has been developed since 1987 at the Institute for Astronomy (University of Lausanne). The extensive collection of observational data covers most significant domains and concerns about 100000 stars in some 500 NGC, IC and anonymous clusters. This includes measurements in most photometric systems in which cluster stars have been observed, spectroscopic observations, astrometric data, various kinds of useful information and extensive bibliography. Maps for about 180 clusters have been scanned and included in the database. The greatest effort has been spent in solving the identification problems raised by the definition of so many different numbering systems.

The database not only aims at storing data but also at offering a versatile working environment covering many aspects of the study of open clusters. It provides tools to compare the data and plot photometric diagrams. It is most suitable for use on a workstation. Independently of the application software proposed, the main utility of the database lies in the extensive data collection brought into uniform numbering systems. The BDA presents a clear report of the present status of observations.

The following document presents the general philosophy of the data organisation and retrieval. It explains with many examples how to use the database.

Contents

Part I

DATABASE DESCRIPTION

Chapter 1

Introduction

1.1 History

About 1200 galactic open clusters are known and approximately half of them have been observed so far, in at least one photometric system. The number of stars per cluster goes from a few tens for the poorest objects, to several thousands for the most prominent clusters.

Modern observations of open clusters developed very rapidly after the definition of the UBV photoelectric system (Johnson & Morgan 1953). These observations produced a number of colour-magnitude (Hertzsprung-Russell) diagrams which made fundamental contributions to the understanding of stellar evolution. At the same time photographic photometry allowed to observe larger areas and reach fainter stars. Additional information, mostly spectroscopic, was gradually obtained, first for stars in the nearby clusters and later in more distant clusters, thanks to the existence of larger telescopes and more efficient detectors. Two-dimensional detectors are best adapted for the observations of star clusters and CCD observing is today becoming the preferred technique. It replaces both the photoelectric and photographic photometry.

Data compilations started already in 1972 at the Institute for Astronomy (University of Lausanne, Switzerland). Mermilliod (1976a) published a first catalogue of UBV photometry and MK spectral types in open clusters. The third version was announced ten years later (Mermilliod 1986a). The systematic determination of cross-references between the many numbering systems in a cluster was the basic work which made the realisation of the data collections possible. Several catalogues were distributed by the Strasbourg Data Center (CDS). The files remained on magnetic tapes until the installation of Unix workstations and large disks in our institute made it possible to keep the data on-line. These compilations were discontinued in their older form and the data were organised in a database designed in March 1987 (Mermilliod 1988a, 1988b).

1.2 Need for a database

The quantity and variety of observations accumulated on about 100000 stars in open clusters are quite impressive and sufficient to motivate the development of a specialised database. Data analysis in open clusters is quite complex and extensive information are often necessary, but are difficult to gather because of the multiple star designations created so far. The star identification problem, already discussed (Mermilliod 1972, 1973, 1976b, 1979a) is not specific to star clusters, but takes on a more accute form. The variety of stars identifications asks for an unavoidable amount of extra work to collect the data available for any star. The necessity to keep on-line track of the cross-references between various numbering systems may be one of the motivations for developing a database.

The database offers one solution to this problem and a coherent data storage and retrieval. The fact that data are kept under a stable format allows to develop tools more easily. It is also a deposit for unpublished data, although the development of anonymous ftp servers has reduced the interest of the database to preserve unpublished data.

1.3 Access mode

The database is maintained on a Sun Sparc worksation and its total size is about 35 MB for a total of about 6000 files. Due to the specific structure of the database, it is presently more convenient to have a copy of it on a workstation.The dissemination of various copies may raise the problem of the simultaneous existence of several revisions. However, this problem is solved by the transfer of the modified or new data files grouped in a tarfile over the Internet. Such a system is already used to maintain a copy of the database at the Meudon observatory in Paris on a DEC workstation, which offers additional possibilities for distribution to other machines. Access through the World Wide Web is being examined. Hypertext description is being written and some database consultation will be provided. There is also a project to access the database through the ADS facility.

1.4 User's guide overview

The first part presents a database overview and describes the concept of datatype and the organisation of the bibliography (Chapter 2), the database structure in directories and the file organisation (Chapter 3), the different query modes that have been considered for BDA (Chapter 4) and the graphics facilities (Chapter 5).

Chapter 6 discusses the detail of the database installation from tape and the various definition files.

The second part is divided in four chapters which explain how to use the basic commands and the various query modes: command, menu, prompt and graphical user interface.

The third part presents applications provided by the database and discusses the underlying philosophy. It also explains how to use them. The use of the graphic program which generates the photometric diagrams is explained here.

The Appendix presents a summary of the command and program names, option meaning, data types, file names, field names and aliases. It also describes the catalogue of red giants in open clusters.

Chapter 2

Database Overview

2.1 Basic concepts

2.1.1 Database concept

The database structure results from the history of data compilation, the constraints of existing application software and mostly reflects the way the research using the database is viewed.

The database has been built not only to query information for one or a few stars, but mainly to study, in a general sense, open clusters. This includes data comparaisons, data elaboration as well as evaluation of cluster properties. The latter analysis relies basically on membership determination.

Therefore clusters, rather than stars, form the basic unit of the database, and each cluster has its own directory and contains the relevant data. The various logical steps of the cluster study and the necessity of starting from the original data justify the adopted structure.

2.1.2 Data types

One important concept in BDA is that of datatype. A datatype designation has been attributed to each kind of data. This designation is generally similar to acronyms currently used, like, for example: **ubv, mk, uvby, jhk**, and so on. The datatype designation is described in Appendix C.

The knowledge of the datatype is important because it is always used for querying the database. The datatype may be an argument to commands, like in `list ubv`, but more generally, many command names are identical to the datatype designation. For example, the command to get the UBV data for some stars, let's say star # 1 2 3, will be simply: `ubv 1 2 3`.

This syntax and the fact that the data are collected in distinct files hide the name of the file that contains the desired data and the field names in each record.

2.1.3 The basic scheme

The analysis of photometric diagrams, and especially the colour-magnitude diagrams that can be built in several photometric systems, remains the key method to determine the cluster parameters (reddening, distance and age). The study of a cluster necessitates to have also easily available all complementary information, remarks and bibliography. This is why everything is organised in the cluster directories.

If the production of a colour-magnitude diagram seems rather easy, the real situation is unfortunately not simple and the data are not always of sufficient quality. Therefore a large amount of work is necessary to plot a reliable colour-magnitude diagram. The various steps involved are:

1. the census of the stars in the cluster field, to determine the completeness in terms of both limiting magnitude and surface coverage;

2. the comparison of the various sources of data;

3. the selection of cluster members. This is certainly the main problem and no straightforward solution has yet been found. The other data contained in the database, like spectral types, radial velocities, remarks, and so on, may help in assigning cluster membership;

4. with the best selected data, it is eventually possible to plot a nice colour-magnitude diagram and determine more accurate cluster distances and ages.

The database cross-reference tables and the collection of rectangular positions are useful to achieve the first step. Tools to perform the second step and compute mean values are also provided.

2.1.4 Star numbering

Star numbering is probably the most serious problem to solve to build a coherent database. The solution adopted is based on earlier work which was undertaken to include photometric data on star clusters into catalogues prepared for the Strasbourg Data Center (Mermilliod 1972, 1973, 1976b, 1979a).

One numbering system has been adopted for each cluster and it provides a unique identificator for each star which is used to register the various data. Therefore the data recorded under the numer #1 in each data file always belong to the same star, i.e. star #1. This identificator is the main key that will be used to access the data. This policy implies that star original numbers found in publications have often to be transformed into the system adopted in the database. Complete cross-references have been determined, either by comparing cluster charts or by using rectangular (x,y) positions and carefully kept in tables, one per cluster. These tables give, for each star, the different identification numbers that are existing in the various numbering systems. The first column of the cross-reference table defines the adopted numbering system.

The second column of the table contains the numbers from the basic source and the numbers adopted in BDA are therefore identical to it. But no numbering system contains all the stars present in the cluster field and the basic numbering systems have been extended.

Two solutions have been adopted to record additional stars. The first one consists in continuing the numering system with the new stars of the second reference, entered in increasing number order, then with the third system and so on. This works fine if the additional stars are not too numerous. The second solution is used when one numbering system contains many new stars, but it is not desirable to change the reference system. This happens, for example, when proper motion studies contain lots of stars outside the main cluster area. In these cases, a constant is added to the star numbers. Adding 1000, makes it easy to read back the original numbers.

2.1.5 Star membership

The membership of stars to open clusters is certainly the most difficult problem to solve as a preliminary to any study. However there are very few methods to used when proper motions probabilities are not available. Existing codes developed at several observatories are not yet publicly available.

An expert system, adapted from Jonathan (Frot 1988), has been implemented in the database with a number of rules that should help the user to determine if a given star is a cluster member or not. Rules have to be improved to resolve the ambiguities resulting from the curvature of the sequences in some photometric diagrams and to extend the number of data taken into consideration. Presently, the expert system looks in the database for the UBV data, spectral types, proper motion membership probability and radial velocity, and distance from the cluster center and makes inference on the membership. It does the work of extracting the data itself and proposes a decision according to the data it has found and the rules it knows to analyse the situation. This behaviour reproduces the way the same question is solved by a human user. It could of course use more information than it does now, speak English instead of French, the language in which it has been developed, and above all, have better rules. It would however be extremely useful to be able to sort out the member stars more or less automatically, on the basis of objective criteria.

2.2 Directory organisation

The database is not a traditional relational database management system, but rather an advanced file management system. Clusters, but not stars, form the basic unit of the database and the structure has been designed to provide a natural working environment. The whole data set for a cluster forms a special relational set, because one key, the star designation, is common to all files.

The database structure uses the directory hierarchy supported by the Unix system. The main directory is the database itself. It contains several subdirectories: description of the database, help information, references, bibliography, programs, shell scripts. The clusters are collected in parent directories according to the source catalogues (NGC, IC or anon). Each cluster defines an independent directory identified by its name and containing the available data in distinct files, one for each data type. This structure allows easy inclusion of any new data type.

The present database structure is thought of as a first step in the organisation of cluster data and bibliography. When enough data analysis has been performed and only one set of data for each type will be available for each star, it may perhaps be more convenient to adopt another structure and collect all the data in one file for each cluster.

2.3 File organisation

The files are organised sequentially and compressed and, within the files, the entries are sorted by star number and source reference. Due to the small size of many files, there is no need for indexing or direct access.

The filenames reflect their content and are conserved throughout the data base. They are presented in Appendix D. The command names are usually identical to that of the data type they handle.

This file organisation offers several advantages: it allows the full use of UNIX file handling tools for searching and sorting and has enabled the straightforward implementation of graphics facilities. Compressing the files with the system function `compress` produces a reduction of disk storage by a factor of two. Many tasks can be performed without physically decompressing the files, by using the system function `zcat` to feed a pipe; it produces an output but leaves the files compressed. Finally, Unix editors are used to maintain the files, saving software development.

2.4 Record structure

Whenever possible, the records have the same structure: each record contains the star identification, the source and the data. White space are used as separators and no padding is used for missing data. The fields of each datatype are listed in Appendix E. The star identification is the main key to access the data, but it is also possible to use filters based on the data sources or astrophysical parameters.

Most data files are 1 to n relations because several sources of data may exist for each star. Files with mean or selected data are 1 to 1 relations. The first aim of the database and of the data analysis facilities developed is to pass from the 1 to n state to the 1 to 1 state. This means that at the end of the analysis process, only one value of each parameter will remain for each star.

2.5 Data sources

The database started with the installation of the data already collected and kept on magnetic tapes. Several catalogues had been announced and made available through the Strasbourg Data Center: UBV photometry and MK spectral types in open clusters (Mermilliod 1976a, 1986a), UBV photographic data (Mermilliod 1984a), individual radial velocities (Mermillliod 1979b, 1984b), cross-reference tables (Mermilliod 1979a), cross-identifications with astronomical catalogues (Mermilliod 1986b). Unpublished compilations made by me and containing CCD data, rotational velocities, membership probabilities and positions were used at the first installation of the database.

Aditional photoelectric photometric data were taken from the compilations made in our institute, and among them data in the following systems: uvby (Hauck & Mermilliod 1985), DDO (Mermilliod & Nitschelm 1989), Walraven (Nitschelm & Mermilliod 1990).

The files remained on magnetic tapes until the installation of Unix workstations and large disks in our institute made it possible to keep the data on-line. The data were organised in a database designed in March 1987 (Mermilliod 1988a, 1988b) and the compilations of cluster data were discontinued in their older form. New published data are entered regularly. A progress report on the introduction of recent data has been published in the CDS Bulletin (Mermilliod 1992a).

The bibliography from Alter et al. (1970) and the information from Lyngå's (1987) catalogue were taken from the files distributed by the Strasbourg Data Center.

2.6 Data content

The database tries to collect all published data for stars in open clusters that may be useful either to determine the star membership, or to study the stellar content and properties of the cluster. The data are usually recorded in their original form, with an indication of the source, but also as averaged values or selected data when relevant. The mean values for UBV (photoelectric, photographic or CCD) are not kept in the database, but can readily be computed. It presently contains:

- **astrometric** data: coordinates, rectangular positions, and some proper motions;
- **photometric** data in most systems in which cluster stars have been observed;
- **spectroscopic** data: spectral classification, radial- and rotational velocities;
- **bibliographic** information;

- **specific** information: (a) membership probabilities, (b) orbital elements of spectroscopic binaries, (c) remarks on peculiarity, variability, duplicity, (d) identifications of double star components, (e) cross-identifications with astronomical catalogues, (f) list of red giants in the cluster field and of non-member stars.

Table 2.1 gives an insight into the present content of the database; it lists for each kind of data the number of clusters involved, the number of measurements, and the number of stars concerned, except for the references and bibliography where the number of entries is indicated. The completeness of the data base should be rather high for many types of data, and is still improving.

Table 2.1: July 1994 database content

	Number of		
Subjects	Clusters	Meas.	Stars
Identifications	365		9876
Transit Table	190		68719
Coordinates	441	30621	27879
Positions	441		39424
Positions (x,y)	385		122819
Double stars	193	1662	1230
UBV photoelectric	413	30341	20632
UBV photographic	268	91458	71899
UBV CCD	69	30409	28322
UBV camera	3	194	194
UBV sit	3	284	284
UBV cmd	5		2867
UBV hrd	7		321
RGU (pg)	74	10191	10191
Geneva 7-colors	184		4306
uvby measures	155	5461	3897
uvby mean	155		3423
uvby Eggen	42	876	775
uvby CCD:	8	935	935
Hβ measures	216	5696	3821
Hβ mean	216		3312
Walraven	58	1432	1382
Vilnius	31	747	698
DDO	126	950	769
Washington	60	533	507
RI (Johnson)	2	405	350
RI (Kron)	4	1254	1111
RI (Eggen)	14	118	118
RI (Cousins)	23	869	847
RI (Cousins) CCD	8	3149	3112

Subjects	Number of Clusters	Meas.	Stars
JHK	33	776	774
uvgr (Thuan, Gunn)	4	144	144
Smith	5	103	103
MK types	272	7754	4645
MK types (selected)	162		3974
HD types	299		8584
Vsini	79	2222	1592
RV mean	63	1799	1612
RV individual	183	29712	3450
RV GPO	10	568	568
RV RFS	7		141
Orbits	35	199	175
Proper motion (abs)	1	2331	1164
Proper motion (rel)	4	3811	3811
Probability (μ)	59	21574	21574
Probability (RV)	2		87
Remarks	234	3703	3146
Periods	3	67	62
gK stars	231		3291
Am stars	36		109
NM	94		2645
Cluster maps	177		
Bibliography (Alter)	1200		
Bibliography (69-94)			3000
References			2500

A part from the published data found in the literature, the database contains also information that was accumulated in the past or prepared especially for it to enhance its capabilities:

- Cross-reference tables contain the basic cross-identifications between the various numbering systems in a cluster. A large part of the work has been done by the author.
- Rectangular (x, y) positions (usually in arbitrary units) have been collected from the literature or measured on published photographs with a digitizing tablet. A file collects the information relevant to the scale of the (x, y) positions and the sources of the data.
- Published photographs defining cluster numbering systems have been scanned and included in the database. About 180 maps have already been processed.

2.7 Access to the information

Thanks to the work on stellar identifications invested in the preparation of the catalogues used for building the database, the identification problem is greatly simplified; the various data for each star are collected under a unique designation, according to the numbering system adopted for the cluster. This identification is the main key for accessing the data of any star in any file. However, there are other possibilities for querying the database:

- One can recover the star number in the adopted numbering system (main key) starting with any identification existing in the cross-reference tables;
- One can access the data directly with HD, DM or other identifications;
- Data can be selected according to a source number;
- Samples can be formed according to astrophysical criteria.

Interesting clusters may be selected according to the number of available data of any type and the implementation of Lyngå's (1987) catalogue allows other types of selection, based on galactic position or cluster age for example.

2.8 Bibliography and references

The access to general or bibliographic information is considered as a very important facility that the database should offer. The catalogues already available in computer-readable form have been installed in the database and an additional bibliographic service covering the recent literature has been developed in another style to make the retrieval easier and more efficient. The various facets are provided by:

- the compilation of cluster parameters by Lyngå (1987), which provides the global information available on open clusters;
- the bibliography compiled by Alter et al. (1970) and its first supplement (Ruprecht et al. 1981), which collect the references from about 1900 or before up to 1973.
- the recent bibliography, covering the years 1969 to the present day, which has been developed for the database. This bibliography is based on chapter 153 of the Astronomy and Astrophysics Abstracts and the recent references are regularly entered in the computer. The bibliography search is based on keywords. The most obvious one is simply the cluster name, but many others can be used. Abstracts are not yet included in the database.
- the information on ongoing work, generally extracted from observatory reports and AAS abstracts is also available from the database.

The source of most data are indicated and the references can be obtained either one by one for any type of data or for any sample.

2.9 On-line help and information

On-line help and information are available at several levels. General descriptions of the database and applications are proposed through various menus. One can easily recover the names of the commands, the content of the files or the descriptions of the options. In addition, each command and function is (or will be) described in a manual file. A rapid help on the command syntax to use to obtain the desired results is available with an on-line menu which offers also many examples. Similar command description is also available in hypertext form and can be displayed with NCSA Mosaic.

2.10 Facilities offered by the database

The database offers several working facilities and some are described below. Three examples are discussed by Mermilliod (1992b).

- **Data comparaison:** Data comparaison is an important step before starting any study. Tools have been developed to compare data coming from different sources. This concerns essentially the UBV system because the UBV data represents a large fraction of the published photometric data. The other photometric systems seldom present two or more different sources of data.
- **Colour-magnitude diagram manipulation:** The main tool of the database allows to plot the various diagrams that can be built in the UBV and Geneva photometric systems. It offers a large spectrum of facilities, like fitting sequences or computing isochrones. Extensions have to be developed for other photometric systems (uvby, Walraven and others).
- **Technical data:** The technical information on instruments and data acquisition systems may also prove important to characterise the data contained in the database. Pioneering work has been done by van Leeuwen (1985) who collected the information on telescopes and plates related to proper motion studies in open clusters. This information is available in the database. The collection of similar information on telescopes and spectrographs used for radial velocity determination has been started. It would also be necessary to collect the same information for CCD photometry.
- **Miscellanous facilities:** To avoid remembering many commands and their options, a number of menus have been written which group most actions concerning a given subject. Those built up so far are generally related to the maintenance and development of the database, and were especially designed to facilitate the introduction of new data. In this category, one finds:

    1. the management of *coordinates:* it was made to determine coordinates of stars in open clusters starting from (x,y) positions in arbitrary units and include them in the database;

2. the management of *rectangular position:* (x,y) positions are now often published with CCD data. In addition many published charts have been measured and the results included in the database;
3. the management of *cross-references:* to maintain the principles of the database, it is necessary to determine the cross-references between any new numbering system and those already existing.

2.11 Graphics

The SM (2.3.1) graphics package written by Robert Lupton and Patricia Monger is extensively used to produce graphical output and hardcopy. It is often used in its interpreter mode for two reasons: firstly, macros take less disk space than compiled Fortran or C programs and are much easier to write, and secondly, the interaction with the user is extremely efficient; values of various parameters (plot size, title, symbols) can be changed interactively and the final plot printed. The dialogs are driven by menus.

2.12 Correction and updating

A program proposing pull down menus drives the updating processes through a number of shell scripts. Correction and updating is done either by editing the files when only a few data are to be added or corrected, or by shell handling file commands when more data are to be added.

2.13 User's profile

Even if the proposed software is not used, the utility of the database lies in the extensive collections of data brought into a uniform numbering system. The database can therefore be useful not only to astronomers working in the field of open star clusters, but also to students for a variety of work, because of the easy access to the data and facility for implementing users' programs.

Copies of the database have been sent to several colleagues in various countries. Other colleagues and several students working on their Ph.D. have also asked for specific data or bibliographic information by E-mail.

2.14 Scientific use

The realisation of a new atlas of colour-magnitude diagrams based on improved and homogeneous cluster parameters was the first scientific use of the database foreseen and the motivation of its development. The way to achieve this goal is however quite long, but it remains a major project. When reliable parameters have been determined for a number of clusters, it becomes possible to do some astrophysical research.

The information collected in the database may be used to study the clusters' stellar content and properties. In spite of the limitations due to the lower precision of some data, or their limited number, the database is the best starting point for many astrophysical studies involving open clusters. Nowhere are complete data collections to be found and one merit of the database is to give a clear report of the present observation status.

The database facilities have been used by Meynet et al. (1993) to compare new theoretical isochrones with the colour-magnitude diagrams of 30 clusters. In the spring of 1994, the database was successfully used to cross-identify the stars detected by the Tycho experiment on board the Hipparcos satellite.

2.15 Future plans

The challenge for the future lies in the management of the rapidly growing number of new data coming from CCD photometry and extensive observations of faint stars in nearby clusters and covering a wide range of groundbased and space techniques.

Due to the increasing quantity of available data, any study will become more and more time-consuming. One has therefore to think to more automated methods that rely not only on extensive data collections, but also on the transfer of knowledge to the computer, to let it do much of the work. Therefore, aside from continuing to collect and install new data and take other data types into consideration, the main development should consist of implementing more analysis facilities and calibrations.

Another important point would be to include in the database the knowledge contained not only in the data or procedures themselves, but in the article and review texts. Some could of course be recovered by searching the bibliography, but it may take a long time because the number of papers written every year on open clusters is increasing regularly and reaches the level of 200 papers for the best years. Therefore new results or exciting hypothesis should be extracted from the papers and stored in the database. This is actually possible thanks to the hypertext facility offered by NCSA Mosaic or the WAIS text indexing and search facilities.

Chapter 3

Database structure

The numerous files are located in a directory hierarchy based on a two or three level structure. The root directory is bda, the database itself and the data, programs, bibliography and documentation are contained in subdirectories. The following describes the database organisation and the content of the numerous directories

3.0.1 Clusters

Each cluster forms a directory identified by its name (like: 2287, 2516 for NGC clusters, 2391, 2602 for IC clusters or cr228, tr16 for anonymous clusters). The clusters are members of parent directories named ngc, ic, and anon depending on the catalogue source. Anon clusters should represent a two level structure author's name / cluster number, but this would have destroyed the general symetry and be a source of difficulty.
Therefore the path to the clusters NGC 2287, IC 2391 and anon tr16 are respectively:

- bda/ngc/2287
- bda/ic/2391
- bda/anon/cr228

Each cluster directory contains a working subdirectory, .T/, which is used to copy data or write temporary files.

3.0.2 Programs

Different programming languages are used to build the database software. The programs and shell scripts are grouped in directories according to their characteristics. The path to the executable programs and shells contained in these directories are set by the command `source .bdarc` which should be normally included in the alias bda.

- bda/bin groups shell commands, options and functions;
- bda/progf contains the Fortran sources and executables, corresponding mostly to application software;
- bda/progc contains the codes written in the C language, mostly system programs using Unix libraries;
- bda/progs groups the graphic programs that use the SM C library;
- bda/progx groups the elements related to the development of the graphical interface;
- bda/graphic contains the many routines of the program developed to plot photometric diagrams. This is a temporary implementation to facilitate the development;
- bda/smacro contains the SM default file (user's macros) and its two variants (one for each language) *default_fr* and *default_eng*;
- bda/src has a small numbers of sources of external routines;
- bda/lib will contain libraries of BDA routines.

3.0.3 Description and documentation

The documentation of the database, and on-line help of the various query modes are localized in several directories. The distribution is not really exclusive and will be improved when the query programs will be more definite.

- bda/introduction contains the user's guide and installation pages;
- bda/descri contains documentation files used by the `dsc` and `doc` menus;
- bda/html contains the hypertext version of the user's guide and the command description for use with NCSA Mosaic;
- bda/emploi contains the command description for line mode;
- bda/man contains man pages for BDA commands and menus which can be displayed with the Unix command `man`. This directory is subdivided in two subdirectories: `man1` and `man3`;
- bda/manuel contains the French manual files corresponding to the command `mnl`;
- bda/help contains help information for various programs;
- bda/xhelp contains the text of the on-line help provided with graphical user interface

- bda/dictionnaire contains files that describes the record structure of the data file;
- bda/information contains files with technical information on the data;
- bda/contenu contains the files collecting the database content summary for each datatype.

3.0.4 Bibliography and references

The files containing the bibliographic references to the literature published from 1969 to the present day are located in the directory bda/bibliographie. There is one file for the bibliography and one file for the keyword for each year.

The files containing the references of the data sources are located in the directory bda/references. There is one reference file for each datatype.

The files with the Alter et al. bibliography (1900 to 1973) and those from the catalogue of Lyngå (1987) are normally distributed in the cluster directories. The files for the clusters which do not have yet their own directory are collected in the following directories bda/catalogue/budapest and bda/catalogue/lynga respectively.

Finally, the original UBV photographic data from Moffat and Vogt (1972) are kept in bda/catalogue/bochum.

3.0.5 Miscellanous

There are more directories depending on bda.

- bda/hipad contains the (x,y) positions measured on a digitizing tablet and awaiting their installation in the database;
- bda/newdata new data generally received by E-mail and awaiting their inclusion in the database;
- bda/resultat results from data comparisons, and other files;
- bda/format templates for datafile format;
- bda/entete column headers for each datatype;
- bda/sequence reference sequences (ZAMS) for photometric diagrams, and evolutionary models for computing isochrones;
- bda/isochrone isochrones computed on the fly. This directory should be cleaned regularly;
- bda/modeles data on stellar inertia to compute binary system evolution
- bda/regles rules for the expert systems;
- bda/tmp temporary files used by commands and programs;

- bda/gestion — miscellanous files, related to the distribution of BDA and especially the file named *journal* which collects the names of the files modified or added and the date of the change;
- bda/sauve — backup copies of data or program files made before an important change;
- bda/oldata — cluster data, saved temporarily before doing a major change, like a numbering system change;
- bda/menus — French and English texts of the various menus;
- bda/poste — a place to put files to be sent by E-mail;
- bda/tarfile — the last tarfiles to update distributed copies of the database;
- bda/labo — a working directory.

Chapter 4

Query modes

4.1 Introduction

It is not so easy to design a query method that remains simple and avoids the use of an extended idiom and complex syntax. The idea was to avoid something like SQL because it is not be necessary to know the file- and field names to use the database. Therefore the query modes have been designed to hide as much as possible the database details.

In BDA four different approaches have been tried in an effort to minimize the learning and keep the typing at a low level. These four modes can be schematized as:

1. The command mode
2. The menu mode
3. The prompt mode
4. The graphical interface mode

4.2 The command mode

4.2.1 General principles

The historically first query mode developed is the command mode. Commands are issued at the Unix shell prompt and the user can benefit of the shell facilities, including command history, command editing and pipe or output redirection. Unix-style commands with several options have been written. These options provide most methods needed to extract information from the database, compare data and plot diagrams. For sake of simplicity, the commands that work on a specific datatype have the same name as the datatype designation. For example the command `ubv` deals with the UBV pe data and the `mk` command, with the MK spectral types. Command names are listed in Appendix A and option meanings, in Appendix B. The file names and content are described in

Appendix D, while the various fields of each data file are described in Appendix E.

The file organisation and their small size (a few hundred records), make the consultation very rapid with UNIX tools: the `grep` family and `awk`. The programming of the Bourne shell, together with specific Fortran or C language codes, offers interesting possibilities for coding elegant and efficient functions and commands.

Commands working on data are organised in the same way:

- Without parameter, the command provides a listing of the entire corresponding file, page by page;
- Given one or several star numbers as argument, the command returns the desired data;
- With a keyword, an option and one or several parameters, the command performs a selection according to the specification given, which may be a limiting value refering to the data or a reference;
- Options performing specific tasks have also been developed.

Commands relating to general information (manual, references, bibliography, help, statistics, and so on) can be called from anywhere in the database. However, most commands relating to data retrieval are active in a cluster directory only. Consequently, one first has to place oneself in a cluster directory, which is simply reached by giving its path: `ngc 2287`. Owing to the fact that the file names are conserved from cluster to cluster there is no need to tell a command which file it applies to: it knows it automatically. Because there is no a priori necessity to display a part of a record only instead of the whole line, it is not useful to know the detailed structure of the record for each datatype. A simple function called `nof` for "name of file" is used to get the corresponding filename. Try `nof ubv` just to see.

4.2.2 Command and function shell scripts

The command lines may have three possible structures:

1. `command star-number-list`: extract information for a number of stars. The datatype is implied by the command name, as for example: `ric 5 21 37`.
2. `command option values`: make a specific action driven by the option chosen. The values may be a list of stars or cutoff parameters, as for example: `ubv -a 1 2 3` or `mk -d 5`.
3. `command parameter comparator values`: perform a selection according to the chosen parameter and limits given by values, as for example: `ubv b-v -gt 1.5` or `mk ref -eq 999`.

Five comparators are used for making selections:

- "greater than" (-gt),
- "less than" (-lt),
- in an interval (-i),
- "equal to" (-eq),
- "nonequal to" (-ne).

The basic principle of the command software is the following:

1. the command parses the command line and calls the appropriate option,
2. the option function collects the necessary elements (column headers, data file, calls a awk function and displays the output,
3. the awk function does the real selection work.

The options have been grouped in a small number of scripts. Their names begin with opt_ and end with the options they contain, as for example `opt_pgine`. The names of older functions are formed simply by concatenation of "op" and the option name, as for example `op-d`. These options often call functions written in the awk language. With this structure and task distribution, any new option added to the system is immediately available for all commands without further modifications.

Figure 4.1 gives the listing of the `ubv` command which handles the UBV data through some 15 options. It has been chosen as an example because nearly all commands working on photometric data are simply links to the `ubv` command. This script first places the command name in the variable TY and writes it in a temporary file, and sets the variable FS to the name of the corresponding data file given by the function `nof` (name of file). Then it tests the number of arguments on the command line. If this number is equal to zero it displays the headers of the columns and lists the whole file. If there are one or more arguments on the command line, the script first tests if the first argument is numerical. If it is, it is assumed that the arguments are stars numbers and it calls the function `opt_noet` and passes to it the source filename, the datatype and all arguments.

If the first argument is not numerical, the script tests if it is a known option, and calls the respective function if the answer is positive. Finally, the command line may contain parameters for a selection and the option is then the second argument, which is tested. The position of the field used to do the selection is searched in the *pos.dic* file located in the directory bda/dictionnaire and is then passed to the function `opt_pgine`.

Figure 4.2 shows the code of this central function that performs the selections on star numbers, references, and parameters. It first puts in the variable

Figure 4.1: Example of a command: ubv

```
TY=`basename $0`
FS=`nof $TY`
typ $TY

case $# in

0) (cat $ENT/$TY.ent ; zex $FS) | page ;;

*) if numeric $1
   then
     opt_noet $FS $TY $*
   else

   case $1 in
     -a) op-a $FS $TY $2 ;;
     -y) op-y $FS $TY ;;
   -plt) diag $TY ;;
  -r|-u|-f|-sort) opt_star $1 $FS $TY $2 $3 ;;
  -d|-h|-t|-dr|-nb) opt_drnbt $1 $FS $TY ${2-1} ;;
   esac

   case $2 in
     -gt|-lt|-i|-eq|-ne)
          pos=`grep $TY  $DIC/pos.dic | \
              awk ' $1 == cle {print $2,$3} ' cle=$1`
          opt_pgine $2 $FS $TY $pos $3 $4 ;;
     -id) opt_star $2 $FS $TY $1 $3 ;;
   esac

   fi ;;
esac
echo ' '
```

Figure 4.2: Example of a function: opt_pgine

```
opt=$1
shift

case $opt in
-lt) (cat $ENT/$2.ent ; zex $1 | awkp $3 $4 $5 |\
        tee $SRT )| more ;;
-gt) (cat $ENT/$2.ent ; zex $1 | awkg $3 $4 $5 |\
        tee $SRT )| more ;;
 -i) (cat $ENT/$2.ent ; zex $1 | awkg $3 $4 $5 |\
        awkp $3 $4 $6 | tee $SRT )| more ;;
-eq) (cat $ENT/$2.ent ; zex $1 | awke $3 $4 $5 |\
        tee $SRT )| more ;;
-ne) (cat $ENT/$2.ent ; zex $1 | awkn $3 $4 $5 |\
        tee $SRT )| more ;;
esac

awk ' { print $1 } ' $SRT | sort -nu > $BTP/liste.noet
echo ' '
case $LNG in
  fr) echo " Etoiles selectionnees:
        `grep -c '^.' $BTP/liste.noet`" ;;
 eng) echo " Selected stars:
        `grep -c '^.' $BTP/liste.noet`" ;;
esac
```

opt the option, passed as the first argument and tests its value. In all five cases the structure of the code is similar: the whole action is put in a sub-shell so that the pagination made by `more` or `page` will leave the headers apparent on the top of the first page. The first part sends the appropriate column headers; the second feeds a pipe with a data file and the data are filtered by the corresponding awk function. The result is displayed on the standard output, but a copy is sent for further use to a temporary file *sortie.out* located in the directory `bda/tmp`, which is recorded as the environment variable **SRT**. When the action is finished, the file *sortie.out* is read to count the number of stars and display a message like: "Selected stars: 10". The command `zex` contains a simple `if` test. If the file is compressed, use `zcat`, else use `cat`.

The following figure 4.3 shows a typical example of the awk functions used to perform the real work. The preamble BEGIN is used to recover the arguments of the command line. The information on the data field passed to the function are the column number (pos) of the beginning of the field, its length in character (lon) and the cutoff value (lim). The substring is extracted and compared to the limit. If the test is positive, the line is printed. The other awk functions used in opt_pgine differs by the comparaison made in the function.

Figure 4.3: Example of a awk function: awkg

```
awk '
BEGIN {
pos='$1'
lon='$2'
lim='$3'+0
}
{
n=substr($0,pos,lon)+0
if( n >= lim )
  printf "%s\n",$0
} '
```

4.2.3 A complementary approach

A different solution was searched for in another approach. New commands with shorter and more direct scripts were written. Their names are evocative of the action they do, like `meas` or `list`. They effectively run a little faster and take usually less than one second, but the price to pay is an extension of typing. A example of the complete syntax would be `meas ubv 1`. The chain `meas` may be ascribed to a function key (F2 to F10) to save typing. Because these short scripts produce a shorter response time, a reverse policy has been adopted for the most common actions. It concerns the extraction of measurements, listing of a data file on the screen, display of the detailed content of a file and data comparaison.

A number of commands have no equivalent in command option syntax, like `carte, ximage, post`, and so on. See the Appendix A for a list and the second part of this guide for a description of their use and actions.

4.3 Menu mode

Because a lot of actions have been coded, it becomes difficult to remember all commands, options and requested parameters. Therefore menus have been designed that group actions related to the same subject, like `xy` or `sbs` do, or to the same working environment, like `ttr`. The text should allow to recognize easily the purpose of the items and the item order follows the logical order in which the work is usually done. An initial menu `go` is the starting point of BDA menu mode. It offers a possibility to get basic information and perform various tasks without knowing any of the command names.

The main menu usually presents a second one, more typical of the possibilities offered to solve the problem, and a third level menu may appear before the task is really done. In the menu cascade, the completion of an action with a menu brings back to the previous one.

Basic database query is possible with the main menu `go` and it is possible to obtain information without knowing anything about the database commands, which fills the schedule of conditions. It is however difficult to provide all facilities with menus, because it will lead to asking too many questions and need too much typing. The present menu `go` is not the best solution for querying data and making subtle selections. The command or prompt modes are better in this case. However, the menu mode offers an easy access to sub-menus that would be called anyway, to use calibrations, to study spectroscopic binaries or work on cross-references. This menu mode is the best way to access them and forget about command names and details of the processes.

Working in the menu mode is described in part II, chapter 8.

4.4 The prompt mode

In another attempt to diminish the typing for repetitive actions and make some functions easier, a prompt mode has been designed. The fact that most commands querying data offer the same options and have many similar functionalities, makes it a little absurd to have so many command names. This was also the reason to develop the three level structure using options and functions.

The reverse philosophy is faster but needs more typing. The prompt mode is an attempt to combine fast answer time and low typing. An attempt has been done to keep the number of files to a minimum. The program `dbms` is a large Bourne shell script. When run, it presents a prompt `bda>` to which commands are issued. The prompt is then modified and contains the command name. An agent has been called, in some similarity with Simbad vocabulary. To the agent prompt, one can enter star numbers, options and parameters, a datatype, depending on the context. One can enter `help` at any prompt to get a list of the valid answers. The agent `help` entered at the bda prompt opens a new window which allows to keep it in parallel. The help presents a list of all agents, datatypes and information on each.

At a certain level of complexity, it becomes again difficult to present a simple syntax and easy methods to execute tasks which requires complex selections, with several verbs and a variable number of parameters. Therefore, in this first version of `dbms` the command mode syntax is used for performing selection, because it is finally the more compact. However the `select` agent allows to use a syntax like `measure ubv> v < 7.5` to select stars with V magnitudes less than 7.5 from the UBV data file.

The complexity is also in the parsing program that analyse the entry line. The shell has been restricted to some standard style. Maybe that writing the program with Perl would bring more facility to do complex entry line parsing.

4.5 The user graphical interface mode

Finally the only way to avoid much typing is to use a graphical interface. A user graphical interface is being developed with the really fantastic tool called OpenWindows Developer's Guide 3.0, which allows to build windows, locate the widgets (choice panel, buttons, menus and so on), and create links with actions. The first version has been made for Open Look with SUN's Xview library. It offers a query mode based on the use of choice settings, command buttons and menus. It would however be useful to have a version of this interface done with the X Intrinsics Toolkit to port the interface to workstations using the Motif widgets.

The basic concept of the window presentation is to have a selection panel for choosing datatypes and command buttons or menus to perform the actions. The command part is a pile of five panels called: Interrogation, Selection, Application, Documentation and Development. The first command panel only has been completed and it allows to query measurements, list files, query the bibliography and references. Short on-line help explaining how to use the interface and each button has been prepared. It is activated in the standard manner, i.e. by setting the mouse pointer on the button and pressing the keyboard key <help>. Parts of the other panels have been realised to date.

Chapter 5

Installing BDA

5.1 From tape to disk

Copies of the database are available for SUN and DEC Unix worksations.

The tape has been created on SUN with the command `tar cfb /dev/rst1 126 .`. The information is thus blocked with a blocksize of 126.

To install the database, create a directory called bda (`mkdir bda`), move to it (`cd bda`) and tar the tape (`tar xbf /dev/unit 126 .`)

bda is the top directory which contains all the sub-directories. It also contains also a file named *.bdarc* (the dot is important!) which collects the environment variables and path. In particular, one has to write in the first line of this file the path to bda. The present one is /home/vaud/mermio/bda. /home/vaud/mermio/ has to be replaced by the new path.

5.2 The environment

The file *.bdarc* in the directory bda contains environment variables which have to be adapted according to your configuration. You have to edit this file and replace the parameters with those corresponding to your configuration.

5.2.1 Definition of environment variables

- setenv BDA /home/vaud/mermio/bda
- setenv SRC 7
- setenv OCL 8
- setenv CUT 25
- setenv RSRC 5
- setenv ROCL 6
- setenv LCUT 25

- setenv LNG eng
- setenv DEFTYPE ubv
- setenv EDT vi
- setenv PLOT sm
- setenv DEVICE "X11 -geometry 640x600+500+270 -bg white -fg black"
- setenv PRINT "postland sp2"
- setenv HELPPATH $BDA/xhelp
- setenv MANPATH $MANPATH:$BDA/man

5.2.2 Meaning of environment variables

- **BDA** is the access path to the root directory of the database bda/
- **SRC** is the number of elements of that path to ngc, ic or anon
- **OCL** is the number of elements to the cluster name
- **CUT** is the number of characters of the path until bda/.

  The data for NGC 2287 are located in the directory: bda/ngc/2287. Therefore, depending on the access path to bda, the variable **SRC**, **OCL** and **CUT** have to be adapted.
- **RSRC ROCL** and **LCUT** represent the same things, but are used by C programmes. They are presently different, this depends on the actual installation of each machine and the links between disks and directories.
- **LNG** is the language used. The two possible values are **eng** and **fr**. By setting **LNG** to **eng**, most dialogs will be in english, instead of french. Sorry for the remaining French.
- **DEFTYPE** sets the default datatype when entering a command that needs a datatype without argument.
- **EDT** contains the editor to use when the work or a menu item propose the edition of a file.
- **DEVICE** is the kind of device used for the graphics output on the screen
- **PRINT** contains the name of the printer device. Both variables are used by macros and C programs.
- **HELPPATH** is the path to the on-line help related to the graphical user interface
- **MANPATH** this line adds the path to BDA man pages to the usual MANPATH.

5.2.3 Checking the installation parameters

A shell program has been prepared to test that the values have been correctly assigned. To use it, move to the directory of NGC 2287 (from `bda: cd ngc/2287`) and enter `testenv`. Each option displays the current value of the parameter, the expected chain and that obtained.

Once you have set the path for BDA, most shell scripts and programs use this path as $BDA or are recovered by the function `getenv()`.

5.2.4 The file *.bdarc*

The file *.bdarc* contains also the path to directories with executable programs or commands:

- set path = ($path $BDA/bin)
- set path = ($path $BDA/progf)
- set path = ($path $BDA/progc)
- set path = ($path $BDA/progs)
- set path = ($path $BDA/progx)
- set path = ($path $BDA/graphic)

5.2.5 The file .alias.bda

The file *.alias.bda* contains some aliases that are used to move more easily within the database. Appendix G lists the most useful. Any other alias may be added in this file.

5.3 Other settings

5.3.1 The file *.cshrc*

The simplest way to go to bda is to add in the *.cshrc* file of each user an alias like:

```
alias bda 'cd /"path"/bda ; source .bdarc ; source .alias.bda'
```

`cd` will move you to the basic directory `bda`, `source .bdarc` will initialise the environment variables, `source .alias.bda`, useful aliases.

The *.cshrc* file should also contain the following instructions:

```
set lpath = ()
set lcd = ()
set cdpath = (..  ../..)
```

Figure 5.1: Menu items

```
"Database"           MENU
"Command mode"       DEFAULT       chdir $BDA ;
        exec $OPENWINHOME/bin/xview/shelltool tcsh
          -WI $BDA/bda.icon -WL "BDA Comm"
          -Wl "Open Cluster Database Command"
"Menu mode"
        exec $OPENWINHOME/bin/xview/shelltool go
          -WI $BDA/bda.icon -WL "BDA Menu"
          -Wl "Open Cluster Database Menu"
"Prompt mode"                      chdir $BDA/ngc/0103 ;
        exec $OPENWINHOME/bin/xview/shelltool dbms
          -WI $BDA/bda.icon -WL "BDA Prompt"
          -Wl "Open Cluster Database DBMS"
"GUI mode"               chdir $BDA/ngc/0103 ; exec xbda
"Mosaic"                 xmosaic $BDA/html/bda.html
"Database"            END
```

5.3.2 The .login file

The *.login* file should contain the terminal code according to the Termcap database if different from "sun".

To access BDA man pages, the path should be added:

```
setenv MANPATH ${MANPATH}:${PATH}/bda/man
```

5.3.3 The .logout file

The *.logout* file should contain the command `recompress` located in bda/bin which compresses at the end of session the files that have been uncompressed for various reasons.

5.3.4 The .openwin-menu file

It may appear convenient to start the various query modes with a choice in the main menu. If the group shown in Figure 5.1 is included in the *.openwin-menu* the menu item "Database" will appear in the main menu and a submenu will offer the five choices: Command mode, Menu mode, Prompt mode, GUI mode and NCSA Mosaic BDA front page.

Of course the action should be put on a single line, which is not possible here.

5.4 Compilation of the C or Fortran programs

The code sources are distributed in several directories:

- bda/progf — Fortran codes and executables;
- bda/progc — C codes;
- bda/progs — programs using the SM library;
- bda/progx — graphical interface C source;
- bda/graphic — source and exectuable of the program diag.

A file called *Makefile* has been prepared and is to be found in each one of these directories. Most programs are simply compiled like:

`f77 -o astrom astrom.f` or `cc -o ajzero ajzero.c`

The name of the executable is identical to the programme name, without the suffix .f or .c.

A few of them need either a routine or a library. See the *Makefile* files for more information.

5.5 Graphics package (SM 2.3)

The SM package has a *.sm* initialisation file. There is an example of such a *.sm* file (but it is not used) in the main directory bda. The line starting with **macro2** defines the path to the user's default file containing the user's macros.

Part of the database graphics is done with SM macros and these are located in the file *default* in the directory bda/smacro. To use the database graphics you have to indicate the right path in macro2, or to add the content of BDA *default* file to yours.

5.6 Tutorial of the database

The present introduction is located in the file *manuel.tex* located in the directory bda/introduction. The file *manuel.ps* is ready for printing on a postscript printer. A hypertext version is available under NCSA mosaic and can be reached from BDA front page.

5.7 References

The database has been described in two issues of the Information Bulletin of the Strasbourg Data Center.

Mermilliod J.-C. 1988, Bull. Inform. CDS 35, 77
Mermilliod J.-C. 1992, Bull. Inform. CDS 40, 115

5.8 Language of the database

EWhenthe language is set to English (**eng**) you shoud not find remaining dialogs in French. The English version has not yet been reviewed by English speaking persons, so you may find awkward phrasing or bad translation. Please send me your comments to improve it, and remarks on this user's guide, for unclear explanations, other things you would like to find in it, and so on.

The number of on-line help texts displayed by the command `emploi` translated in English is growing. If some yet non-existing file would be very useful to you, pleased drop me a mail, I'll do my best to prepare it.

5.9 Comments and questions

I am open to any comments, questions, suggestions, criticism, to improve the portability (I had a number of change to do to accomodate to DEC Unix), make the installation more easy and versatile to take into account various situations, and to add more application software to make the database more useful.

My E-mail address is: mermio@scsun.unige.ch

I wish you much pleasure in using BDA.

Bibliography

[1] Alter G., Ruprecht J., Vanysek V. 1970, Catalogue of Star Clusters and Associations. Akademiai Kiado, Budapest

[2] Frot P. 1988, 3 Systèmes Experts en Turbo C (Sybex, Paris)

[3] Hauck B., Mermilliod M. 1985, A&AS 60, 61

[4] Johnson H.L., Morgan W.W. 1953, ApJ 117, 313

[5] Lyngå G. 1987, Catalogue of open clusters parameters (5th ed.), CDS

[6] Mermilliod J.-C. 1972, Bull. Inform. CDS 3, 19

[7] Mermilliod J.-C. 1973, Bull. Inform. CDS 4, 22

[8] Mermilliod J.-C. 1976a, A&AS 24, 156

[9] Mermilliod J.-C. 1976b, A&AS 26, 419

[10] Mermilliod J.-C. 1979a, A&AS 36, 163

[11] Mermilliod J.-C. 1979b, Bull. Inform. CDS 16, 2

[12] Mermilliod J.-C. 1984a, Bull. Inform. CDS 27, 141

[13] Mermilliod J.-C. 1984b, Bull. Inform. CDS 26, 9

[14] Mermilliod J.-C. 1986a, Bull. Inform. CDS 31, 175

[15] Mermilliod J.-C. 1986b, A&AS 63, 293

[16] Mermilliod J.-C. 1988a, Bull. Inform. CDS 35, 77

[17] Mermilliod J.-C. 1988b, in Astronomy from Large Database I, Eds F. Murtagh & A. Heck, ESO Conf. and Work. Proc. no 28, p. 419

[18] Mermilliod J.-C. 1992a, Bull. Inform. CDS 40, 115

[19] Mermilliod J.-C. 1992b, in Astronomy from Large Database II, Eds A. Heck

[20] Mermilliod J.-C., Nitschelm C. 1989, A&AS 81, 401

[21] Meynet G., Mermillod J.-C., Maeder A. 1993, A&AS 98, 477

[22] Nitschelm C., Mermilliod J.-C. 1990, A&AS 82, 331

[23] Ruprecht J., Balasz B., White R.E. 1981, Catalogue of Star Clusters and Associations, Supplement I. Ed. B. Balasz, Akademiai Kiado, Budapest

[24] van Leeuwen F. 1985, in IAU Symp. no 113, Eds J. Goodman & P. Hut (Reidel, Dordrecht), p. 579