USER GUIDE
for a
DATABASE of CATEGORIES
(abridged)

by
M. Fleming, R. Gunther, R. Rosebrugh
Mathematics and Computer Science
Mount Allison University

1 Introduction

Use of this program requires a basic knowledge of category theory. It allows storage and manipulation of finitely-presented categories. The program was written in ANSI C and is menu-based. Its format for storing categories and functors, and some of the data structures are based on those developed by S. Carmody and R. F. C. Walters of Sydney University [1]. The basic program allows creation, editing and storage of finitely-presented categories and functors. In addition, there are several tools for testing properties of objects and arrows, and the computation of right and left Kan extensions is implemented.

This document is an abridged version. It explains some of the options available to the user. The full User Guide, with more detail, is available as
`ftp://sun1.mta.ca/pub/papers/rosebrugh/catuser.{tex,dvi}`
. The program is stored in executable form (Sun Sparc1+) as
`ftp://sun1.mta.ca/pub/sources/rosebrugh/unix/category.exe`
A DOS executable is stored as
`ftp://sun1.mta.ca/pub/sources/rosebrugh/DOS/category.exe`

2 The Main Menu

When the program starts the Main Menu is displayed:

```
                    Categories Database

              (1)  Category Menu
              (2)  Functor Menu
              (3)  Category Tools
              (4)  Right Kan Extension
              (5)  Left Kan Extension
              (6)  Change maximum order of endomorphisms

              (0)  Quit

Your choice...
```

Most of the menu options will call up another menu, while others will prompt the user to enter some form of input. The first five choices will be discussed later in this document. We first look at options (0) and (6).

`(0)Quit`
Before terminating, the program will ask the user if they wish to save each category or functor currently in memory.

`(6)Change maximum number of endomorphisms`
Although most of the manipulation tools below require a finite category, it is possible to store any finitely presented category. This option allows the user to control the maximum number of times an endomorphism will be traversed by some of the tools in the program. The default value is 2.

3 The Category Menu:

Selecting option one, `Category Menu`, from the Main Menu will display:

```
CATEGORY MENU

(1)  Create category
(2)  Load category
(3)  Edit category
(4)  Display category
(5)  List current categories
(6)  Save category
(7)  Remove category

(0)  Back to main menu
```

Option three, `Edit category`, calls a new menu and will be discussed later.

`(1)Create Category:`
Once this option is chosen the following prompt will be displayed:
`Category name:`
The user should type the name of the new category. The screen will be cleared and at the bottom of the screen you will see:
`Enter @ to display all objects`
`Enter object name (type ´enter´ when finished) :`
An object name must be a single character other than '@'. If you wish to view the objects already entered, type in '@'. When all objects in the category are named, type <*CR*>at the prompt to proceed.

The following will be displayed on the screen:
`Enter @ to display all arrows`
`Enter arrow name (type ´enter´ when finished) :`
Entering arrows is very similar to entering objects. After the name of an arrow has been entered, the program will ask the user to input the domain of the arrow and then the codomain. The domain and codomain must be objects of the category and an arrow cannot be named `1`, the symbol reserved by the program for identity arrows.

Next the user enters the equations of the category. The following will be displayed on the screen:
`Enter @ for left side to display all arrows and relations`

`Enter left side of equation:`
To enter an equation, type in the left side of the equation. Then type in the right side of the equation. If you wish to have some path of arrows in the category equal the identity, type in this path for the left side and enter `1` for the right side of the equation. To display the current arrows and equations type in '@' for the left side and <CR>for the right side. Once the equations have been entered, typing <CR>for both the left and right sides of the equation will bring back the Category Menu. An equation is invalid if it contains arrows that are not in the category, or an illegal composition, or if the the domain and codomain of the left side do not equal the domain and codomain of the right side.

`(2)Load Category`
Choosing this option will clear the screen and display the following prompt at the bottom of the screen:
`(-1 to cancel)`
`Enter name of category you wish to load>`
Enter the name of the file containing the category you wish to load and the program will load the category, or display an error message.

`(4)Display Category`
This option will prompt the user to select which one is to be shown on the screen.

`(5)List current categories`
Choosing this menu option will display a list of the categories that are currently in memory.

`(6)Save Category`
This selection prompts the user to select a category to be saved, and then asks for a file name in which to store the category.

`(7)Remove Category`
This selection will prompt the user to select the category to be removed from memory.

`(0)Back to Main Menu`
This option will return program control to the Main Menu.

4 The Category Edit Menu:

Choosing option three, `Edit category`, from the Category Menu will clear the screen and display a list of the categories in memory. Enter the number of the category you wish to edit. A new menu, the Category Edit Menu, will be displayed with the following options:

`(1)Change Name`
This selection allows the user to give the category a new name.

`(2) Add Objects, (3) Add Arrows, (4) Add Relations`
These options allow the user to enter new objects, arrows and relations.

`(5) Remove Objects, (6) Remove Arrows, (7) Remove Relations`
These selections allow the user to delete objects, arrows or relations from the current category.

`(8) Display Category`
This selection will display a list of categories currently in memory, followed by the prompt:
`Category to display>`
Enter the number of the category you wish to see, and $<CR>$. The category will be shown on the screen. Press $<CR>$to return to the Category Edit Menu.

5 The Functor Menu:

Choosing option (2) from the Main Menu will clear the screen and display the Functor Menu:

```
FUNCTOR MENU

(1)  Create functor
(2)  Create functor to SET
(3)  Load functor
(4)  Load functor to SET
(5)  Remove functor
(6)  Save functor
```

```
(7)  Display functor
(8)  List current functors

(0)  Exit to main menu

Your choice>
```

Type the number of the option you would like, and <CR>.

`(1) Create Functor`
Once this option has been selected, a list of categories will be displayed, followed by the prompt:
`(-1 to cancel)`
`Functor from>`
Select the appropriate number from the list for domain category. This will bring up the prompt
`Functor to>`
where you should enter number of the codomain category. You will be prompted to enter the functor name. You will see a display of the two categories, then a prompt like:
`(Object) X-->_`
In this example, `X` is an object in the first category, and you are being asked to enter the object in the second category. Once you have defined the functor on objects, you will be prompted to do the same for arrows. This input can be a single arrow or a composable string of arrows in the second category. Once you have successfully defined the functor for arrows, the program will check that equations in the domain category hold for values of the functor in the second category. If all equations hold, the creation of the functor is complete.

`(2) Create functor to SET`
This selection is quite similar to the `Create functor` option. Here, you will be able to create a functor to the category of finite sets. Since this functor will always go to SET, you only need to respond to the
`Functor from>`
prompt. You will see the source category displayed on the screen, followed by a prompt such as:

`(Object) A-->`
Enter the number of elements in the set which is the image of `A` under the functor. Next define the SET-valued functor on arrows. This prompt will appear as, for example:
`{-1 to quit} (Arrow) f : [2] --> [3].`
In this example, the domain of arrow `f` is an object which has as its image the set [2]= $\{1, 2\}$, and the codomain of `f` has as its image the set [3]. If your set functor is named `X`, you will be asked to define `X(f)`. For each element in the domain, you will be prompted for its image.

`(3) Load functor`
This option allows you to load a functor saved on disk.

`(4) Load functor to SET`
This is basically identical to choice (3).

`(5) Remove functor`
This selection will display a numbered list of functors, followed by a prompt.

`(6) Save functor`
This choice will display a list of functors in memory, and then a prompt. You will be asked to type in the filename under which you would like to save the functor.

`(7) Display Functor`
Again, you will see a numbered list of functors. Enter a number.

6 The Category Tools Menu

Choosing option 3 from the Main Menu, `Category Tools`, will clear the screen and display a list of all the categories currently in memory. Type in the number of a category. The Category Tools Menu is displayed:

```
CATEGORY TOOLS

(1) Make Confluent
(2) Initial Object?
(3) Equality of Composites
```

```
                    (4) Make Dual
                    (5) Sum?
                    (6) Display Category

                    (0) Exit to Main Menu

  Your choice...
```

The options are:

`(1) Make Confluent`
Choosing this option will cause the program to make the set of equations in the current category confluent by adding new equations if necessary.

`(2) Initial Object?`
NOTE: Before choosing this option, be sure that the current category is confluent; choose option (1) if unsure.
After displaying the current category, this option will allow you to choose either to test all objects in the category, or to test one specific object. The program will then show you whether or not an object is an initial object.

`(3) Equality of Composites`
This feature will determine if two composable paths are equal. *NOTE: Before choosing this option, be sure that the current category is confluent.* You will be prompted to enter two paths.

`(4) Make Dual`
This selection will create the dual (opposite) of the current category and store it on disk.

`(5) Sum?`
Choosing this item will determine if an object and two paths, the candidate injections, into the object are a sum diagram in the category. *NOTE: Before choosing this option, be sure that the current category is confluent.*

`(6) Display Category`
This selection will display the active category on the screen.

7 Kan Extensions of **set**-valued Functors

NOTE: If you intend to use a particular functor in a Right or Left Kan Extension, and you have edited one of the categories involved in the functor, the functor itself *must* be re-created.

7.1 Right Kan Extension

Option (4) from the Main Menu, `Right Kan Extension`, will allow the user to compute a Right Kan Extension.

The user will first see a list of categories, and will be asked to select category **A** and then category **B** for the Right Kan Extension. Then, a list of functors will be displayed and the user will be asked to select a functor from category **A** to category **B**, followed by a functor from **A** to the category of sets. The user will then be asked to enter a file name in which to store the output.

Before proceeding with the Right Kan calculation, the program will automatically make sure that both category **A** and category **B** contain a confluent set of relations.

The output for the Right Kan Extension is somewhat complex. Consider an object B in category **B**. For this object B, the category B/F is calculated. We construct a full sub-category $J(B/F)$ of B/F whose inclusion in B/F is initial.

The objects of $J(B/F)$ are displayed for an object B in the Right Kan output. $R(B)$ is computed as a subset of the product:

$$\prod_{\substack{\beta: B \to FA \\ \beta \text{ in } J(B/F)}} X(A)$$

After the objects of $J(B/F)$ are displayed, the tuples in R(B) are listed. If R(B) is empty, the number 0 will appear. If there were no objects in $J(B/F)$, R(B) will be displayed as 1.

To the right of each tuple will be a column for each object A in category **A** with $B = FA$. This column provides information about the A component $\rho_A : RF(A) \to X(A)$ of the natural transformation ρ.

After all of the tuples have been displayed with the ρ information, they will be redisplayed with information about the action of the Right Kan Extension on arrows of $\mathbf{B}$. The user will see the objects of $J(B/F)$, and then the objects of $J(B'/F)$ for each object B' which is the codomain of an arrow out of B. To the right of each tuple, for each arrow $f : B \to B'$, you will see where $\mathrm{R}(f)$ takes that tuple in $\mathrm{R}(B')$. This entire procedure will be repeated for all objects in $\mathbf{B}$.

7.2 Left Kan Extension

The algorithm is an implementation of the Todd-Coxeter algorithm as described in Walters' book. The user input for a Left Kan Extension is identical to that for a Right Kan Extension.

The output of the Left Kan Extension begins with information about the natural transformation $\epsilon : X \to LF$. For each object A in category $\mathbf{A}$, $\epsilon_A : XA \to LFA$ is shown.

After this, you will see the action of the Left Kan Extension L on the objects and arrows of $\mathbf{B}$. Each object B in $\mathbf{B}$ will be displayed with all of the elements of $L(B)$ below it. To the right will appear all generating arrows out of B with their action under L printed below.

Reference

[1] R. F. C. Walters, *Categories and Computer Science*, Cambridge University Press, 1991.