Crash Course in Minlog and Emacs

Kenji Miyamoto
miyamoto@mathematik.uni-muenchen.de

June 10, 2016

1 Installation

For detailed steps of installation, see http://minlog-system.de/.

2 Starting up Minlog

Assume we have installed Minlog. We refer to the Minlog home directory by `MINLOG_HOME`.

Linux/Mac There is a script whose name is `minlog` in `MINLOG_HOME`. Go to the command line terminal, and run the command `minlog`.

Note 2.1. *It is convenient to configure the environment variable `PATH`, so that you can just type `minlog` whatever your current directory is. If you use `bash` for command line shell, you edit `~/.bash_profile` to add your `MINLOG_HOME` to the `PATH`. Without the help of `PATH`, one can still run `minlog` by eg. issuing the command with relative path as `MINLOG_HOME/minlog`.*

Windows Double click the short cut icon.

3 Emacs

It is recommended that you open two emacs windows (in literature it says emacs *frames* instead), as it is what the start up script/short cut prepares. In one window you write a Minlog proof script, eg. your exercise to submit, and in the other window you see responses/messages from the Minlog system. After launching Minlog using the start up script, you should have an Emacs frame which displays the message "Minlog loaded successfully". This window is the one actually dedicated to see what Minlog says. We call this window *Minlog window* or *Minlog frame*. The other window to edit proof scripts would be called *Work window* or something like this.

We are going to use the following Emacs key notation in Table 1. For example, C-x means to press and to hold the control key and hit x.

C-	press and hold the control key
M-	press and hold the meta key, which is usually the Alt key
C-M- and M-C-	press and hold the control key and the meta key at the same time

Table 1: Emacs Key Notation

3.1 Work with Emacs Frames

Open a new frame C-x 5 2, meaning, press and hold the control key and hit x, then release everything, hit 5 and hit 2.

Close the current frame C-x 5 0. The current frame means the frame which is currently foreground. You can also use common GUI of closing the window.

When you got confused about what command you are typing, do the following to cancel it.

C-g Cancel running or partially typed command.

3.2 Open and save files

When Emacs frames are set up, you are going to open a file to edit.

Note 3.1. *In Emacs, when you edit something from scratch, you first specify some (not yet existing) file name. This is done in the same way as opening an existing file. There is nothing exactly same as "Untitled" used in other text editors as `notepad.exe`.*

Open file C-x C-f, then type the file name in the mini buffer, which is a small text area at the bottom of an Emacs frame. Alternatively, you can use the menu bar, File → Open file... It can also offer a dialog box with a file browser.

Save file C-x C-s

Save file as (another file) C-x C-w, then you input another file name in the mini buffer. Alternatively, you can use the menu bar, File → Save file as... It can also offer a dialog box with a file browser.

Note 3.2. *The tab key can be used for auto-completion in many cases. You can try it in the mini buffer, when you open an existing file.*

3.3 Edit texts

C-x u Undo. In some environment, C-/ also works as undo. The menu bar also offers it (Edit → Undo).

C-d Delete the character on the cursor.

C-k Delete all characters between the cursor and the end of the line.

C-⟨space⟩ Set the mark, which is used as an end of a region for copying or for cutting.

M-w Copy the region between the mark and the current cursor.

C-w Cut the region between the mark and the current cursor.

C-y Paste the copied/cut text.

Note 3.3. *It is also an option to drag some text region and use menu bar to copy, to cut and also to paste. Pasting does not respect the current region, but just the current position of the cursor.*

Emacs does not throw away previously copied/cut text when there is something new to copy/cut, thanks to a list of blocks of texts which is so-called kill ring (in Emacs, kill means cut). Assume you cut a text (say, text1) and after that you again cut another text (text2). In order to paste the text1, you paste by C-y (you see text2 pasted), then do M-y immediately.

3.4 Move cursor

C-a Move the cursor to the head of the current line.

C-e Move the cursor to the end of the current line.

The following shortcuts are needed to work with Scheme code, where you find many parentheses.

M-C-f Using this on the open parenthesis, the cursor moves to the corresponding closing parenthesis.

M-C-b Opposite of the above.

Note 3.4. *The above explanation for M-C-f and M-C-b are rather too simplified, but practically it is enough for this time.*

C-v Page down.

M-v Page up.

C-l Once you type C-l, the line of the cursor comes to the center of the frame. Doing C-l again, the line of the cursor comes to the top of the frame. Doing C-l once again, the line of the cursor comes to the bottom of the frame. You can further repeat the above three.

3.5 Search and replace

C-s Forward search. You type any text to find. You can use C-s repeatedly to find the next candidate.

C-r Backward search.

M-% Query replace. In the mini buffer, you input a string to be replaced, then you input another string to replace. Starting from the current cursor, you interactively replace strings.

M-x replace-string ⟨enter⟩ Replace (not interactively). Basically same as the above M-%, but it replaces every candidates without any interaction with you.

3.6 Emacs Tags

M-. Assume you have issued make command in `MINLOG_HOME/src`, so that an Emacs TAGS file is existing there. This command is to find for the given tag the definition of it in the source code of Minlog. Inputting this command, you will be asked what tag you want to find. Tags are roughly speaking identifiers in Scheme programs for this case, namely, tags contain Minlog commands as `set-goal`, `add-alg`, `elim`, `intro` and etc. After entering the tag you want to find, you would be asked the location of the TAGS file. You should then tell Minlog the path `MINLOG_HOME/src/TAGS`. Your cursor in Emacs jumps to the corresponding definition of the input tag, and you can see what the definition of the tag is.

4 Minlog

Emacs offers some means for us to communicate with Minlog. What we can do is to send Minlog commands, which are in fact expressions in Scheme language, by means of the following commands.

C-x C-e Assume the cursor is at the end of a Minlog command. It sends the command to Minlog.

C-c C-r Send multiple commands in the selected region. Use C-⟨space⟩ to make a mark, then you move to somewhere to select a region, which is in fact between the marked position and the current cursor. Using C-c C-r, everything in the region is sent to Minlog.

By means of Minlog's `(undo)` command, some steps of interactive proof can be undone. You can optionally give a number (eg. `(undo 5)` for 5) to undo the last 5 steps.

If you would like to send a command to Minlog, which is not needed to be saved in your proof script, you can just go to the Minlog frame and enter the command you like. It happens to do this when you undo some proof commands which have given to Minlog.