Editing Scheme Files

Electronic Handout for CS 2135 C95
by RSR

When using MIT Scheme, it is usually desirable to edit procedure definitions, not to type them directly at the Read-Eval-Print loop prompt. This way, if (or should I say when) an error is made it can be corrected without retyping the entire definition. Saving it in a file is also useful in case you need it again at another session.

There are a few ways this can be done. The first and simplest is to use your favorite editor to simply edit the procedure and save it as a text file. Let’s say the file is called myfile.scm. (Note that Scheme source files should end in .scm, and a single file may contain many definitions.) You may then load the file into an active Scheme session by typing (load "myfile.scm") at the REPL prompt. If you are running on a workstation in an X Windows environment, it is convenient to keep one window for the editor (saving changes without exiting) and another for the Scheme session.

If the editor you use is emacs and the filename ends in .scm, the editor will give you some assistance in formatting and debugging your program as you type. When you type a return followed by a tab, the cursor will indent the appropriate amount for the current nesting level. This not only pretty-prints your code but may help you detect improper nesting levels. Also, when you type a close parenthesis, emacs will briefly move the cursor to the matching open parenthesis.

In addition, if you use emacs you can run Scheme as a subprocess. To do this, use M-x run-scheme from within emacs to start Scheme. (If you do not understand this syntax, you probably do not know enough about emacs to use it yet.)

Another way to edit procedures is with the Edwin editor from within Scheme. Edwin is an emacs-like editor with similar syntax and functionality for use with Scheme. To use this, run Scheme with the optional -edwin argument. At the REPL prompt type (edwin) to start things up. You should be familiar with emacs commands to use this, though there is some help from within the program.

Finally, if you have defined a procedure (let’s call it proc) in Scheme, either by typing it at the REPL prompt or by loading it in from a file, you may print out its definition by typing (pp proc) at the REPL prompt. Note that this is not standard for all versions of Scheme.