Documentation for Cedille

Aaron Stump

January 27, 2016

1 Basic setup

Cedille is used via the `cedille-mode` emacs mode.[1] This mode is based on Carl Olson's `structured-editing` (`se`) mode, which provides a generic, language-independent (minor) mode for navigating through structured text. If you open a `.ced` file, Cedille mode will be invoked, and a request will be made to `cedille.sh` to parse the file. This `cedille.sh` wrapper script will invoke the actual `cedille` executable on the file. When the `cedille` executable runs on a file like `nat.ced`, it will produce a file called `nat.cede` containing all the typing information for `nat.ced`. It will also do this for any files that are imported by `nat.ced`. (Note that currently, any imported files must be in the same directory.) The `cedille.sh` script will send the contents of the `.cede` file (e.g., `nat.cede`) back to the emacs mode. Then Carl's code will parse this information to construct a parse tree, which can then be navigated within the emacs mode, as described more below.

2 Entering and exiting navigation mode

To enter navigation mode for a buffer holding a `.ced` file, you type "Meta-s", where Meta is likely bound to the Alt key on your keyboard. In a pinch, the Escape key always works for Meta. Once in navigation mode, you will see the *mode line* at the bottom of the buffer for the `.ced` file display "Cedille navi" instead of just "Cedille". To exit the mode, you can type "q", or "Control-g" will also work. You cannot modify the text while in navi mode. You can only traverse through it and view type information.

To get some rudimentary help on the navi mode commands, you can type "h" while in navi mode.

3 Structured navigation

The basic idea of `se-mode` is to allow the user to navigate through text following the structure of a parse tree for that text. The parse tree information is generated by the external tool, in this case the `cedille` executable, in the form of *spans*. A span consists of a name, a starting character position in the file, an ending character position, and additional information. The nested structure of spans is what `se-mode` uses to reconstruct the parse tree for the text. The basic navigation commands are the following:

- "p" selects the innermost span which starts at point ("point" is the character position at which the cursor is currently located in the buffer). If a span is already selected, then select its parent span (i.e., go up in the parse tree).
- "n" selects the span corresponding to the first child of the parse tree.

[1] How to install this mode is not covered in this document.

- "f" select the span corresponding to the next sibling (to the right) of the node in the parse tree whose span is currently displayed.
- "b" select the span corresponding to the previous sibling (to the left) of the node in the parse tree whose span is currently displayed.
- "a" select the first span out of all the siblings of the currently displayed span.
- "e" select the last span out of all the currently displayed span's siblings.

4 The info buffer

For each file `X.ced` being viewed in navi mode, the Cedille emacs mode will create an info buffer, with the name `*cedille-info-X*`. To display this buffer, you type "i" in navi mode. To hide it, you also type "i". This buffer shows information associated (by the `cedille` executable) with the currently selected span of the buffer.

5 Errors

Parse errors in Cedille can be hard to debug, and due to the home-grown parser technology (`gratr`) that I am using, a file with parse errors can be very slow to process by the `cedille` executable. If there are parse errors, they will be displayed in a small buffer when navi mode is activated. They must be fixed before navigation features will work.

Assuming there are no parse errors, you will frequently have type errors to deal with in your Cedille code. To see the next type error, type "r" in navi mode. To see the previous one, type "R". If there are no errors, you will see "No errors" displayed when you type "r" or "R".