CSP-Mode

The Emacs major mode for editing CSP code in FDR2-syntax, Edition 2
Nov 2001

by Olaf Bergmann, Markus Dahlweid

A mode for programming CSP in Emacs.

Designed and written by

Olaf Bergmann	bergmann@informatik.uni-bremen.de
Markus Dahlweid	dahlweid@informatik.uni-bremen.de
Uwe Schulze	uschulze@informatik.uni-bremen.de

1 Introduction

Csp-Mode is a major mode for use with Emacs.

1.1 History

In winter 1996/1997 there were some students of computer science at the University of Bremen, Germany, who felt that there was need of a mode for editing CSP-code using the ASCII representation for the FDR2-model-checker.

2 Commands and keybindings

As usual for common major modes dealing with program languages, there are some often used keys which take care for correct indentation of expressions and comments. The indentation itself may be controlled by a number of parameters we look at in the following chapter.

- Pressing `TAB (electric-csp-tab)` as usual indents the current line looking at its preceding line. Each tab is converted to spaces if and only if `csp-untabify` is set to `t`.
- Pressing `LFD (electric-csp-terminate-line)`indents the current line correctly and terminates it. The next line is indented as well.
- `C-c C-c (csp-comment-area)` will comment out the currently marked area. This is done by inserting the b-style comment-start-sequence `--` at the beginning of each in line in the current region. If there is already such a character combination, nothing is inserted. This may cause problems when you press `C-c C-c` being in a block-comment since executing the inverse operation `csp-uncomment-area` will delete all leading comment-start-sequence characters.
- `C-c C-u (csp-uncomment-area)` removes all leading comment-start-sequence characters in the commented area surrounding the current position. The first commented line in this area will be searched. The actual "Area" is then delimited by the first and the last line which have no leading comment-start-sequence (i.e. they even might be empty).
- Pressing `; (electric-csp-semi)` and `csp-auto-newline` is non-nil will cause a semicolon to be inserted at the current position. After that, the current line is terminated and the following line will be indented correctly.
- `C-> (electric-csp-arrow)` does the same as `electric-csp-semi` described before, but the arrow-operator → is inserted at the end of the current line.
- `C-c e (electric-csp-external-choice)` inserts an external choice operator `[]` at current position.
- `C-c l (electric-csp-interleave)` inserts an interleave operator `|||` at current position.
- `C-c ~` or `C-c i (electric-csp-internal-choice)` inserts an internal choice operator `|~|` at current position.
- `C-c |` or `C-c p (electric-csp-sync)` inserts a synchronisation operator `[| |]` at current position and places the point centered between the two vertical bars.
- `C-c {` or `C-c s (electric-csp-channel-set)` inserts a set operator `{| |}` at current position and places the point centered between the two vertical bars.
- `C-c C-v (csp-validate)` invokes the validation command specified in `csp-validate-command`. If errors occurred, you can use `next-error` to find the actual code line corresponding to the line given in the message.

3 Customizing

This chapter describes the variables you can use to customize the behaviour of csp mode. At least there are switches to determine whether lines are allowed to be indented or tabs to be converted into spaces automatically.

- `csp-indent-level` is an integer-value denoting the amount of spaces which are inserted when a line is indented with respect to a containig block. Its default value is `3`.
- If `csp-auto-newline` is non-nil, this means that there will be inserted a newline automatically after inserting a semicolon or the array-operator. The default is set to `t`.
- `csp-tab-always-indent` means that a `TAB` in Csp-mode should always reindent the current line, regardless of where in the line point is when the `TAB` command is used. Default is `t`.
- `csp-untabify` specifies whether tabs should be converted into spaces or not. If `t` which is the default value, tabs will be converted.
- `csp-validate-error-regexps` contains an alist of regular expressions to recognize error messages from `csp-validate`.
- `csp-offer-save`. If this is `t`, invocation of `csp-validate` will cause Emacs to ask whether modified buffers should be saved before executing `csp-validate`. Default is `t`.
- `csp-validate-command` contains the shell-command to validate the CSP program in current buffer. The file name of the buffer will be appended, seperated by a space.

4 Bugs

There are several bugs known by now, but none of them are severe. As for the most syntax-controlled programming modes, correct statements will be indented correctly, but the result of indenting wrong statements will sometimes have funny effects. Fortunately, most of them indicate incorrect use of the language's constructs, so this will help you not to make too many syntactic errors. However, there still remain some dirty effects even if typed syntax is correct. Here is a list of the known bugs:

- Block comments are not recognized correctly if there is a normal comment starting with `--` in it.