CoCoA-5 - Bug #280

EmacsUI: indentation is wrong with parentheses

29 Nov 2012 11:30 - Anna Maria Bigatti

Status:	Closed	**Start date:**	29 Nov 2012
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	EmacsUI	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.0.3	**Spent time:**	4.65 hours

Description

cocoa5-mode indents wrongly withing parentheses (and one student got worried ;-)

```
M := F(K, (1,
        2));
```

This is because pascal-mode (the original source for cocoa(5)-mode), indents that way.
.... where to copy the "correct" code?

History

#1 - 29 Nov 2012 11:52 - John Abbott

I have had a quick look at the Emacs LISP source code. It is not terribly readable. I believe the relevant function is **cocoa5-calculate-indent** (in the file src/CoCoA-5/emacs/cocoa5.el; well, the name is promising. I could probably fix it in half a day, but it's always hard to estimate how long it'll take to correct undocumented code written by an unknown person...

#2 - 29 Nov 2012 15:04 - John Abbott

I believe the Emacs code for C/C++ is in cc-mode.el. On my computer the full path is:
/usr/share/emacs/22.1/lisp/progmodes/cc-mode.el.gz

#3 - 29 Nov 2012 18:26 - John Abbott

I believe the problem is in lines 697-698 when the value for par is computed.
It is not clear to me how the value of par is used later. As far as I can tell the call to **scan-lists** produces the "wrong" answer; but playing with the args to scan-lists just produced errors.

Manually setting the value of par to the correct value produces the desired behaviour in emacs.

I would have thought that calling **(backward-up-list)** instead of (goto-char (scan-lists ...)) would achieve the desired result, but it does not. Emacs reports Expression ends prematurely... Puzzled!

#4 - 30 Nov 2012 11:31 - Anna Maria Bigatti

- Status changed from New to Feedback

- Assignee set to John Abbott

- % Done changed from 0 to 100

changed relevant code in cocoa5.el into

```
                ((> (car state) 0)
;;                  (goto-char (scan-lists (point) -1 (car state)))
                (backward-up-list)
                (setq par (1+ (current-column)))
                (throw 'nesting 'contexp)
                )
```

this way it ignores the following code which was confused by the unbalanced parentheses.
We are interested only in indenting the open parentheses: I wonder why pascal-mode doesn't...

#5 - 15 Feb 2013 17:53 - Anna Maria Bigatti

- Target version changed from CoCoA-5.0.9 to CoCoA-5.0.3

#6 - 21 May 2013 17:34 - John Abbott

- Status changed from Feedback to Closed

There has been no negative feedback in 3 months, so I'm closing this issue.

```
                ((> (car state) 0)
;;                  (goto-char (scan-lists (point) -1 (car state)))
                (backward-up-list)
                (setq par (1+ (current-column)))
                (throw 'nesting 'contexp)
```