CoCoA-5 - Bug #1594

Parser bug: missing close square bracket

08 May 2021 16:53 - Anna Maria Bigatti

Status:	Closed	**Start date:**	08 May 2021
Priority:	High	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	Parser/Interpreter	**Estimated time:**	3.51 hours
Target version:	CoCoA-5.4.0	**Spent time:**	3.40 hours

Description

```
/**/ C := record[ A:= 1;
--> ERROR: Expected a comma or a closed square bracket to end the record definition
--> C := record[ A:= 1;
-->                   ^
libc++abi.dylib: terminating with uncaught exception of type CoCoA::ParserNS::AskingForNewInteract
iveInputDuringRecoveryException: std::exception

Process cocoa5 abort trap: 6
```

investigate

Related issues:

Related to CoCoA-5 - Bug #1595: Bad input causes crash	**Closed**	**13 May 2021**

History

#1 - 10 May 2021 10:42 - John Abbott

I confirm the bug exists also for the Linux version.

#2 - 10 May 2021 10:57 - John Abbott

After some *ad hoc* testing, it seems that the bug appears only when using CoCoA-5 through the Emacs interface.

gdb detected nothing strange; valgrind detected nothing strange (compiled for debugging)

Odd! Now I must prepare lectures.

#3 - 11 May 2021 12:06 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The exception is thrown at line 81 in LineProviders.C, and ought to have been caught
in line 376 of Parser.C. Maybe the catch is in the wrong place?

The bug appears only when CoCoA-5 is run with the option --no-readline; the readline version of line-provider does not throw the exception.
So I wonder if it is ever useful to throw it... :-/

#4 - 11 May 2021 12:10 - John Abbott

Here are some more inputs which cause trouble:

```
a := record[b)
a := record[b:=1)
```

I think what is happening is that the parser is scanning ahead looking for a semicolon (which is not present on the line).
The original test input did contain a semicolon, but it was "consumed" as the syntax error was discovered, so the parser scanned ahead for another semicolon...
so the original failing input does not cause a crash if a second semicolon is appended:

```
a := record[B := 1;;
```

#5 - 11 May 2021 12:12 - Anna Maria Bigatti

- Subject changed from Parser bug on MacOS: missing close square bracket to Parser bug: missing close square bracket

#6 - 12 May 2021 10:45 - John Abbott

Some relevant source code is in Parser.C around line 1990, in the fn **Parser::parseRecord**

#7 - 12 May 2021 14:06 - John Abbott

Here are some more failing cases:

```
[1,]
a := L[1,]
```

#8 - 12 May 2021 16:44 - John Abbott

Huh???

If I run CoCoA inside emacs (so with readline disabled) and type in [1,] as input then CoCoA crashes.
But if I put [1,] as an input line in a cocoa5-mode buffer and send the line then it does not crash.
Very odd! What is the difference?

If I cut-and-paste from an emacs buffer into cocoa running in a terminal then it crashes.

#9 - 12 May 2021 16:57 - John Abbott

It is really annoying that CoCoA-5 treats inputs via different mechanisms differently.
Anyway, I have some new failing inputs:

```
a,
[,
```

#10 - 13 May 2021 19:54 - John Abbott

The bug seems to vanish if I disable the "fancy" delayed prompt code (which creates a new thread).
I am vaguely aware that threads and exceptions are tricky mixture... sigh.
I do see a potential for a dangling reference, though I thought the code would be safe...

#11 - 13 May 2021 20:18 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 10 to 50

I have rewritten the "fancy delayed prompt" code, and now the problem seems to have gone away (after some rather minor testing).
I have checked in my revised code (in LineProviders.C).

#12 - 13 May 2021 21:59 - John Abbott

- Related to Bug #1595: Bad input causes crash added

#13 - 14 May 2021 16:34 - Anna Maria Bigatti

John Abbott wrote:

> I have rewritten the "fancy delayed prompt" code, and now the problem seems to have gone away (after some rather minor testing).
> I have checked in my revised code (in LineProviders.C).

It works fine now on my Mac.

#14 - 21 May 2021 17:04 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 50 to 90

#15 - 14 Sep 2021 11:46 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 3.51 h