

CoCoA-5 - Bug #182

Unescaped double quote inside superstring

08 Jun 2012 19:30 - John Abbott

Status:	Closed	Start date:	08 Jun 2012
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Parser/Interpreter	Estimated time:	7.50 hours
Target version:	CoCoA-5.1.1 Seoul14	Spent time:	0.75 hour
Description			
<p>The parser goes into a strange state if an unescaped double quote character is found inside a superstring. It correctly reports an error, but then seems still to be inside the superstring.</p>			
<pre># ""abc"def""; ERROR: Expected a double-quote character; you've either forgotten a backslash to escape this quote , or two double quote characters to end the multi-line string literal ""abc"def""; ^ ... ; #</pre>			
<p>Decide what the correct behaviour should be, and implement it.</p>			
Related issues:			
Related to CoCoA-5 - Design #473: Multiline string literals - useful or obsol...		Closed	13 Mar 2014
Related to CoCoA-5 - Design #576: Disallow juxtaposition for string literals?		Closed	25 Jun 2014
Related to CoCoA-5 - Feature #1587: Multiline string literals (again)		Closed	02 Apr 2021

History

#1 - 23 Jan 2013 12:32 - Anna Maria Bigatti

- Category set to Parser/Interpreter

#2 - 02 Apr 2014 17:34 - Anna Maria Bigatti

- Target version set to CoCoA-5.1.0 Easter14

#3 - 09 Apr 2014 14:54 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.1.1 Seoul14

#4 - 25 Jun 2014 12:52 - John Abbott

- Status changed from New to In Progress

If I recall correctly, we abolished "superstrings" in CoCoA-5.1. So the problem is no longer relevant. In any case, the behaviour now is different:

```
>>> ""abc"def"";
--> ERROR: Unexpected identifier, are you forgetting a "*" or a ";" ?
--> ""abc"def"";
-->      ^^^
```

The interpreter sees the input as "" (empty literal) followed by "abc" followed by the identifier def.

I continue to have doubts about the wisdom of allowing juxtaposed string literals; I wonder if we ever documented why we wanted it? If we disallow juxtaposition, then the error will be signalled at the third ", which is probably more helpful to the user.

#5 - 26 Jun 2014 12:06 - John Abbott

- % Done changed from 0 to 50

I have (temporarily?) disabled recognition of juxtaposed strings. The resulting behaviour on the example input is:

```
>>> """abc"def""";  
--> ERROR: Expecting an operator, or a semicolon to end the statement  
--> """abc"def""";  
-->    ^^^^
```

I think this is sufficiently clear and helpful.

#6 - 26 Jun 2014 15:11 - John Abbott

- Status changed from *In Progress* to *Closed*

- Assignee set to *John Abbott*

- % Done changed from 50 to 100

This bug essentially vanished when we abolished multiline string literals.

Now that juxtaposed strings literals are being abolished, I think the overall behaviour is fine.

So closing.

#7 - 02 Apr 2021 11:06 - John Abbott

- Related to Feature #1587: *Multiline string literals (again) added*