CoCoA-5 - Bug #1294

CoCoA-5 tests may fail even if executed correctly

06 Jun 2019 17:55 - Julian Danner

Status:	Closed	**Start date:**	06 Jun 2019
Priority:	Low	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Portability	**Estimated time:**	2.75 hours
Target version:	CoCoA-5.4.0	**Spent time:**	2.85 hours

Description

If I compile Cocoa5 and one of the packages loaded automatically (so any package-file inside of src/CoCoA-5/packages or some package source by the init.cocoa5 file) throws a warning, all the tests will display a fail, despite being executed correctly. A simple fix is to add the flag -w0 to the call of CoCoAInterpreter in line 54 of src/CoCoA-5/tests/RunTests.sh.
If however, one of these packages even throws an error, this fix will not help, as in that case it seems that not even any test computation is started, but in that case this error should be resolved either way... More importantly, if some package-exported function has the same name as a variable used in some test, that test is exited 'incorrectly' with an error and marked as failed.

The error occured, since I am working both on CoCoALib-functions which I made available to Cocoa5 and some Cocoa5-packages which are sourced from init.cocoa5 (which lies in that package folder).
(To be honest I don't think its good practice to join my packages with the library ones in this way, however it would also be very annoying if I would need to load all my packages manually each time Cocoa5 is started...)

Maybe, it would be good to move the init-file to another location which is therefore not loaded by only giving the path to the packages folder. In that case one just needs to ensure that also this init-file is loaded within the cocoa5 shell file.

Related issues:

Related to CoCoA-5 - Design #1496: Use "special" names for variables inside t...	**Rejected**	**02 Oct 2020**

History

#1 - 09 Jan 2020 20:29 - John Abbott

- Category set to Portability

- Target version set to CoCoA-5.?.?

Not sure how to improve the situation here.

In a sense, all tests passing should mean that CoCoA-5 is "clean"; if some package is triggering a warning then it is not "clean".
I see that it could seem excessive to report failure for every test for such a "minor matter".

I suppose we could run an "empty test", and report any warnings that gives; then run all other tests with warnings disabled.
If the "empty test" fails, then we would not attempt the other tests (since CoCoA-5 failed to start properly).

Would this be a reasonable compromise?

#2 - 15 Sep 2020 14:35 - Julian Danner

Yes, I think that would be an improvement, but also only solve the issue partly. Still if for instance init.cocoa5 (or any other standard package) loads a protected variable name that is also used in one of the tests, that specific test will for sure fail.

#3 - 15 Sep 2020 15:20 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Your comment about a protected variable causing trouble is a good point (albeit also rather abstruse).

We should use only "funny" variable names in the tests, and presumably "funny" function names (if we need to define a function).

One possibility is simply to use names which start with underscore: this is allowed and was used in CoCoA4 for something (hidden package values?).
Perhaps even double underscore? That would be quite unusual but not impinge too badly on readability.

It might be nicer to use names which depend on the test file: *e.g.* in exbugs.cocoa5 we could use names like exbug_f. This reduces the chance of a name clash, but also reduces readability.

MOVED TO NEW ISSUE #1496

#4 - 17 Sep 2020 15:59 - John Abbott

- % Done changed from 10 to 20

I have modified the script RunTests.sh so that it first runs CoCoA with an empty input (/**dev**/**null**).
If that produces an error (*i.e.* non-zero exit code) then the script prints a message and exits.

Otherwise it continues.

I have just tried with an empty package (no exported names). This produces a warning, which is printed on **cout** (rather than **cerr** or **clog**);
the exit code is 0 (so the shell script thinks that CoCoA has run fine). This "unexpected" message will cause all tests to fail (with "wrong output").

Options to consider:

- **(A)** modify the RunTests.sh script so that if empty input produces any output then the script exits
- **(B)** print warnings on cerr or clog, then RunTests.sh could be modified to ignore output on cerr
- **(C)** take the output from the "empty" run, and prepend it to the expected output, complain only if that differs from actual output obtained
- **(D)** other ideas?

#5 - 23 Sep 2020 10:42 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 20 to 50

I have implemented option **(A)** and checked it in. I chose **(A)** because it is relatively simple to implement.
Waiting for feedback to see if **(A)** is reasonable or whether one of the other options is preferable.

#6 - 02 Oct 2020 11:53 - John Abbott

- Related to Design #1496: Use "special" names for variables inside tests added

#7 - 10 Nov 2021 19:41 - John Abbott

- Status changed from Resolved to Feedback

- Assignee set to John Abbott

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0

- % Done changed from 50 to 90

I made the change about a year ago (according to redmine), and no complaints have surfaced.
So advancing to **feedback**. Maybe close in a month?

#8 - 03 Feb 2022 19:40 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 2.75 h

No complaints in several months... must be OK.
Closing!