CoCoA-5 - Feature #1289

assert-function for cocoa5

27 May 2019 13:19 - Julian Danner

Status:	Closed	**Start date:**	27 May 2019
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	CoCoA-5 function: new	**Estimated time:**	1.23 hour
Target version:	CoCoA-5.3.0	**Spent time:**	1.30 hour

Description

For debugging purposes an 'assert'-function comparable to COCOA_ASSERT (from CoCoALib) would be useful.

(One can already replicate similar behaviour by making use of VerbosityLevel(), however this is not very convenient and moreover does not directly reflect the idea of verbosity itself.)

History

#1 - 27 May 2019 14:08 - Anna Maria Bigatti

- Status changed from New to In Progress

- % Done changed from 0 to 10

True. In fact, most of tests in src/CoCoA-5/tests contain this function (which does a few more things)

```
define TEST_ASSERT(A,B)
  toplevel TestCount;
  toplevel PrintInfo;
  TestCount := TestCount+1;
  If A<>B Then
    error("TEST: " + Sprint(A) + " <> " + Sprint(B));
  endif;
  if PrintInfo then print "."; EndIf;
enddefine; -- TEST_ASSERT
```

I'll think how to make it general.

Name: should it be assert (as for error, coming from CoCoA_ERROR)

#2 - 28 May 2019 10:32 - John Abbott

- Category set to CoCoA-5 function: new

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

Would someone like to post a couple of hypothetical examples of how you might like to use this feature (so that we can consider design options)?

A very simple implementation is:

```
define assert(CondToCheck)
  if not(CondToCheck) then error("assertion failed"); endif;
enddefine;
```

The standard CoCoA-5 error reporting mechanism should then indicate the source-code line where the failing call to assert is. Unfortunately it does not indicate the the actual source code of the call.

However, if assert were a built-in function then the standard CoCoA-5 error reporting mechanism should print out the actual source of the failing assert.

#3 - 28 May 2019 10:38 - John Abbott

- % Done changed from 10 to 20

Here is a potential impl:

```
DECLARE_STD_BUILTIN_FUNCTION(assert, 1) {
  intrusive_ptr<BOOL> arg = runtimeEnv->evalArgAs<BOOL>(ARG(0));
  if (!(arg->theBool))
    throw RuntimeException("assertion failed", invocationExpression);
  return VoidValue::theInstance;
}
END_STD_BUILTIN_FUNCTION
```

I have just tried it, and it worked as expected in a simple test.
I put the defn into the file BuiltinFunctions.C

Maybe I can give Julian a copy of my private development version of CoCoA so he can test it...

#4 - 29 May 2019 11:03 - Julian Danner

Thank you, it works flawlessly (with the C++ impl).
However, I think it would be better it there would be some kind of debugging-flag such that only in 'debugging'-mode the assert-statements are evaluated. With the current setup, we perform these computations in any case. As far as I know CoCoA_ASSERT is also only evaluated if compiled

with -DCoCoA_DEBUG.

#5 - 29 May 2019 11:28 - John Abbott

May I ask why you want to have assertions which can be globally enabled/disabled?
[I know that this is what we do in CoCoALib, but I enable assertions only when I need help debugging].
I suppose one could claim that assertions are "active comments", but normally
for argument checks I prefer an explicit test with error-throw.

Implementing "delayed evaluation" in the interpreter would be a major change (and probably beyond my abilities).

You can achieve something similar to what you want, but the syntax is messy:

```
TopLevel AssertFlag;
 ...
if AssertFlag then assert(...); endif
```

Alternatively you can use func..endfunc to delay evaluation:

```
assert(func() ImportByValue X; ImportByValue Y; return X=Y; endfunc);
```

This second version is not very readable :-(

#6 - 20 Feb 2020 10:02 - John Abbott

- Target version changed from CoCoA-5.?.? to CoCoA-5.3.0

Should assert be in 5.3.0 release?
I've changed target to 5.3.0 so that we discuss it!

#7 - 20 Feb 2020 16:16 - John Abbott

- Status changed from In Progress to Feedback

- Assignee set to John Abbott

- % Done changed from 20 to 90

- Estimated time set to 1.23 h

I have checked that **assert** exists and works, and is documented...
JAA thinks it is reasonable to include it in 5.3.0.

#8 - 26 Feb 2020 16:32 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100