

## CoCoA-5 - Feature #708

### ExternalLib Normaliz: verbose flag?

17 May 2015 09:55 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	17 May 2015
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	enhancing/improving	<b>Estimated time:</b>	3.01 hours
<b>Target version:</b>	CoCoA-5.2.0 spring 2017	<b>Spent time:</b>	3.05 hours
<b>Description</b>			
Is it possible to make Normaliz be verbose from the CoCoA-5 interface? Christof says he/they often find it help to see the verbose messages from Normaliz.			
I note that in example ex-normaliz1.C there is a commented out reference to libnormaliz::verbose so I suppose that it is possible to get such messages when calling from CoCoALib.			
<b>Related issues:</b>			
Related to CoCoALib - Feature #962: General verbose mode?		<b>Closed</b>	<b>07 Nov 2016</b>

### History

#### #1 - 17 May 2015 10:12 - John Abbott

Some ideas are:

- a global flag which says what the default verbosity is
- an optional arg to calls to Normaliz which indicates the verbosity for that call

Where are verbose messages printed? Presumably on whatever output stream CoCoA-5 is using.

How many levels of verbosity are there?

- just 2? in which case we could use a boolean
- more than 2? in which case we should probably use a (non-negative) integer

#### #2 - 18 May 2015 15:20 - John Abbott

Christof says that Normaliz uses a bool.

Should CoCoALib simply follow what Normaliz does for its normaliz related flags?  
And follow what Froby does for its froby related flags?  
etc?

#### #3 - 30 Jun 2015 11:31 - Anna Maria Bigatti

- Target version changed from CoCoA-5.1.2 summer 2015 to CoCoA-5.1.3/4 Jan 2016

#### #4 - 17 Feb 2016 15:33 - John Abbott

Let's get this sorted out for 5.1.4!

**#5 - 17 Feb 2016 15:33 - John Abbott**

- Target version changed from CoCoA-5.1.3/4 Jan 2016 to CoCoA-5.2.0 spring 2017

**#6 - 17 Feb 2016 15:34 - John Abbott**

- Priority changed from Normal to High

**#7 - 18 Feb 2016 17:17 - Christof Soeger**

The cone object in cocoalib has a SetVerbose method. But that is not so useful from cocoa5 since we don't have an object there.

There is also the possibility to set the global verbose default value which is useful from c5 and I already implemented NmzSetVerboseDefault as a one-line built-in function. It is even documented and is working!

One point needs improvement: The output is written to std::cout. This can be changed with the libnormaliz functions

```
void setVerboseOutput (std::ostream&);  
void setErrorOutput (std::ostream&);
```

but where/when should it be done and set to what?

**#8 - 05 Sep 2016 15:40 - John Abbott**

- Status changed from New to In Progress

- % Done changed from 0 to 10

Perhaps we can finish this on Wednesday (2016-09-07)?

**#9 - 28 Apr 2017 09:13 - Anna Maria Bigatti**

- Estimated time set to 1.01 h

Should we pass the VerbosityLevel from CoCoALib?

**#10 - 28 Apr 2017 11:32 - John Abbott**

- % Done changed from 10 to 50

JAA prefers that there be independent verbosity settings for CoCoALib and any external libraries.

**#11 - 28 Apr 2017 12:16 - Anna Maria Bigatti**

should we call it NmzSetVerbosityLevel?

**#12 - 28 Apr 2017 13:41 - John Abbott**

If the behaviour from the point of view of a user is similar to that of CoCoA's own SetVerbosityLevel then it would make sense to give it a similar name. Presumably there would be just two levels: 0 and 1 (or anything positive).

I think there may be a subtle difference for a CoCoALib user: if I recall well, each cone may have its own "verbosity" setting (but the default value is taken from the system/global setting).

**#13 - 28 Apr 2017 16:02 - Anna Maria Bigatti**

John Abbott wrote:

If the behaviour from the point of view of a user is similar to that of CoCoA's own SetVerbosityLevel then it would make sense to give it a similar name. Presumably there would be just two levels: 0 and 1 (or anything positive).

In view of doing something similar for other external libraries, I propose the general rule

- XxxSetVerbosityLevel/XxxVerbosityLevel

- levels 0 and 10 (and above)

(as we say that values 1..9 are reserved for the users)

I think there may be a subtle difference for a CoCoALib user: if I recall well, each cone may have its own "verbosity" setting (but the default value is taken from the system/global setting).

Then we say in the documentation that these functions are for the default levels.

**#14 - 28 Apr 2017 16:32 - John Abbott**

Currently Normaliz has just two verbosity levels (true and false).

There is not much problem in saying that numerical levels below 10 correspond to false, an higher levels correspond to true. **BUT** what happens if Normaliz also decides to have several numerical levels of verbosity? For instance, they may have levels 0-5 inclusive. How do we map those numerical levels to CoCoA?

**#15 - 28 Apr 2017 16:46 - Anna Maria Bigatti**

John Abbott wrote:

Currently Normaliz has just two verbosity levels (true and false).

There is not much problem in saying that numerical levels below 10 correspond to false, an higher levels correspond to true. **BUT** what happens if Normaliz also decides to have several numerical levels of verbosity? For instance, they may have levels 0-5 inclusive. How do we map those numerical levels to CoCoA?

as you wish, the important thing for me is the names: I don't want to confuse users with different kindof names when we are just advertising the new feature.

**#16 - 28 Apr 2017 19:21 - John Abbott**

OK for the names XXXSetVerbosityLevel and XXXVerbosityLevel where XXX could be Nmz or some other prefix for an external library.

I would like to consult briefly with the Normaliz people what they would like to have, since I expect they will be the main users of "verbosity" for Normaliz. I'm not sure if any of the other external libraries has a verbosity setting.

Winfried, we would like to use non-negative integers to indicate verbosity levels (where higher numbers trigger printing of more information, up to some limit). Currently, all built-in CoCoA commands produce no "verbose" output with a setting below 10... this allows users to employ values up to 9 for their own code.

I would like to have separate verbosity levels for CoCoALib and for "external libraries" (such as Normaliz).

There appear to be two main options (assuming we use integer verbosity levels):

- **(A)** use levels 0 and 1 for Normaliz
- **(B)** use levels 0 and 10 (or some higher threshold) for Normaliz

In either case, a setting below the threshold would correspond to Normaliz verbosity false, and any setting greater than or equal to the threshold would correspond to true (unless/until Normaliz employs more than 2 levels of verbosity).

No CoCoA function should produce any "verbose output" just because NmzVerbosityLevel is positive; in other words, the CoCoA verbosity setting and the Normaliz one are completely independent.

Comments?

**#17 - 28 Apr 2017 20:15 - Winfried Bruns**

Normaliz has only two levels of verbosity: off or on. Via libnormaliz this can be toggled. I think the CoCoA interface to Normaliz then needs a function that switches verbose on or off for Normaliz.

I have thought about introducing three levels of verbosity: off, the most important steps, everything. But this has really low priority.

**#18 - 28 Apr 2017 20:18 - Anna Maria Bigatti**

Done.

Two functions (4 in cocoalib, for default and for cone): NmzSetVerbosityLevel/NmzVerbosityLevel, as in cocoa/cocoalib behaviour.

NmzSetVerbosityLevel takes an INT but for now converts 0 to off, and >0 to on.

Documented.

Close?

**#19 - 28 Apr 2017 20:21 - Anna Maria Bigatti**

- Estimated time changed from 1.01 h to 3.01 h

**#20 - 28 Apr 2017 20:22 - Anna Maria Bigatti**

- Status changed from *In Progress* to *Feedback*
- Assignee set to *John Abbott*
- % Done changed from 50 to 90

**#21 - 28 Apr 2017 20:22 - Anna Maria Bigatti**

- Related to Feature #962: *General verbose mode?* added

**#22 - 05 May 2017 10:02 - John Abbott**

- Status changed from *Feedback* to *Closed*
- % Done changed from 90 to 100

**#23 - 05 May 2017 10:36 - Anna Maria Bigatti**

Winfried Bruns wrote:

I have thought about introducing three levels of verbosity: off, the most important steps, everything. But this has really low priority.

This confirms it has been a good idea to have the Normaliz verbosity implementation in CoCoALib accepting INT (as for the CoCoA verbosity implementation).