

CoCoA-5 - Feature #1731

Verbosity level in CoCoA5

27 Mar 2023 19:41 - Anna Maria Bigatti

Status:	In Progress	Start date:	27 Mar 2023
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	70%
Category:	enhancing/improving	Estimated time:	0.00 hour
Target version:	CoCoA-5.4.4	Spent time:	3.70 hours
Description			
We should provide a mechanism to mimick verbosity in CoCoA-5, so that it looks similar to that in CoCoALib			
Related issues:			
Related to CoCoALib - Design #1721: Verbose messages: print level?			Closed 19 Jan 2023

History

#1 - 27 Mar 2023 19:41 - Anna Maria Bigatti

- Related to Design #1721: Verbose messages: print level? added

#2 - 27 Mar 2023 19:45 - Anna Maria Bigatti

I made a test in the package monomial\_ideals.

```
define IsVerbosePrint(level, FnName)
  if VerbosityLevel() < level then return false; endif;
  print "[L", level, ",", FnName, "]"  ";
  return true;
enddefine; -- IsVerbosePrint
```

then called

```
if IsVerbosePrint(50,"TryPrecisions") then
  ...
endif;
```

We can do better than this, but it is better than before.  
I should make some more experiments before taking it out for public use (interface will likely change)

#3 - 27 Mar 2023 19:48 - Anna Maria Bigatti

- Target version changed from CoCoA-5.4.2 to CoCoA-5.4.4
- % Done changed from 0 to 20

#### #4 - 12 Apr 2023 14:50 - Anna Maria Bigatti

- Status changed from New to In Progress
- Assignee set to Anna Maria Bigatti
- % Done changed from 20 to 70

New approach: easier to use, more similar to CoCoALib and sophisticated use of CoCoA-5 anonymous functions (func) ;-)

[ is it possible also to add depth (and therefore indentation)? I don't think so :-/ ]

```
define IsVERBOSE_FnName(level, FnName)
  if VerbosityLevel() < level then return false; endif;
  print "[L", level, ", ", FnName, "]" ";
  return true;
enddefine; -- IsVERBOSE_FnName

define IsVERBOSE_func(FnName)
//  return func(l) return IsVERBOSE_FnName(l,FnName); endfunc;
  return func(l)
    ImportByValue FnName;
    if VerbosityLevel() < l then return false; endif;
    print "[L", l, ", ", FnName, "]" ";
    return true;
  endfunc;
enddefine; -- IsVERBOSE_func
```

#### #5 - 12 Apr 2023 22:05 - John Abbott

Not sure if it is really a good idea....

I remind you that it is possible to have global variables in packages. May this could be used to keep track of the depth?

A disadvantage is that the variable could be left with a non-zero value if the user interrupts.

Maybe indentation could be limited to 10 (or 20?), so even if a "large" value is left in the global, it won't affect too adversely the printing...

#### #6 - 10 May 2023 20:53 - John Abbott

Anna pointed out verbally that my idea in comment 5 (above) cannot easily be made to work, because the "depth counter" needs to be automatically incremented/decremented as functions are called/return.

I think it might be helpful to put 1 or 2 spaces at the start of each line printed as "verbose" output; this would help distinguish it from normal output (which typically does not use indentation, unless you use **indent**). What do you think?

John Abbott wrote:

I think it might be helpful to put 1 or 2 spaces at the start of each line printed as "verbose" output; this would help distinguish it from normal output (which typically does not use indentation, unless you use **indent**). What do you think?

A good idea. I made a few tests and decided for this:  
2 spaces for CoCoA-5  
2 spaces + depth (therefore  $\geq 3$ spaces) for CoCoALib

Try this to test a rich output:

```
/**/ G := [x^2, x*y - z^2, y^2];  
/**/ SetVerbosityLevel(90);  
/**/ sagbi(G);
```