CoCoALib - Bug #756

frobby (v0.9.0) does not compile with g++-4.8

31 Jul 2015 16:01 - John Abbott

Status:	Closed		Start date:	31 Jul 2015
Priority:	Normal		Due date:	
Assignee:	John Abbott		% Done:	100%
Category:	Portability		Estimated time:	0.00 hour
Target version:	CoCoALib-0.99560		Spent time:	2.25 hours

Description

frobby fails to compile with g++ version 4.8.x

History

#1 - 31 Jul 2015 16:03 - John Abbott

There are missing includes in main.cpp and randomDataGenerators.cpp; you need to add:

```
#include <sys/typs.h>
#include <unistd.h>
```

Also in StatisticsStrategy.cpp:143 change the type mpz_class into mpq_class.

#2 - 31 Jul 2015 16:20 - John Abbott

- *Status changed from New to Resolved*

- *Assignee set to John Abbott*

- *% Done changed from 0 to 80*

JAA has made the changes in the previous note, and all seems to work.

JAA has made a revised TGZ file: frobby_0.9.1.tgz with these changes.

Should we tell Bjarke?

#3 - 08 Jun 2016 15:46 - Anna Maria Bigatti

Could you write the changes in the documentation ExternalLibs-Frobby.txt?

#4 - 10 Jun 2016 16:09 - John Abbott

OK, I have changed the documentation. Could you check to see if it makes sense?

#5 - 10 Jun 2016 16:50 - Anna Maria Bigatti

John Abbott wrote:

OK, I have change the documentation. Could you check to see if it makes sense?

I believe I followed the instructions, but I still have problems like non recognizing
std::_Construct; std::_Destroy, std::__iterator_category....

Do you recognize this problem? (otherwise I investigate myself)

#6 - 10 Jun 2016 16:55 - John Abbott

I do not recognise the error message you have posted.

Do you have to tell your compiler to use the C++03 standard rather than the C++11 standard?

BTW: I have just checked in a new file ExternalLibs-GMP.txt since there are some useful notes about compiling GMP (that's why I could not get CoCoALib to run on the netbook)

#7 - 10 Jun 2016 17:08 - Anna Maria Bigatti

I've found this ;-)
[[https://github.com/Macaulay2/M2/issues/66]]

I copy here the reply from Bjarke Roune ===============
As I recall from that time, old GCC had hash_map but new GCC did not - they removed it! I wanted the code to compile on both the new and old gcc so I copied the file. At this point it might be reasonable to drop support for that older GCC, in which case I believe hash_map could just be replaced with std::unordered_map from the C++11 standard library, which at this point should be available in most compilers. I vaguely remember that there might have been some #ifdef'ing to do different things on different compilers here, so the fix might be to just change those #ifdefs. This is just what I remember, but I can look further into it if that isn't enough information. ===============

#8 - 10 Jun 2016 17:21 - John Abbott

Can you tell your compiler (clang/llvm) to use the old standard? You will have to look in the documentation to see how (but it might simply be --std=c++03).

I have got a "clean compile" here with g++ version 5.3.1 both with and without --std=c++11, though with the option set to specify C++11 there are lots of warnings about auto_ptr being deprecated.

#9 - 10 Jun 2016 17:32 - Anna Maria Bigatti

John Abbott wrote:

Can you tell your compiler (clang/llvm) to use the old standard? You will have to look in the documentation to see how (but it might simply be --std=c++03).

I tried, but I get the same error:

```
src/hash_map/hashtable.h:79:14: error: no member named '_Construct' in namespace
      'std'
  using std::_Construct;
        ~~~~~^
```

Indeed I cannot find _Construct in /Library/Developer/CommandLineTools/usr/include/c++/v1/

#10 - 22 May 2017 13:38 - Redmine Admin

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99560

#11 - 06 Nov 2017 16:48 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

JAA has a temporary version of FROBBY available in frobby_v0.9.3.tgz

This seems to avoid the portability problems of the original code.

Closing.

```
src/hash_map/hashtable.h:79:14: error: no member named '_Construct' in namespace
      'std'
  using std::_Construct;
        ~~~~~^
```