

## CoCoALib - Support #861

### Janet basis code: TmpJB files give some problems with C++11 (using CLANG/LLVM)

31 Mar 2016 13:08 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	31 Mar 2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Portability	<b>Estimated time:</b>	3.33 hours
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	3.35 hours
<b>Description</b>			
Bruns has reported compilation errors in the Janet Basis code (files called TmpJB...) when compiling with the flag <b>-std=c++11</b> on a recent Macintosh which uses the clang/llvm compiler.			
Clean the JB code so that it compiles fine also on that platform (hopefully without having to introduce #ifdef trickery)			
<b>Related issues:</b>			
Related to CoCoALib - Feature #82: C++11 compatibility questions		<b>Closed</b>	<b>26 Jan 2012</b>

## History

### #1 - 31 Mar 2016 13:10 - John Abbott

Bruns reported the following errors during compilation:

```
In file included from SparsePolyRing.C:27:
In file included from ../../include/CoCoA/DUPFp.H:23:
In file included from ../../include/CoCoA/factorization.H:28:
In file included from ../../include/CoCoA/VectorOperations.H:32:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:212:
9: error:
    field has incomplete type 'CoCoA::Involutive::JanetHandle'
    _Tp __value_;
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/type_traits:1839:27: note:
    in instantiation of template class 'std::__1::__list_node<CoCoA::Involutive::JanetHandle, void *>' requested here
decltype(__is_constructible_test(declval<_Tp>(), declval<_Args>()...))
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/type_traits:1888:14: note:
    in instantiation of template class 'std::__1::__is_constructible<false,
std::__1::allocator<std::__1::__list_node<CoCoA::Involutive::JanetHandle, void *>, >' requested here
: public __is_constructible<is_scalar<_Tp>::value || is_reference<_Tp>::value,
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/type_traits:1914:14: note:
    in instantiation of template class 'std::__1::__is_constructible_void_check<false,
std::__1::allocator<std::__1::__list_node<CoCoA::Involutive::JanetHandle, void *>, >' requested here
: public __is_constructible_void_check<__contains_void<_Tp, _Args...>::value
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/type_traits:2394:34: note:
    in instantiation of template class
'std::__1::is_constructible<std::__1::allocator<std::__1::__list_node<CoCoA::Involutive::JanetHandle, void *>
>,
>' requested here
: __is_nothrow_constructible<is_constructible<_Tp, _Args...>::value, _Tp, _Args...>
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/type_traits:2516:14: note:
    in instantiation of template class
'std::__1::is_nothrow_constructible<std::__1::allocator<std::__1::__list_node<CoCoA::Involutive::JanetHandle,
void *>, >' requested here
: public is_nothrow_constructible<_Tp>
      ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:796:
```

```

20: note:
    in instantiation of template class
'std::__1::is_nothrow_default_constructible<std::__1::allocator<std::__1::__list_node<CoCoA::Involutive::JanetHandle,
void *> >' requested here
_NOEXCEPT_(is_nothrow_default_constructible<__node_allocator>::value)
^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/__config:
287:34: note:
    expanded from macro '_NOEXCEPT_'
# define _NOEXCEPT_(x) noexcept(x)
^
../../include/CoCoA/TmpJBDatastructure.H:404:13: note: in instantiation of exception specification for 'list'
requested
    here
        myArm() {
            ^
../../include/CoCoA/TmpJBDatastructure.H:28:11: note: forward declaration of 'CoCoA::Involutive::JanetHandle'
    class JanetHandle;
    ^
In file included from SparsePolyRing.C:27:
In file included from ../../include/CoCoA/DUPFP.H:23:
In file included from ../../include/CoCoA/factorization.H:28:
In file included from ../../include/CoCoA/VectorOperations.H:32:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:679:
24: error:
    non-const lvalue reference to type '__node_base' (aka '__list_node_base<value_type, __void_pointer>') cannot bind
    to a value of unrelated type 'std::__1::__list_node<CoCoA::Involutive::JanetHandle, void *>'
        __unlink_nodes(*__f, *__l->__prev_);
        ^~~~~
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:664:
5: note:
    in instantiation of member function 'std::__1::__list_imp<CoCoA::Involutive::JanetHandle,
std::__1::allocator<CoCoA::Involutive::JanetHandle> >::clear' requested here
    clear();
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:769:
24: note:
    in instantiation of member function 'std::__1::__list_imp<CoCoA::Involutive::JanetHandle,
std::__1::allocator<CoCoA::Involutive::JanetHandle> >::~~__list_imp' requested here
class _LIBCPP_TYPE_VIS list
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:527:
45: note:
    passing argument to parameter '__f' here
    static void __unlink_nodes(__node_base& __f, __node_base& __l) _NOEXCEPT;
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1667:
:30: error:
    non-const lvalue reference to type '__node_base' (aka '__list_node_base<value_type, __void_pointer>') cannot bind
    to a value of unrelated type '__node' (aka '__list_node<value_type, __void_pointer>')
base::__unlink_nodes(const_cast<__node&>(*__f.__ptr_), *__l.__ptr->__prev_);
    ^~~~~~
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1253:
:9: note:
    in instantiation of member function 'std::__1::list<CoCoA::Involutive::JanetHandle,
std::__1::allocator<CoCoA::Involutive::JanetHandle> >::erase' requested here
    erase(__i, __e);
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1166:
:9: note:
    in instantiation of function template specialization 'std::__1::list<CoCoA::Involutive::JanetHandle,
std::__1::allocator<CoCoA::Involutive::JanetHandle>
::assign<std::__1::__list_const_iterator<CoCoA::Involutive::JanetHandle, void *> >' requested here
    assign(__c.begin(), __c.end());
    ^
../../include/CoCoA/TmpJBDatastructure.H:435:15: note: in instantiation of member function
'std::__1::list<CoCoA::Involutive::JanetHandle, std::__1::allocator<CoCoA::Involutive::JanetHandle> >::operator='
requested here
    myArm = NextArm;
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:527:

```

```
45: note:
    passing argument to parameter '__f' here
    static void __unlink_nodes(__node_base& __f, __node_base& __l) _NOEXCEPT;
                                   ^
```

3 errors generated.

```
make[2]: *** [SparsePolyRing.o] Error 1
```

```
***** Compilation failed in CoCoALib source subdirectory AlgebraicCore/ *****
```

```
make[1]: *** [library] Error 1
```

```
make: *** [library] Error 2
```

Winfried

Am 30.03.2016 um 15:35 schrieb Winfried Bruns:

[Hide Quoted Text]

Dear John,

we are just preparing Normaliz 3.1.1. On Mac we tried to compile CoCoAKib after the following configuration:

```
./configure --with-cxx=clang++ --with-cxxflags="-std=c++11 -stdlib=libc++ -mmacosx-version-min=10.7"
```

Our CLang is

Apple LLVM version 5.0 (clang-500.2.79) (based on LLVM 3.3svn)

Target: x86\_64-apple-darwin13.3.0

Thread model: posix

It shows a problem with another "JB":

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:679:
24: error:
```

```
non-const lvalue reference to type '__node_base' (aka '__list_node_base<value_type, __void_pointer>') cannot bind
```

```
to a value of unrelated type 'std::__1::__list_node<CoCoA::Involution::JanetHandle, void *>'
```

```
__unlink_nodes(*__f, *__l->__prev_);
               ^~
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:664:
5: note:
```

```
in instantiation of member function 'std::__1::__list_imp<CoCoA::Involution::JanetHandle, std::__1::allocator<CoCoA::Involution::JanetHandle> >::clear' requested here
```

```
clear();
^
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:769:
24: note:
```

```
in instantiation of member function 'std::__1::__list_imp<CoCoA::Involution::JanetHandle, std::__1::allocator<CoCoA::Involution::JanetHandle> >::~~__list_imp' requested here
```

```
class _LIBCPP_TYPE_VIS list
^
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:527:
45: note:
```

```
passing argument to parameter '__f' here
```

```
static void __unlink_nodes(__node_base& __f, __node_base& __l) _NOEXCEPT;
                                   ^
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1667:
30: error:
```

```
non-const lvalue reference to type '__node_base' (aka '__list_node_base<value_type, __void_pointer>') cannot bind
```

```
to a value of unrelated type '__node' (aka '__list_node<value_type, __void_pointer>')
```

```
base::__unlink_nodes(const_cast<__node&>(*__f.__ptr_), *__l.__ptr_->__prev_);
               ^~
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1253:
9: note:
```

```
in instantiation of member function 'std::__1::list<CoCoA::Involution::JanetHandle, std::__1::allocator<CoCoA::Involution::JanetHandle> >::erase' requested here
```

```
erase(__i, __e);
^
```

```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:1166:
9: note:
```

```
in instantiation of function template specialization 'std::__1::list<CoCoA::Involution::JanetHandle, std::__1::allocator<CoCoA::Involution::JanetHandle>
```

```
::assign<std::__1::__list_const_iterator<CoCoA::Involution::JanetHandle, void * >' requested here
```

```
assign(__c.begin(), __c.end());
^
```

```

../../include/CoCoA/TmpJBDatastructure.H:435:15: note: in instantiation of member function
  'std::__1::list<CoCoA::Involution::JanetHandle, std::__1::allocator<CoCoA::Involution::JanetHandle> >::operator='
  requested here
    myArm = NextArm;
    ^
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../lib/c++/v1/list:527:
45: note:
    passing argument to parameter '__f' here
    static void __unlink_nodes(__node_base& __f, __node_base& __l) _NOEXCEPT;
    ^
3 errors generated.
make[2]: *** [SparsePolyRing.o] Error 1
===== ***** Compilation failed in CoCoALib source subdirectory AlgebraicCore/ *****
===== make[1]: *** [library] Error 1
make: *** [library] Error 2

```

## #2 - 31 Mar 2016 13:11 - John Abbott

Some notes about Bruns's platform:

we are just preparing Normaliz 3.1.1. On Mac we tried to compile CoCoAKib after the following configuration:

```
./configure --with-cxx=clang++ --with-cxxflags="-std=c++11 -stdlib=libc++ -mmacosx-version-min=10.7"
```

Our CLang is

Apple LLVM version 5.0 (clang-500.2.79) (based on LLVM 3.3svn)

Target: x86\_64-apple-darwin13.3.0

Thread model: posix

## #3 - 31 Mar 2016 13:20 - John Abbott

JAA tried compiling on a recent Linux box (with recent gcc, version 5.3.1) explicitly setting the flag -std=c++11. No compilation errors were found in CoCoALib (but we do need to stop using auto\_ptr); there is a small problem in debug\_new.C, but that is not part of CoCoALib.

## #4 - 03 May 2016 10:34 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Mario has given me an updated version of his code. He has used the `__cplusplus` preprocessor variable to have code which works with both C++03 and C++11. It all works fine on my ancient MacBook; perhaps I'll also check on a couple of Linux boxes. I shall check in shortly.

Hoping to take the new code to Osnabrueck this weekend to check that it works fine there too.

**#5 - 16 Sep 2016 16:19 - John Abbott**

- Target version changed from *CoCoALib-0.99550 spring 2017* to *CoCoALib-0.99560*

**#6 - 20 Feb 2017 13:12 - John Abbott**

@Winfried: could you check whether this problem persists in the latest version of CoCoALib? There is version from 16 Feb on our website; maybe you already have it?

Thanks,  
John.

**#7 - 06 Nov 2017 14:02 - John Abbott**

- Target version changed from *CoCoALib-0.99560* to *CoCoALib-0.99600*

**#8 - 12 Jun 2018 18:35 - John Abbott**

- Target version changed from *CoCoALib-0.99600* to *CoCoALib-1.0*

**#9 - 26 Jun 2018 15:21 - John Abbott**

- Related to Feature #82: *C++11 compatibility questions added*

**#10 - 12 Mar 2020 14:34 - John Abbott**

- Target version changed from *CoCoALib-1.0* to *CoCoALib-0.99800*

What is the status this issue? Is it now fully resolved?  
Anna succeeded (2020-03-11) in compiling on her Mac, presumably using clang.

**#11 - 29 Oct 2020 14:26 - John Abbott**

- Assignee set to *John Abbott*  
- % Done changed from *10* to *40*

Since we are no longer guaranteeing pre-C++11 compatibility, we can remove from Mario's code all #if trickery related to old versions of the compiler.

The only files involved are `TmpJBDatatructure` (both header & impl).  
I have just commented out the relevant parts. All seems to compile fine, and all tests pass. All examples too.

Should I now remove the commented-out code?

**#12 - 30 Oct 2020 15:19 - John Abbott**

- Status changed from *In Progress* to *Feedback*  
- % Done changed from *40* to *90*

Anna says it compiled cleanly now.

JAA: must remove the commented out code.

**#13 - 20 Sep 2021 16:17 - John Abbott**

- *Status changed from Feedback to Closed*
- *% Done changed from 90 to 100*
- *Estimated time set to 3.33 h*

I have removed the commented-out code (and another function which existed only for that code).  
It all compiles, and all tests pass.  
So closing this issue.