

## CoCoALib - Design #1346

### C++14: use the new for loop syntax where appropriate (like cocoa's foreach)

21 Oct 2019 12:26 - John Abbott

<b>Status:</b>	In Progress	<b>Start date:</b>	21 Oct 2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	80%
<b>Category:</b>	Improving	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoALib-0.99880	<b>Spent time:</b>	12.55 hours
<b>Description</b>			
Since C++11 there is a new "range-based" <b>for</b> loop.			
Employ it where appropriate in CoCoALib.			
<b>Related issues:</b>			
Related to CoCoALib - Design #1225: Move to C++14 (skipping C++11)		<b>In Progress</b>	<b>06 Sep 2018</b>
Related to CoCoALib - Design #1242: C++14: Use type auto where appropriate		<b>In Progress</b>	<b>08 Feb 2019</b>
Related to CoCoALib - Feature #1472: Idea for for loops		<b>In Progress</b>	<b>23 Jul 2020</b>
Related to CoCoA-5 - Design #1520: Replace BOOST_FOREACH by new C++14 range f...		<b>Closed</b>	<b>26 Oct 2020</b>

#### History

##### #1 - 21 Oct 2019 12:26 - John Abbott

- Related to Design #1225: Move to C++14 (skipping C++11) added

##### #2 - 21 Oct 2019 12:27 - John Abbott

We should maintain a list of files to change and/or a list of files already changed.

- SparsePolyOps-ideal-monomial.C
- TmpPPVector.C

##### #3 - 21 Oct 2019 12:28 - John Abbott

- Related to Design #1242: C++14: Use type auto where appropriate added

##### #4 - 21 Oct 2019 18:44 - Anna Maria Bigatti

used massively for monomial ideals.  
Tested.  
cvs-ed.

##### #5 - 21 Oct 2019 18:45 - Anna Maria Bigatti

- Status changed from New to In Progress  
- % Done changed from 0 to 10

##### #6 - 13 Jan 2020 17:20 - Anna Maria Bigatti

- Subject changed from C++14: use the new for loop syntax where appropriate to C++14: use the new for loop syntax where appropriate (like cocoa's foreach)

Example: so I know where to find one ;-)

```
for (const auto& f: GB)
  if (!IsSqFree(LPP(f))) return myAssignRadicalFlag(false);
```

#### #7 - 11 Feb 2020 18:31 - Anna Maria Bigatti

- % Done changed from 10 to 20

worked on TmpGReductor: much more readable!!!

#### #8 - 13 Feb 2020 16:21 - Anna Maria Bigatti

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

- % Done changed from 20 to 50

I looked for iterator and changed in a few more files. (can't remember which)  
Not so many left to do.

#### #9 - 17 Feb 2020 18:15 - John Abbott

I have changed some for loops to use the new syntax.

It is slightly worrying that one can change the code so easily without actually understanding what it does :-/  
Anyway, the code still compiles, and all the tests pass...

I prefer to use auto only when the real type is "long and complicated": for example I find it clearer to write

```
for (const RingElem& f: gens(I)) ...
```

rather than

```
for (const auto& f: gens(I)) ...
```

The first form helps the reader (but the second form is easier to write if you do not really know the code).

**NOTE** I do agree that it usually makes the code much more readable :-)

#### #10 - 18 Feb 2020 12:35 - John Abbott

- % Done changed from 50 to 60

I have done the following files:

ApproxPts2.C

ApproxPts.C  
ExternalLibs-Normaliz.C  
GEnv.C  
QuotientRing.C  
RingHom.C  
TmpGReductor.C (ONLY PARTLY?)  
TmpIsTree.C  
TmpJBAAlgorithm.C  
TmpJBDatastructure.C  
TmpJBMill.C  
TmpLDP.C  
TmpPBMill.C  
TmpStabilityAlgorithm.C  
VectorOps.C

**#11 - 25 Sep 2020 11:22 - John Abbott**

- Related to Feature #1472: Idea for for loops added

**#12 - 26 Oct 2020 10:04 - John Abbott**

- Related to Design #1520: Replace BOOST\_FOREACH by new C++14 range for syntax added

**#13 - 04 Feb 2022 21:57 - John Abbott**

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

How should this be done in a systematic way (which can be paused)?  
Postponing since it is just "tidying up".  
We must try to remember to use the new syntax when writing new code!

**#14 - 13 Mar 2023 21:57 - John Abbott**

- % Done changed from 60 to 80

This is a mind-numbing task.  
Something weird in **SparsePolyOps-reduce.C**:  
the obvious transformation of the code seem not to work... no idea why!  
I think more could still be done: there are lost of iterators in Mario Albert's code (phew!)

**#15 - 14 Mar 2023 21:23 - John Abbott**

I have done some more. Anna fixed yesterday's bug.  
Need a break from this... anyway it does seem to make the code more readable :-)

**#16 - 14 Mar 2023 21:23 - John Abbott**

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880

#17 - 22 Apr 2024 21:08 - John Abbott

Some files still need to be updated: I think the "morse" files need to be updated.

The "JB" files look scary to touch!

Maybe TmpStability

I wouldn't do any more...

Any volunteers? **Anna?**