

CoCoALib - Design #1242

C++14: Use type auto where appropriate

08 Feb 2019 21:35 - John Abbott

Status:	In Progress	Start date:	08 Feb 2019
Priority:	Normal	Due date:	
Assignee:		% Done:	40%
Category:	Tidying	Estimated time:	0.00 hour
Target version:	CoCoALib-0.99880	Spent time:	1.40 hour
Description			
When we switch to C+11/C++14 consider replacing some explicit type declarations with auto .			
Related issues:			
Related to CoCoALib - Feature #82: C++11 compatibility questions		Closed	26 Jan 2012
Related to CoCoALib - Design #1225: Move to C++14 (skipping C++11)		In Progress	06 Sep 2018
Related to CoCoALib - Design #1346: C++14: use the new for loop syntax where ...		In Progress	21 Oct 2019

History

#1 - 08 Feb 2019 21:36 - John Abbott

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

#2 - 18 Oct 2019 11:15 - John Abbott

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

#3 - 18 Oct 2019 11:19 - John Abbott

I must re-read the relevant parts of Meyers's book (Eff. Modern C++). Where should we use **auto**?

- use **auto** instead of the explicit type for C++ iterators (usu. the explicit type is "unreadable"); but see also Meyers's notes about the new **for** loop syntax
- I suggest **not** using **auto** if the explicit type is "simple and easy to read" (e.g. I think I would not use it for `vector<RingElem>`).
- overall guideline: we should use **auto** to help make the code *easier to read and understand*

#4 - 18 Oct 2019 11:23 - John Abbott

It may be helpful to keep a list of source files which have been "done" (this may also include that the file was looked at, but in the end we decided not to use auto anywhere)

- SparsePolyOps-ideal-ZeroDim.C (for const iterator in for loops)
- SparsePolyOps-ideal-monomial.C (for const iterator in for loops)

NOTE: I expect that we would use auto only very rarely in header files...

#5 - 18 Oct 2019 17:04 - Anna Maria Bigatti

- Status changed from New to In Progress

- % Done changed from 0 to 10

#6 - 21 Oct 2019 12:06 - Anna Maria Bigatti

John Abbott wrote:

- use **auto** instead of the explicit type for C++ iterators (usu. the explicit type is "unreadable"); but see also Meyers's notes about the new **for** loop syntax

I have tried the new **for** loop syntax in SparsePolyOps-ideal-monomial.C.
It's like a dream come true :-)

#7 - 21 Oct 2019 12:28 - John Abbott

- Related to Design #1346: C++14: use the new for loop syntax where appropriate (like cocoa's foreach) added

#8 - 21 Oct 2019 12:29 - John Abbott

I have created a new issue ([#1346](#)) about using the new **for** loop syntax; please put relevant progress reports there :-)

#9 - 08 Jan 2020 22:55 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

#10 - 06 Oct 2020 15:39 - John Abbott

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

I suggest we work through the source file systematically, and note which ones we have updated to use **auto** (and which are still to do).
This will be a long task :-/

Source files which have already been done:

- **bool3.C**, **ApproxPts.C**, **ApproxPts2.C**, **assert.C**, **BigInt.C** *nothing to do*
- **apply.C** *nothing to do* (see also [#1467](#))

#11 - 08 Mar 2023 19:55 - John Abbott

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

#12 - 22 Apr 2024 20:43 - John Abbott

- DynamicBitset uses iterators explicitly; maybe switch to auto?
- possibly review TmpJB... code too?
- LongRange but see also std::ranges in C++20

#13 - 22 Apr 2024 21:08 - John Abbott

- % Done changed from 10 to 40