CoCoALib - Design #581

C++14: MachineInt

04 Jul 2014 14:36 - John Abbott

Status:	Closed	**Start date:**	04 Jul 2014
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Portability	**Estimated time:**	4.70 hours
Target version:	CoCoALib-0.99800	**Spent time:**	4.65 hours

Description

We must modify MachineInt when switching to C++14 because there is a new integral type: (unsigned) long long

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 #925: MachineInt or long for args which are indi...	**In Progress**	**20 Sep 2016**
Related to CoCoALib - Design #934: MachineInt: change semantics?	**In Progress**	**30 Sep 2016**
Related to CoCoALib - Support #1666: MachineInt: chase through ULL changes	**In Progress**	**16 Feb 2022**

History

#1 - 04 Jul 2014 14:40 - John Abbott

It might be possible to use templates for the ctors of MachineInt together with a mechanism for ensuring that the actual arg type is integral.

See enable_if (part of C++11, also in BOOST).

Also see http://stackoverflow.com/questions/12073689/c11-template-function-specialization-for-integer-types (the solution by mitchnull).
Also see http://stackoverflow.com/questions/14676574/differences-between-stdis-integer-and-stdis-integral
These last two may allow the same result to be achieved in C++03... not sure though!

#2 - 04 Jul 2014 14:51 - John Abbott

- Estimated time set to 5.00 h

#3 - 05 Apr 2019 16:06 - John Abbott

- Subject changed from C++11: MachineInt to C++14: MachineInt

- Description updated

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99700

I have modified the subject and description to refer to C++14 instead of C++11.

There is a summary of the integral types available in C++14 at this link:
https://en.cppreference.com/w/cpp/types/integer

Presumably the type intmax_t should be used for the internal representation.

#4 - 05 Apr 2019 16:06 - John Abbott

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

#5 - 05 Apr 2019 16:06 - John Abbott

- Related to Design #925: MachineInt or long for args which are indices (yet again) added

#6 - 05 Apr 2019 16:06 - John Abbott

- Related to Design #934: MachineInt: change semantics? added

#7 - 08 Jan 2020 22:54 - John Abbott

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

#8 - 06 Oct 2020 15:42 - John Abbott

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

#9 - 15 Feb 2022 21:43 - John Abbott

This is not as straightforward as I was hoping.
It is not hard to change the internal repr (to long long int or even intmax_t).
What is harder is deciding what to do about IsSignedLong, AsSignedLong *etc.*
I think the GMP interface takes long for machine integers, and there is no interface for long long.
Of course, there should be no problem if long and long long are the same size...

Ummmm!

#10 - 16 Feb 2022 19:37 - John Abbott

- Status changed from New to Resolved

- Assignee set to John Abbott

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

- % Done changed from 0 to 80

I have modified the impl to use unsigned long long int as the internal repr;
there is also a second impl which uses signed long long int, but that it is not normally used -- have not yet tested it.

Anyway the ULL version seems to work. The main problem will be if ULL and UL are different,
and a caller supplies a value which does not fit into UL. This will likely trigger a run-time
error (ArgTooBig). I'll make a new issue about chasing through all the changes.

#11 - 16 Feb 2022 19:39 - John Abbott

- Related to Support #1666: MachineInt: chase through ULL changes added

#12 - 18 Feb 2022 15:26 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

- Estimated time changed from 5.00 h to 4.70 h

I have fixed some minor bugs -- Bruns compiled on some weird platform which highlighted them.
Checked in. Hoping it is all OK now.