CoCoALib - Bug #783

abs for MachineInt

09 Oct 2015 10:46 - John Abbott

Status:	Closed	**Start date:**	09 Oct 2015
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Various	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99540 Feb 2016	**Spent time:**	2.50 hours

Description

There is a fn called abs for MachineInt; it works fine (and returns an unsigned long).

The problem is that inside the CoCoA namespace abs(MachineInt) hides the standard library functions called abs defined in cstdlib and/or cmath (and which return a signed value).

The standard library functions are surely faster than abs(MachineInt).

Arrange for the standard library fns abs to be preferred over abs(MachineInt).

History

#1 - 09 Oct 2015 10:50 - John Abbott

I found the problem when trying to investigate why test-NumTheory1.C produced warnings (about comparing signed and unsigned values) during compilation.

The code in test-NumTheory1.C looked to be perfectly correct; I also checked carefully the C++ standard definition, and whether there were reports of an old compiler like mine erronesously returning the wrong type of result.

This comes under the category (mild) "nasty surprise", so needs to be fixed!

#2 - 09 Oct 2015 10:52 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 10

I think there are two possible approaches:

1. rename abs(MachineInt)
2. ensure that the standard library abs fns are just as visible as abs(MachineInt)

Approach **(1)** is clearly OK, and I know how to achieve it.
Approach **(2)** should be OK, and can perhaps be achieved by suitable using directives.

#3 - 09 Oct 2015 11:31 - John Abbott

- % Done changed from 10 to 20

I have observed to uses for abs(MachineInt):

- to obtain the absolute value (*e.g.* in some division/remainder functions)
- to obtain the negation of a value known to be negative

I am now considering the following:

- create a new function negate(MachineInt) which asserts that its arg is negative
- rename abs(MachineInt) to uabs

The name abs is possibly misleading as the standard C++ library says it returns a signed value, whereas the function for MachineInt has to return an unsigned value (so a different name is desirable).

#4 - 09 Oct 2015 12:23 - John Abbott

- % Done changed from 20 to 50

I have implemented the suggestion in comment 3 above. Also changed all files which need to be changed; now running a final check that all is well. Hope to check in this evening.

The function negate is strictly redundant since uabs gives the same result on all valid inputs to negate, but I think that negate better expresses the programmers intentions.

#5 - 09 Oct 2015 13:49 - John Abbott

The problem of ::std::abs being hidden (when inside namespace CoCoA) persists even after abs(MachineInt) has been renamed to uabs. The point is that there are other functions inside namespace CoCoA which are called abs (*e.g.* for BigInt, and BigRat, and even RingElem); these continue to hide std::abs from view.

There are two easy solutions: write std::abs at each call, or write using std::abs at the start of the scope in which you wish to call it.

It might also be possible to put an explicit using std::abs inside the CoCoA namespace in files BigInt.H, BigRat.H and RingElem.H, **but** guidelines for C++ programmers discourage the use of using inside header files.

#6 - 09 Oct 2015 20:20 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 50 to 90

Checked in several files: new MachineInt.H and several consequential changes.

I got the redmine reference wrong when checking in: will try to fix now :-(
NOTE corrected redmine references.

#7 - 23 Mar 2016 15:27 - Anna Maria Bigatti

- Status changed from Feedback to Closed

- % Done changed from 90 to 100