

CoCoALib - Bug #858

floor for TwinFloat can produce ERR::SERIOUS

26 Mar 2016 11:03 - John Abbott

Status:	Closed	Start date:	26 Mar 2016
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Maths Bugs	Estimated time:	3.90 hours
Target version:	CoCoALib-0.99550 spring 2017	Spent time:	3.90 hours
Description			
Fix impl of RingTwinFloatImpl::myFloor so that the following does not happen			
<pre>ring RR = NewRingTwinFloat(64); RingElem x = one(RR); RingElem smaller = x - power(BigRat(1,2),145); // 1- 2^(-145) BigInt N = floor(smaller); // throws ERR::SERIOUS</pre>			
Something similar will be needed for myCeil, and perhaps also for myNearestInt.			
Related issues:			
Related to CoCoALib - Bug #853: NearestInt can needlessly throw InsufficientP...		Closed	23 Mar 2016
Related to CoCoALib - Design #859: Twin-float: comparisons and equality test		Closed	28 Mar 2016
Related to CoCoALib - Bug #860: Check impl of RingTwinFloatImpl::myIsRational		Closed	30 Mar 2016

History

#1 - 26 Mar 2016 11:03 - John Abbott

- Related to Bug #853: NearestInt can needlessly throw InsufficientPrecision added

#2 - 26 Mar 2016 11:06 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 10

The point is that the candidate value for floor should really be floor of primary component plus a very small epsilon. Perhaps look at the impl of myIsInteger to see how to proceed.

It's quite hard getting the details right.

What actually failed was a self test inside myFloor; maybe this should be an assertion?

#3 - 26 Mar 2016 11:28 - John Abbott

One way out of the problem would simply be to reinstate the test:

```
BigInt N;
if (myIsInteger(N, rawx)) return N;
```

This does seem to be a slightly crude solution, but there may be nothing better. I'll investigate.

#4 - 26 Mar 2016 21:21 - John Abbott

- % Done changed from 10 to 20

I am now convinced that myFloor must effectively make a myInteger test.

If myInteger succeeds then its result is surely the result of myFloor.

If myInteger says "no" then the value is "far from" an integer, and taking the floor of the primary component gives the only possible candidate.

If myInteger throws InsuffPrec then the value is very close to an integer, i.e. the implicit (open) interval contains an integer, thus there are two possible values for the floor, and I see no justification for choosing between them (i.e. throwing InsuffPrec is the correct behaviour).

#5 - 26 Mar 2016 21:23 - John Abbott

An ad hoc test behaved as I expected, so I think myFloor (and myCeil are now OK).

Wish I could say the same for myNearestInt... test-OrderedDomain2 is failing again :(

#6 - 26 Mar 2016 22:54 - John Abbott

- % Done changed from 20 to 50

I found a bug in myInteger, so have written a completely new version (it's quite a nightmare debugging code written directly with mpf_XYZ functions).

Previously it called myIntegerRational and simply checked whether the denom was 1, but this threw InsuffPrec when it should not have done. Now myInteger is separate code.

I have a nasty feeling that myIntegerRational may still sometimes throw InsuffPrec when it should not... someone else can check that!!

#7 - 28 Mar 2016 21:19 - John Abbott

- Related to Design #859: Twin-float: comparisons and equality test added

#8 - 30 Mar 2016 18:15 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 50 to 90

I am now happy with the impls; they are simple enough that they are "obviously correct" (I hope!). Also I believe that the impls are reasonably efficient.

Putting issue into feedback.

In principle, it is enough to implement just one of myFloor and myCeil; e.g. myCeil could just be if (integer) return value; else return 1+floor. I'm not convinced that the loss of readability is sufficiently compensated by the gain in "slickness".

#9 - 30 Mar 2016 18:34 - John Abbott

- Related to Bug #860: Check impl of RingTwinFloatImpl::myIsRational added

#10 - 25 Jun 2016 12:08 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 3.90 h

I have added a new test (test-bug7.C) for this issue.

Since there have been no problems for 3 months, I am closing the issue.