CoCoALib - Feature #565

FloatApprox for TwinFloat values?

22 May 2014 22:20 - John Abbott

Status:	In Progress	**Start date:**	22 May 2014
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	20%
Category:	New Function	**Estimated time:**	20.00 hours
Target version:	CoCoALib-1.0	**Spent time:**	1.35 hour

Description

Does it make sense to have a fn like FloatApprox for twin-float values?

Currently the only way to get a "real value" out of a twin-float is to apply IsRational (or IsInteger); this is very limited.

Related issues:

Related to CoCoA-5 - Slug #907: ApproxSolve very slow on this example	**Closed**	**14 Jul 2016**

History

#1 - 14 Jul 2014 14:39 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 20

- Estimated time changed from 40.00 h to 20.00 h

After further thought it seemed to make more sense to output the twin-float value as a MantExp2 structure, which better allows the actual accuracy to be represented. It is then simple to convert the MantExp2 structure into a rational if so desired.

I have a first prototype impl; it worked on a couple of trivial tests. Needs a lot more work though.

#2 - 19 Jul 2016 14:26 - John Abbott

- Related to Slug #907: ApproxSolve very slow on this example added

#3 - 19 Jul 2016 14:30 - John Abbott

My argument that creating a MantExp2 structure makes evident the precision is correct, but there are several occasions when the precision is of secondary importance (after all RingTwinFloat guarantees that it is better than a certain bound), and it would be helpful to have a function which produces directly a BigRat.

I am undecided exactly what the fn should be called. I observe that there is a fn called SimplestBinaryRatBetween, so the name should probably contain BinaryRat.

#4 - 19 Jul 2016 22:44 - John Abbott

How about the name BinaryRatApprox?
Mmm, well, that is exactly what FloatApprox does in other cases, but it's called FloatApprox.