CoCoA-5 - Feature #980

CoeffSize: function to measure the size of coeffs in a poly

24 Nov 2016 18:38 - John Abbott

Status:	In Progress	**Start date:**	24 Nov 2016
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	20%
Category:	CoCoA-5 function: new	**Estimated time:**	3.00 hours
Target version:	CoCoA-5.?.?	**Spent time:**	1.85 hour

Description

In email Anna proposed a function, tentatively named CoeffSize, to measure the size of the coeffs in a poly (over QQ).

Discuss.

History

#1 - 24 Nov 2016 18:51 - John Abbott

The definition originally proposed in email was to return a pair of integers being effectively [max(FloorLog10(num(c))) | c in coeffs] and [max(FloorLog10(den(c))) | c in coeffs].

JAA wrote that he is perplexed by this defn, and proposed two alternatives:

- determine the least common denom of the coeffs, return log of the commondenom and the max of log of numerators (after clearing the denom)
- return sum of logs of numerators and sum of logs of denominators (roughly the print-size of the poly)

Anna wrote saying that the idea was to estimate the modulus needed so that (fault-tolerant) rational recovery would work. In this case the result should probably be a single integer being max([FloorLog10(num(c)) + FloorLog10(den(c)) | c in coeffs]. Maybe it would be better to find the max of FloorLog10(num(c)*den(c))?

If the rational recovery tries to be clever by keeping track of common-denom-so-far then CoeffSize needs to be a little more sophisticated (keep track itself of the common-denom-so-far, then return max of log(ScaledNum*CommonDen).

#2 - 24 Nov 2016 19:01 - John Abbott

Anna has put a first prototype in experimental.cpkg5; it follows the original defn. It is deliberately not documented.

JAA thinks that if we find a function useful then it should be made public because it will probably be useful (eventually) to someone else.

Regarding fault-tolerant rational recovery, JAA thinks the most appropriate size measure for a rational is something like 0.3*(3.5+FloorLog2(num(q)*den(q))+FloorSqrt(FloorLog2(num(q)*den(q)))) if the "continued fraction method" is used.
Probably 4+FloorLog10(num(q)*den(q)) gives a fair approximation in most cases.

#3 - 24 Nov 2016 19:12 - Anna Maria Bigatti

- % Done changed from 0 to 20

- Estimated time set to 3.00 h

Maybe the one I wrote could then be called NumDenSize10?
I still think it is useful, for a human, to know that.

#4 - 28 Nov 2023 22:29 - John Abbott

- Status changed from New to In Progress

JAA notes that FloorLog2 is faster than FloorLog10 -- a quick check suggest at least 100 times faster.
Also FloorLog2 gives a finer measure of size, but it is less immediate to comprehend in terms of "decimal digits" -- you just need to multiply by 0.30103 (approx)

At the moment I am unsure when such a function would actually be used. It has been in experimental.cpkg5 for about 7 years, but I'd never really noticed it.