

CoCoA-5 - Bug #275

Unhelpful error messages when SmallExponent_t is unsigned char.

15 Nov 2012 18:46 - John Abbott

Status:	Closed	Start date:	15 Nov 2012
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	3.80 hours
Target version:	CoCoA-5.2.4	Spent time:	3.50 hours
Description			
I changed config.H so that the type SmallExponent_t is unsigned char (in an attempt to tackle Enrico Carlini's computation). Two CoCoA-5 tests failed: whatiscocoa.cocoa5 and SourceAnna.cocoa5. In both cases the error produced is Unable to convert value. Investigate why these examples need exponents bigger than 255.			
Related issues:			
Related to CoCoALib - Design #268: Exponent range (in power products)		Closed	18 Oct 2012
Related to CoCoALib - Feature #269: PPMonoids: check for exponent overflow in...		Closed	18 Oct 2012
Related to CoCoALib - Design #683: Module index component in internal compres...		Closed	14 Apr 2015

History

#1 - 02 Apr 2014 17:34 - Anna Maria Bigatti

- Target version set to CoCoA-5.1.0 Easter14

#2 - 04 Apr 2014 14:31 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.?.?

#3 - 02 Sep 2014 14:59 - John Abbott

- Category set to enhancing/improving

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 50

I have just retried the experiment. Several CoCoALib tests fail:

```
***** The following CoCoALib tests failed, please tell us about it.
***** test-GOperations1 test-GOperations2 test-QuotientRing1 test-SparsePolyRing1 test-SqFreeFactor1 test-Tm
pMorseGraph
```

In each case the error produced was CoCoA::ERR::ExpTooBig which I think is quite clear.

I compiled CoCoA-5 anyway and 3 tests failed:

```
***** The following CoCoA-5 tests failed, please tell us about it.
***** whatiscocoa.cocoa5 radical.cocoa5 SourceAnna.cocoa5
```

Again all the errors were "Exponent too big" (or a direct consequence of such an error). These messages are quite satisfactory.

Is it OK for the tests to require such high exponents? I find it a little surprising that so many tests require exponents over 255... perhaps we should take a look?

Addendum also ex-module fails with ExpTooBig, but I find that hard to believe given how simple it is -- this should be checked!!

#4 - 02 Sep 2014 16:01 - John Abbott

I have just (20140902) checked using typedef unsigned short SmallExponent_t; and everything worked fine.

Reminder: we should still investigate why so many tests (& 1 example) fail with unsigned char! **DONE** see comment 6 below!

#5 - 13 Apr 2015 15:13 - John Abbott

I have again encountered the problems (failing tests) with using unsigned char for exponents. I wanted to use very compact exponents because I wanted to compute with polynomials with >10000000 terms in many variables with low degrees (< 10), so it made sense to put exponents in unsigned char (the most compact representation CoCoALib offers).

After a little debugging I found that the problem was an exponent of 10000 in one example; and this high degree came from the "trick" used to represent module elements as polynomials with an extra "module component indet". The module was used to compute GCD between two "multivariate" polys over a finite field.

Hint: GBEnv.H:141 definition of compile time constant myMaxComponentIndex

#6 - 13 Apr 2015 15:43 - John Abbott

I have just tried all tests with SmallExponent_t set to unsigned char, and myMaxComponentIndex set to 100 (instead of the default 10000). All CoCoALib tests passed. Just 1 CoCoA-5 test failed: radical -- not sure why. All CoCoALib examples ran fine too (except ex-SmallFp1, but that is an unrelated problem).

Note radical.cocoa5 produces an answer with generators of degree 30; according to the debugger at some point it did try to multiply together two polynomials with high degrees (23 and 240ish, if I recall correctly)

#7 - 11 May 2015 14:29 - John Abbott

- Target version changed from CoCoA-5.?.? to CoCoA-5.1.3/4 Jan 2016

#8 - 16 Feb 2016 16:24 - John Abbott

- Target version changed from CoCoA-5.1.3/4 Jan 2016 to CoCoA-5.?.?

#9 - 16 Jan 2018 18:21 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 50 to 90

- Estimated time set to 2.90 h

After changing some "ugly" code for finding which indets are actually in a poly (in radical.cpkg5 and in BringIn.cpkg5), all CoCoA-5 tests run fine. All CoCoALib tests run fine. All CoCoALib examples run fine.

Just one strange observation: I thought powering did a quick check for exp overflow, but it seems not if the power is 2

```
J := x^128;
J^2; --> wrong answer with no warning!
J := x^86;
J^3; --> error about exponent overflow
```

UPDATE fixed the problem with squaring (actually there were two bugs)

#10 - 17 Jan 2018 11:44 - John Abbott

- *Status changed from Feedback to Closed*
- *Target version changed from CoCoA-5.?.? to CoCoA-5.2.4*
- *% Done changed from 90 to 100*
- *Estimated time changed from 2.90 h to 3.80 h*

I think I have solved all problems