CoCoA-5 - Slug #1390

CoCoA-5 test: radical too slow

09 Jan 2020 20:46 - John Abbott

Status:	Closed	**Start date:**	09 Jan 2020
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	2.11 hours
Target version:	CoCoA-5.3.0	**Spent time:**	2.10 hours

Description

The CoCoA-5 test radical.cocoa5 is too slow: it takes longer than all the other CoCoA-5 tests combined.

The culprit seems to be the last example (over ZZ/(32003) which takes almost 90% of the time.

Can we make this test faster?

Related issues:

Related to CoCoALib - Feature #1780: radical for ideals in SparsePolyRing: c...	**Closed**	**06 Feb 2024**

History

#1 - 09 Jan 2020 20:49 - John Abbott

On my computer with debugging the test radical.cocoa5 takes more than 60s. It is just about tolerably fast when I compile without debugging.

Is the last example in the test file really so important as to justify taking as much time as all the other tests combined?

NOTE If we resolve this, I should probably remove the ulimit call in the RunTests.sh script.

#2 - 10 Jan 2020 15:56 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 20

The radical.cocoa5 test took 160s (2 min 40s) on the netbook.

I propose simply commenting out that last slow case in the file.
When eventually a new impl of radical becomes available we can perhaps reactivate it.

#3 - 10 Jan 2020 17:08 - John Abbott

- Assignee set to John Abbott

- % Done changed from 20 to 50

- Estimated time set to 1.33 h

I have disabled the slow test, and checked in.

Now all CoCoA-5 tests complete in 12s instead of 23s

#4 - 13 Jan 2020 12:16 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

- % Done changed from 50 to 80

I separated the file into two files
radical0dim.cocoa5 and radical.cocoa5.
Even though the function is the same the underlying algorithms are drastically different.

This split will make it easier to test them separately.

Moreover, there was another slow test which was already commented out.

```
M:=6;   use R::=QQ[x[1..M,1..M]], DegLex;
```

Now it's fast, so I activated it.

I made it easier to activate the slow test when one wants to actually check it.

#5 - 19 Jan 2020 10:47 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 80 to 90

- Estimated time changed from 1.33 h to 2.11 h

I have removed ulimit from RunTests.sh.

All tests pass, even with debugging active.
Now all CoCoA-5 tests together take 66s with debugging active.
Without debugging active the tests take about 13s.

#6 - 04 Mar 2020 16:22 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

#7 - 06 Feb 2024 11:44 - Anna Maria Bigatti

- Related to Feature #1780: radical for ideals in SparsePolyRing: code in C++ added