CoCoALib - Feature #1103

Pseudo-zero-dim ideals

19 Sep 2017 14:34 - John Abbott

Status:	In Progress	**Start date:**	19 Sep 2017
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	30%
Category:	Improving	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	1.00 hour

Description

We have several functions which work well for 0-dim ideals but which give "Not yet implemented" for non-0-dim ideals.

However, the current implementations would work fine in the "obvious subring":

```
use Q[x,y,z];
IsPrimary(ideal(x^2,y));
--> ERR::NYI
```

Of course, CoCoA will compute the answer if the ideal is mapped into QQ[x,y]...
so why not do this automatically?

Related issues:

Related to CoCoALib - Feature #658: Indets actually in a poly (or vector or m...	**Closed**	**22 Jan 2015**
Related to CoCoALib - Feature #1254: New function: MinPoly for pseudo-zero-di...	**New**	**06 Mar 2019**
Related to CoCoALib - Design #849: Cleanup MinPoly code	**In Progress**	**22 Mar 2016**
Related to CoCoALib - Slug #1569: IsInRadical too slow (test-RadicalMembership)	**Closed**	**22 Jan 2021**

History

#1 - 19 Sep 2017 23:12 - John Abbott

Here is an example where it would be faster to recognize that the ideal is 0-dim in an "obvious subring":

```
use QQ[x,y,z];
I := ideal(x^5 +x^2*z^3 -y^2*z^2,  x*y^4 -z^5 +x*y^2*z,  y^5 +x^2*z^3 -z^4);
t0:=CpuTime();
rI := radical(I);
TimeFrom(t0);
---> 68.255

-- Now adjoin redundant indet
use P ::= QQ[x,y,z,t];
I := ideal(x^5 +x^2*z^3 -y^2*z^2,  x*y^4 -z^5 +x*y^2*z,  y^5 +x^2*z^3 -z^4);
t0:=CpuTime();
rI := radical(I);
TimeFrom(t0);
-->  don't know how much time -- it takes *AGES*
```

#2 - 16 Jan 2018 18:17 - John Abbott

- Related to Feature #658: Indets actually in a poly (or vector or matrix) added

#3 - 05 Mar 2019 15:54 - John Abbott

- Related to Bug #1248: MinPolyQuot: guaranteed and ideal without GBasis added

#4 - 05 Mar 2019 15:57 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Note also that if we map into a temporary ring without redundant indets, then we can also choose a convenient term-ordering (*e.g.* DegRevLex if we need to compute a GBasis)

#5 - 06 Mar 2019 15:10 - Anna Maria Bigatti

- Related to Feature #1254: New function: MinPoly for pseudo-zero-dimensional ideals added

#6 - 06 Mar 2019 16:46 - Anna Maria Bigatti

- Related to deleted (Bug #1248: MinPolyQuot: guaranteed and ideal without GBasis)

#7 - 05 Nov 2021 16:07 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

There is something very similar for RationalSolve in CoCoA-5.
Anna: consider that design and copy:

```
ListX := FindIndets(PolyList); // will be non-empty since we've checked PolyList
P := RingOf(PolyList[1]);
SurplusIndets := diff(indets(P), ListX);
J := ideal(PolyList)+ideal(P,SurplusIndets);
```

#8 - 05 Nov 2021 16:08 - Anna Maria Bigatti

- % Done changed from 10 to 30

#9 - 05 Nov 2021 16:09 - Anna Maria Bigatti

see also #1254

#10 - 16 Feb 2024 18:23 - John Abbott

- Related to Design #849: Cleanup MinPoly code added

#11 - 16 Feb 2024 20:46 - John Abbott

- Related to Slug #1569: IsInRadical too slow (test-RadicalMembership) added