CoCoALib - Feature #1227

exgcd; solve Bezout equation

19 Sep 2018 14:13 - John Abbott

Status:	New	**Start date:**	19 Sep 2018
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	1.70 hour

Description

I'd like to use an **exgcd** function: one which calculates both the GCD and the corresponding linear combination of the 2 args.

What exactly should this function do? Consider the following session:

```
use P ::= QQ[x,y,z];
f := x+1;
g := x;
exgcd(f,g);
-- what should happen here?
```

Note that P is not a PID (nor a "Bezout domain"); so in general the Bezout equation has no solution. But with the specific input a solution does exist. Should the call exgcd(f,g) produce a "not PID" error, or should it actually compute the answer?

Related issues:

Related to CoCoALib - Feature #1306: exgcd over integers (ZZ)	**In Progress**	**02 Sep 2019**

History

#1 - 19 Sep 2018 14:23 - John Abbott

- Description updated

- Category set to New Function

- Target version set to CoCoALib-0.99650 November 2019

According to Wikipedia there are Bezout domains which are not PIDs (but it seems all such non PIDs are "funny rings" which CoCoA does not handle).

Several functions in CoCoA which work only for univariate polys, still work in a multivariate ring, so long as the supplied arg is actually univariate. This is very handy for the user (rather than having to map a univariate poly ring, compute the answer there, and then map it back). It would similarly be handy if exgcd works this way.

Cases where exgcd easily/clearly works:

1. ZZ
2. univariate polys (actual args) over field
3. bivariate homogeneous polys (is it worth handling this case?)

Case (3) is a just a special case of the inputs being images of two univariate polys under some homomorphism. How to recognise this?

#2 - 19 Sep 2018 15:25 - John Abbott

- Description updated

#3 - 21 Sep 2018 16:17 - John Abbott

Maybe restricting the problem to just 2 args is too limiting?

Consider the following 3 multivariate polynomials: f1=x, f2=y, f3=x+y+1.
Clearly gcd(f1,f2,f2)=1 and we can express the gcd as a linear combination 1 = -f1-f2+f3.

Yet if we take any 2 of the polys we cannot write the gcd as a linear combination of them.

In general, one can compute a ReducedGBasis; if the result is a single poly then it is clearly the GCD, and we can
get a repr as a linear combination of the generators. It'd be nice to get a **simple linear combination** (not entirely sure what this means).

#4 - 18 Mar 2019 14:10 - John Abbott

- Description updated

- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-1.0

I prefer to defer this question until later, until when we have decided what exactly exgcd should do (and whether it would actually be useful to anyone).

#5 - 29 Aug 2019 14:55 - Florian Walsh

An exgcd function would be very useful.
It is used in computing the Hermite normal form (#1304) and in computing Groebner basis over the integer (#1272).

#6 - 02 Sep 2019 16:08 - John Abbott

- Related to Feature #1306: exgcd over integers (ZZ) added