

## CoCoALib - Design #1653

### Discriminant: return value

27 Jan 2022 14:54 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	27 Jan 2022
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Various	<b>Estimated time:</b>	3.50 hours
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	3.50 hours
<b>Description</b>			
To which ring should the return value of Discriminant belong?			
<pre>/**/ use QQ[x,y]; /**/ f := y^2-2; /**/ discriminant(f); 8 /**/ RingOf(It); --&gt; ???</pre>			
<b>Related issues:</b>			
Related to CoCoALib - Bug #1656: SEGV in resultant		<b>Closed</b>	<b>29 Jan 2022</b>

### History

#### #1 - 27 Jan 2022 15:01 - John Abbott

discriminant is actually two functions: univariate inputs, and multivariate input with specified indet. The multivariate version has to return a result in the original polynomial ring. The expectation for the univariate version is that the discriminant is an element of the coeff ring, BUT at the moment CoCoA returns a result in the poly ring.

We could get an "inconsistency" by supplying a univariate input **and** specifying its indet:

```
/**/ use QQ[x,y];
/**/ f := y^2-2;
/**/ discriminant(f);
8
/**/ discriminant(f,y);
8
```

I wanted to find a polynomial whose discriminant is divisible by 30, but my program did not work as expected:

```
for-loop
  f := some-univariate-poly in QQ[x]
  d := discriminant(f);
  if gcd(d,30) = 30 then println f; endif;
endfor;
```

This code never printed anything... because the gcd was computed in QQ[x], so always came out to 1. If the discriminant has been in the coeff ring (QQ) then an error would have been signalled (since QQ is not a "true gcd domain"). I wasted some time tracking down the problem.

**#2 - 28 Jan 2022 12:56 - John Abbott**

- Status changed from New to In Progress
- % Done changed from 0 to 10

Anna agrees with what I'd like to do.  
Should something similar be done for resultant too?

**#3 - 28 Jan 2022 17:30 - John Abbott**

- Assignee set to John Abbott
- % Done changed from 10 to 20

Since resultant is closely related to discriminant they should have compatible behaviour.  
This includes presumably adding a 2-arg version of resultant which requires that its args are univariate.

Strangely, resultant is built-in while discriminant is not. This is inconsistent, and should be remedied!

**#4 - 29 Jan 2022 15:22 - John Abbott**

- Related to Bug #1656: SEGV in resultant added

**#5 - 29 Jan 2022 17:06 - John Abbott**

- Status changed from In Progress to Resolved
- % Done changed from 20 to 80

There are now 2 resultant functions and 2 discriminant functions(in each case with or without specified indet).  
If no indet is specified the inputs must be univariate, and the result is in the coeff ring!

Must still update doc, and add some tests.

Still slightly unsure about some edge cases: e.g. resultant gives error if either poly is zero, and discriminant gives error if the poly is constant.

**#6 - 31 Jan 2022 17:52 - John Abbott**

The old impl is still in coclib.cpkg5 but renamed to discriminant\_old. I'll delete it soon (after some more testing).

**#7 - 04 Feb 2022 21:52 - John Abbott**

- Status changed from Resolved to Closed
- % Done changed from 80 to 100
- Estimated time set to 3.50 h