

Polinomok legnagyobb közös osztója

October 17, 2023

1 Polinomok maradékos osztása

Az alábbi számolások a SageMath segítségével történtek, ami egy Python alapú komputeralgebra rendszer. Azt vagy lokálisan (terminálban vagy Jupyter nootbookban) vagy online a <https://cocalc.com/features/sage>

A következő parancsal definiálhatjuk polinomok halmazát. Itt x a változó, $GF(5) = \mathbb{Z}_5$ az együtthatók halmazát jelöli. Általában használhatjuk a $GF(p)$ jelölést a \mathbb{Z}_p (p prímszám) halmazára, ill. QQ , \mathbb{R} , \mathbb{C} a racionális, valós ill. a komplex együtthatókra.

```
[1]: R.<x> = PolynomialRing(GF(5))
```

Az alpműveletek a szokásos szabály szerint működnek:

```
[2]: (x^2+x+2)+(x^3+4*x+4)
```

```
[2]: x^3 + x^2 + 1
```

```
[3]: (x^2+x+2)*(x^3+4*x+4)
```

```
[3]: x^5 + x^4 + x^3 + 3*x^2 + 2*x + 3
```

```
[4]: (x^2+x+2)-(x^3+4*x+4)
```

```
[4]: 4*x^3 + x^2 + 2*x + 3
```

Maradékos osztásra a következő parancsot használhatjuk. Itt az első kimenet a q hányados, a második az r maradék.

```
[5]: f = x^2+1
     g = x+1
```

```
[6]: q,r = f.quo_rem(g); (q,r)
```

```
[6]: (x + 4, 2)
```

```
[7]: q*g+r
```

```
[7]: x^2 + 1
```

```
[8]: f,g = x^3+4*x^2+x+1, 3*x^2+1
      q,r = f.quo_rem(g); (q,r)
```

```
[8]: (2*x + 3, 4*x + 3)
```

```
[9]: q*g+r
```

```
[9]: x^3 + 4*x^2 + x + 1
```

További példák

```
[10]: f, g = x^3+4*x^2+x+1, 3*x^2+1
      q,r = f.quo_rem(g)
      print("f =",f," g =",g)
      print("q =",q," r =",r)
      print("g*q+r =", g*q+r)
```

```
f = x^3 + 4*x^2 + x + 1 , g = 3*x^2 + 1
q = 2*x + 3 , r = 4*x + 3
g*q+r = x^3 + 4*x^2 + x + 1
```

```
[11]: f, g = x^4+x^3+x^2+2*x+1, x^4+x^3+4*x^2+1
      q,r = f.quo_rem(g)
      print("f =",f," g =",g)
      print("q =",q," r =",r)
      print("g*q+r =", g*q+r)
```

```
f = x^4 + x^3 + x^2 + 2*x + 1 , g = x^4 + x^3 + 4*x^2 + 1
q = 1 , r = 2*x^2 + 2*x
g*q+r = x^4 + x^3 + x^2 + 2*x + 1
```

2 Euklideszi algoritmus legnagyobb közös osztó kiszámolására

Adott f, g polinomokra végezzük el a követjező maradékos osztásokat:

$$\begin{array}{ll}
 f = q_1 g + r_1 & \deg r_1 < \deg g \\
 g = q_2 r_1 + r_2 & \deg r_2 < \deg r_1 \\
 r_1 = q_3 r_2 + r_3 & \deg r_3 < \deg r_2 \\
 \vdots & \\
 r_{\ell-2} = q_{\ell} r_{\ell-1} + r_{\ell} & \deg r_{\ell} < \deg r_{\ell-1} \\
 r_{\ell-1} = q_{\ell+1} r_{\ell} &
 \end{array}$$

Ekkor az utolsó r_{ℓ} nemnulla maradék a **legnagyobb közös osztó** (leosztva a főegyütthatóval).

```
[12]: f, g = x^4+x^3+x^2+2*x+1, x^4+x^3+4*x^2+1
      q,r1 = f.quo_rem(g); r1
```

```
[12]: 2*x^2 + 2*x
```

```
[13]: q,r2 = g.quo_rem(r1); q,r2
```

```
[13]: (3*x^2 + 2, x + 1)
```

```
[14]: q,r3 = r1.quo_rem(r2); q,r3
```

```
[14]: (2*x, 0)
```

```
[ ]:
```

```
[15]: f, g = x^3+4*x^2+x+1, 3*x^2+1
      q,r1 = f.quo_rem(g); r1
```

```
[15]: 4*x + 3
```

```
[16]: q,r2 = g.quo_rem(r1); r2
```

```
[16]: 3
```

```
[17]: q,r3 = r1.quo_rem(r2); r3
```

```
[17]: 0
```

Tehát a legnagyobb közös osztó: $(f, g) = 1$ ($r_\ell = 3$, ennek főegyütthatója 3, $r_\ell/3 = 1$)

```
[18]: def lnko(f,g):
      R = (f,g)
      while R[1].degree()>=0:
          R = R[1], R[0].quo_rem(R[1])[1]
          print("r_i =", R[1])
      return R[0]
```

```
[19]: f, g = x^4+x^3+x^2+2*x+1, x^4+x^3+4*x^2+1
      lnko(f,g)
```

```
r_i = 2*x^2 + 2*x
r_i = x + 1
r_i = 0
```

```
[19]: x + 1
```

```
[20]: for a in range(5):
      print( "f(",a,") =", f(a))
```

```
f( 0 ) = 1
f( 1 ) = 1
f( 2 ) = 3
f( 3 ) = 4
f( 4 ) = 0
```

```
[21]: for a in range(5):
      print( "g(",a,") =", g(a))
```

```
g( 0 ) = 1
g( 1 ) = 2
g( 2 ) = 1
g( 3 ) = 0
g( 4 ) = 0
```

```
[22]: f, g = x^3+4*x^2+x+1, 3*x^2+1
      lnko(f,g)
```

```
r_i = 4*x + 3
r_i = 3
r_i = 0
```

```
[22]: 3
```

```
[23]: for a in range(5):
      print( "f(",a,") =", f(a))
```

```
f( 0 ) = 1
f( 1 ) = 2
f( 2 ) = 2
f( 3 ) = 2
f( 4 ) = 3
```

```
[24]: for a in range(5):
      print( "g(",a,") =", g(a))
```

```
g( 0 ) = 1
g( 1 ) = 4
g( 2 ) = 3
g( 3 ) = 3
g( 4 ) = 4
```

```
[25]: f, g = (x-1)*(x-2)^2, x*(x-1)^2*(x-2)
      h = lnko(f,g)
```

```
r_i = x^3 + 3*x + 1
r_i = 2*x^2 + 4*x + 4
r_i = 0
```

```
[26]: f
```

```
[26]: x^3 + 3*x + 1
```

```
[27]: g
```

```
[27]: x^4 + x^3 + 3*x
```

```
[28]: for a in range(5):  
      print( "f(",a,") =", f(a))
```

```
f( 0 ) = 1  
f( 1 ) = 0  
f( 2 ) = 0  
f( 3 ) = 2  
f( 4 ) = 2
```

```
[29]: for a in range(5):  
      print( "g(",a,") =", g(a))
```

```
g( 0 ) = 0  
g( 1 ) = 0  
g( 2 ) = 0  
g( 3 ) = 2  
g( 4 ) = 2
```

```
[30]: for a in range(5):  
      print( "h(",a,") =", h(a))
```

```
h( 0 ) = 4  
h( 1 ) = 0  
h( 2 ) = 0  
h( 3 ) = 4  
h( 4 ) = 2
```