

Komputeralgebra rendszerek

Polinomok I Alapfogalmak

Czirbusz Sándor

czirbusz@gmail.com

Komputeralgebra Tanszék
ELTE Informatika Kar

2010-2011 ősz

Index I

- 1 Egyváltozós polinomok
 - Alapfogalmak
 - Egyváltozós polinomok a Maple–ban
 - Műveletek
 - Egyéb műveletek
 - FaktORIZÁCIÓ és GCD
 - Egyváltozós polinomok a Sage–ben
 - Alapok I
 - Alapok II
 - Polinomok \mathbb{Z} fölött
- 2 Többsváltozós polinomok
 - ... a Maple–ben
 - ..a Sage–ben
- 3 Racionális törtfüggvények
- 4 Konverziók

Index II

5 Példák

Alapfogalmak

- A polinom fogalma : egy gyűrű elemeiből alkotott csak véges sok nem zérus elemet tartalmazó sorozat.
- Jelöléstechnika : $\sum_{i=1}^n a_i x^i$ ahol a_i a gyűrű eleme, az x szimbólum a "változó".
- Jellemzők: fokszám, főegyüttható, főtag.
- Számítógépes ábrázolás : kanonikus forma (extended canonical form), összevont forma (collected form)

Műveletek

Információk és alapműveletek

Információk

- `type(p,polynomial)`
- `lcoeff, tcoeff, coeff`
- `coeffs,powers`
- `degree`
- `testeq`

Műveletek

- összeadás
- szorzás
- osztás :
 $q := \text{quo}(p_2, p_1, x, 'r')$,
 $r := \text{rem}(p_2, p_1, x, 'q')$,
`divide (p2,p1,'q')` (Ez
 utóbbi csak \mathbb{Q} fölött.)
- `expand`
- `collect`
- `sort()`

A PolynomialTools csomag

Csak az egyszerűbbek :

- **CoefficientList**(...) – az együtthatók listája, futtatható nem kanonikus polinomra
- **CoefficientVector**
- **Translate**(p, x, c) – az "x" változó helyett "x+c"-be "tolja" a polinomot (Létezik olyan változata, amely csak az expand-ot használja)
- **GcdFreeBasis**(p_1, p_2, \dots, p_n) – a polinomok által generált ideál egy gcd-mentes bázisa (Mi is az ideál?)

Faktorizáció és GCD

A \mathbb{Z} -beli számítás esetén a név egy „i” szuffixet kap, alapértelmezett \mathbb{Q} .

A gcd és lcm

- Az lko **[i]gcd**(p_1, p_2)
- A bővített euklidesz algoritmusra : **[i]gcdex**
- Létezik lkkt is : **[i]lcm**

Faktorizáció

- Az alapfüggvény : **[i]factor**(**p**), ez \mathbb{Z}, \mathbb{Q} -ban
- Bővítőben : **factor**(**p, K**), ahol a "K" a bővítés, pl. A Gauss egészek fölött : **factor**(**p, I**)
- Maradékosztályokban : $Factor(p) \bmod m$, a függvény **inert** formájával
- Véges testekben a bővítő polinommal adhatjuk meg. (Pl \mathbb{Z}_4 -ben $x^2 + x + 1$)
- A két forma keverhető
- Az inert forma használható a **sort**, **expand** függvényekre is

A Sage algebrai értelmezése

- A Maple-el ellentétben a Sage a "csak úgy" beírt polinomokat nem tekinti számstruktúra felettinek, ez automatikusan a "szimbolikus gyűrű" eleme lesz
- Ha számstruktúra fölött polinomokkal akarunk dolgozni, előre kötelező a gyűrűt definiálni.
- A gyűrű (is) önálló matematikai objektumként használható programozáshoz is
- Az objektumok két fontos adata :
 - Hova tartozunk : **.parent()**
 - Honnan "jöttünk" : **.base_ring()**
- Egy gyűrű általában több módon definiálható.

A szimbolikus polinomgyűrű

A szimbolikus gyűrű

- A **.parent()** függvény eredménye: Symbolic Ring
- A beszorzás itt sem automatikus, kell az **.expand()**
- Az azonos fokú tagok összegyűjtése és a rendezés automatikus
- A megfelelő függvények hasonlóak:
.leading_coeff(), **.trailing_coeff()** **.degree()**
- Itt nincs osztás !
- Összeadás, szorzás, faktorizáció, gcd működik, bővített gcd nincs !
- A függvényekben általában kötelező a változót feltüntetni (hiszen szimbolikus!)

Számstruktúra fölötti polinomok

$\mathbb{Z}[x]$ és a többiek

- A gyűrű létrehozása : pl. $R. < x > = \text{PolynomialRing}(\mathbb{Z}\mathbb{Z})$ vagy $R = \mathbb{Z}\mathbb{Z}[x]$ (avagy QQ , stb)
- A beszorzás **is** automatikus (néha sajnos)
- A függvénynevek nem konzisztensek : pl. **.leading_coefficient()** (rövidítés nélkül)
- Nincs utolsó tag lekérdezés, csak konstans-tag
- Az együtthatók lekérdezhetők : **.coefficients()**, ez egy lista, nem tartalmazza a 0-kat.
- Listává alakítható : **.list()** vagy **.coeffs()** – tartalmazza a 0-kat
- Van bővített euklideszi is : **.xgcd()**
- Van maradékos osztás **.quo_rem**, vagy külön **//** és **%**

... a Maple-ben

Specialitások

- Műveletek
Ugyanazok, mint az egyváltozósoknál
- Rendezés sort(expr, method)
 - Teljes fokszám alapján (default)
 - Tiszta lexikografikus

$$x^i y^j < x^{i'} y^{j'} \iff i < i' \text{ vagy } (i = i', j < j')$$

- A disztributív és rekurzív alak között a **collect()** függvény paramétereivel váltunk (**recursive**, **distributed**, ezen belül a változósorrend is állítható)

..a Sage-ben

Specialitások

- Az egyváltozóséhoz hasonlóan ha nem definiálunk alapgyűrűt, akkor a Symbolic Ring-ben dolgozunk
- A gyűrű definiálása hasonló az egyváltozóséhoz, opcionálisan megadható a kitevő-rendezettség (order)
 - lex = lexikografikus
 - degrevlex = teljes fokszám szerinti, egy kifejezésen belül fordított lexikografikus
 - deglex, invlex, ...
- Az "x" kivételével kötelező a változókat a **var** kulcsszóval definiálni.
(Kivéve, ha a munkalapot egy **automatic_names** utasítással kezdjük)
- A fokszám szerinti rendezés automatikus

.. a Maple-ban

- Racionális törtfüggvény, **numer**, **denom**
- Az egyszerűsítés nem automatikus, kivéve ha azonnal felismerhető a közös faktor
- Egyszerűsítés: **normal**

..a Sage-ben

- Itt is különbséget kell tennünk a szimbolikus gyűrű és a számstruktúrák fölötti objektumok között.
 - Két szimbolikus polinom hányadosa a szimbolikus gyűrűben van (ugyanis test)
 - $\mathbb{Z}[x]$ esetén automatikusan létrehozza a hányadostestet.
- A számláló, nevező : **.numerator()** és **.denominator()**
- $\mathbb{Z}[x]$ esetén automatikusan az egyszerűsítés

Konverziók Maple–ban

- Használt package : **codegen**, illetve **numapprox**
- Egy kifejezés költsége (műveleteinek száma) : **cost**
- Horner elrendezés : **convert(p, 'horner')**
- Lánctörtbe fejtés : **convert(p, 'confrac', x)**
- Parciális törtekre bontás : **convert(p, 'parfrac, x[, domain])**
- Parciális törtekre bontás a felbontási test fölött : **convert(p,'fullparfrac',x)**

.. és a Sage-ben

Sajnos, ezen a területen a Sage gyakorlatilag semmit sem tud

Példák

Lásd a munkalapokon