

Komputeralgebra rendszerek

Polinomok III

További műveletek polinomokkal

Czirbusz Sándor

czirbusz@gmail.com

Komputeralgebra Tanszék
ELTE Informatika Kar

2010-2011 őszi

Index I

- 1 Egyszerűsítés, normalizálás
 - Az absztrakció szintjei
 - Az egyszerűsítés problémája
 - Egyszerűsítők
- 2 Maple megvalósítások
 - Kifejtés
 - Szorzattá alakítás
 - Egyszerűsítés
 - Normalizáció
 - Együtthatók összevonása
 - Rendezés
- 3 Sage megvalósítások
 - Kifejtés
 - Szorzattá alakítás

Index II

- Egyszerűsítés
- Normalizáció
- Együtthatók összevonása

Az absztrakció szintjei

Objektum-szint

Maga a matematikai objektum.

A forma szintje

Az objektum egy megjelenési formája a kiválasztott szimbólumok segítségével.

- $a(x, y) = 12x^2y - 4xy + 9x - 3$
- $a(x, y) = (3x - 1)(4xy + 3)$
- $a(x, y) = (12y)x^2 + (-4y + 9)x$

Az adatstruktúra szintje

Az adott forma számítógépes reprezentációja.

Az egyszerűsítés problémája

Az egyszerűsítés célja

- Erőforrás-kímélés
- Emberi értelmezhetőség
- $(x + y)^{1000} - y^{1000}$ vs. $x^{1000} - y^{1000}$

Zéró-ekvivalencia

Egyszerűbb kérdés, teljes általánosságban algoritmikusan megoldhatatlan:

$$\log\left(\tan\left(\frac{x}{2}\right)\right) + \sec\left(\frac{x}{2}\right) - \sinh^{-1}\left(\frac{\sin(x)}{1 + \cos(x)}\right)$$

Egyszerűsítők

- E a kifejezések halmaza, valamilyen rendezési relációval.
 \sim egy ekvivalenciareláció.
- \equiv a forma szintű, $=$ az objektum szintű egyenlőség.
- **Normál** függvény: $f : E \rightarrow E$ -re teljesülnek a következők:
 - $f(a) \sim a$ minden $a \in E$ -re
 - Ha $a \sim 0$, akkor $f(a) \equiv f(0)$
- **Kanonikus** függvény: normálfüggvény és ha $a \sim b$, akkor
 $f(a) \equiv f(b)$
- Normálfüggvény megvalósítása: végezzük el a polinomra a lehetséges beszorzásokat, majd gyűjtsük össze az azonos fokú tagokat.
- Kanonikus függvény megvalósítása: normálfv + rendezzük a tagokat csökkenő sorrendben.

Kifejtés

expand

- Szintaxis: **expand(kif[,kif1,kif2,...])** – "Felbontja a zárójeleket"
- A **kifn**–részkifejezéseket „nem bántja”
- A negatív kitevőkkel nem foglalkozik (a nevezőre külön alkalmazandó)
- A nem-egész kitevőkkel nem foglalkozik
- Véges gyűrűk, testek fölött: **Expand(expr) mod n** plusz **evala**
- Nem csak polinomokra alkalmazható (lásd HELP)

Szorzattá alakítás

factor

- Szintaxis: A **factor(a [, K])** procedúra – az **expand** „testvére”.
Tényezőkre bontja az alapstruktúra fölött a polinomot. (Faktorizált normálforma)
- A K polinom a \mathbb{Q} test bővítését írja le
- Véges gyűrűk fölött: **Factor**
- Felbontási test generálása:
PolynomialTools:-Split(polynom)
- **AFactor** - a \mathbb{C} fölötti faktorizáció
- Négyzetmentes faktorizáció **convert(poly,squarefree), s|Sqrfree(poly)**

Egyszerűsítés

Az automatikus egyszerűsítés mellett szükségünk van az „igény szerinti” egyszerűsítésre:

`simplify`

Szintaxis: `simplify(expr, options)`. Teljes: `?simplify`

Értelmezés: A kifejezést egyszerűsíti „igény” szerint

Opciók : pl. `radical`, `ln`. A különböző jellegű kifejezések speciális egyszerűsítőinek meghívására.

Normalizáció

`normal`

Normalizálást (\mathbb{Q}) fölötti racionális kifejezéseken. (Faktorizált normálforma)

- Szintaxis: **`normal(f [, expanded])`**
- Racionális törtfüggvényt hoz normálformára
- Az **`expanded`** paraméter a számlálót és a nevezőt felbontja
- Általánosított racionális kifejezéseken is működik
- **`Normal + mod`** a véges struktúrákban

Együtthatók összevonása

collect

- Szintaxis: **collect(a, x [, form] [, func])**: az együtthatók összevonását végzi.
- A **form** paraméter „recursive” (default), vagy „distributed” lehet
- A **func** opció többnyire egyszerűsítés, vagy faktorizáció
- Alkalmazható általánosított rac. kifejezésre
- LargeExpressions csomag: **Veil Unveil** összetett kifejezés elrejtésére

Rendezés

sort

- Szintaxis: **sort(A, [V] [, opt1, opt2, ...])**
- „A” a polinom, „V” a változók
- Az opciók: a rendezés módja
 - Lehet a polinom fokszám-kezelési módja
 - Lehet „ascending”, „descending”
- Listákra is alkalmazható

Kifejtés

`.expand`

- Szintaxis: **`.expand([side=None])`**
- Használható relációs–kifejezésekre, ekkor van értelme a „side” paraméternek
- Használható funkcionális formában
- Léteznek **`.expand_log()`**, **`.expand_trig()`**, **`.expand_rational()`** formák is a megfelelő kifejezésekre

A számstruktúrák fölött automatikus a beszorzás!

Szorzattá alakítás

A `.factor()` és társai

- Szimbolikus gyűrű fölött csak egyszerű illetve **`.factor()`** illetve egy rögtön listát adó **`.factor_list()`** függvény létezik
- Számstruktúrában az alapfüggvény mellett lehetőség van a gyűrűből való kilépés nélkül modulárisan prím szerint faktorizálni: **`.factor_mod(3)`**. Ha a faktorizálással eredménye a zéruspolinom, hibaüzenetet kapunk-
- Szintén itt fölött működik a p-adikus felbontás: **`.factor_padic(p, prec=10)`**

Egyszerűsítés

simplify

A különböző feladatokra A Maple-től eltérően különböző függvények szolgálnak:

- Polinomiális kifejezésekre: **.simplify()**
- Trigonometrikus, logaritmikus, exponenciális kifejezések: **.simplify_trig()**, **.simplify_log()**, **.simplify_exp()**,
- Törtkifejezésekre, gyökös kifejezése: **.simplify_rational()**, **.simplify_radical()**
- Kombinatorikus kifejezések: **.simplify_factorial()**
- A **.simplify_full()** függvény sorrendben a „factorial”, „trig”, „rational” és „radical” futtatása

Normalizáció

A Sage–ben igazi normalizáció nincs. A **.simplify()** függvény részleges megoldást nyújt.

- A **method** paraméter használata
 - „simple” két kiszorzott polinom hányadosaként állítja elő
 - „full” (default) ha szükséges, az előző többszöri alkalmazása
 - „noexpand” csak közös nevezőre hoz
- A „map” (default=false) paraméter a „true” értéknél rész kifejezéseken végzi az egyszerűsítést
- A **.collect_common_factors()**–al kikényszeríthető a lehetséges egyszerűsítés

Együtthatók összevonása

A Sage-ban a beszorzás utáni együttható összevonás számstruktúra fölött automatikus. (A gyűrűbeli fokszám-rendezettségnek megfelelően).

Szimbolikus gyűrűben:

.collect(sym) a megfelelő rekurzív formába alakít

Példák

Lásd a munkalapokon