

Komputeralgebra rendszerek

III. A számok kezelése

Czirbusz Sándor

`czirbusz@gmail.com`

Komputeralgebra Tanszék

ELTE Informatika Kar

D2.711A

2009-2010 tavasz

Index

1 Egész- és racionális számok

- Egzakt aritmetika
- A Maple egészei
- Egészek a Sage-ben
- Racionális számok
- A Maple valós számai
- A Sage valós számai
- Algebrai számok
- Komplex számok

- # Komputeralgebra rendszerek

Példák (Maple)

- Egész : $2^{100} = 1606938044258990275541962092341162602522202993782792835301376$
- Racionális $5/7 = \frac{5}{7}, 2/4 = \frac{1}{2}$
- Valós : $\text{evalf}(5/7) = 0.7142857143$ (10 jegyig pontos)
- $2/3 \cdot 3/2 = 1$

Egészek

- Dinamikus adatvektor

$\text{int} \pm n$	i_0	i_1	\dots	i_n
--------------------	-------	-------	---------	-------

- Kis egészek : egy szóban
- Néhány függvény :
 - `kernelopts(maxdigits)`
 - `isprime`, `ifactor`, `nextprime`, `isqrt`
 - `iquo`, `irem`, `testeq`, `igcd`, `igcdex`

Egészek a Sage-ben

A Sage objektumként kezeli a számokat is, így a függvények jelentős része kétféleképp érhető el. Az előzőeknek megfelelő függvények :

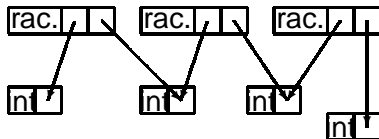
- Kétféle szintaxis : $\text{factor}(n)$ és $n.\text{factor}()$;
- $n.\text{is_prime}()$, $n.\text{next_prime}()$, $n.\text{is_square}$
- $n.\text{quo_rem}$, $n.\text{gcd}()$, $n.\text{xgcd}()$

A Python listakezelő nyelv, ezért a több értéket visszaadó függvények listába rakják az eredményt.

- Az objektumok „családi kapcsolata” lekérdezhető :
 $n.\text{parent}()$, $n.\text{base_ring}()$

Racionális számok I

Ha az egészek ábrázolása adott, a CAS-ok a racionális számokat ugyanúgy ábrázolják :



Az „összenyilazás” a Maple jellegzetessége

Racionális számok II

- Mind a Maple mind a Sage automatikus egyszerűsítést végez : a számláló és a nevező gcd-jével osztja a számlálót és a nevezőt
- A számláló és a nevező hasonló nevű függvényekkel külön kezelhető : *numer()*, *denom()*

Lebegőpontos és hardware lebegőpontos számok

- Belső ábrázolás adatvektora :

FLOAT	mantissza	karakterisztika
-------	-----------	-----------------

- A kitevő kezelése a C egyszeres pontosságú aritmetikájával
- Lebegőbontos szám megadása : 0.000001 , $0.1 * 10^{-5}$,
Float(10, -7)
- *evalf*(..), *convert*(..)
- *interface*(displayprecision=*n*) : kijelzési pontosság, $n = 1$ az alaphelyzet
- A **evalhf** procedúra – HW lebegőpontos aritmetika, a grafika használja. Az eredményeket duplapontosan adja, (Digits=15)

Tetszőleges- és dupla pontoságú lebegőpontos számok

Tetszőleges pontosságú számok

- Ez felel meg a Maple lebegőpontosnak
- A RR gyűrű elemei, alapértelmezetten 53–bites pontosságú, gyűrűdefinícióval ez módosítható
- Konverzió $a.n()$, a Sage megkülönbözteti paraméterként az értékes jegyeket és a pontosságot

Dupla pontos számok

- A hardware–lebegőpontosnak felel meg, az RDF –gyűrű a szülő–objektum.
- A pontosság 15 bit.

Algebrai számok a Maple–ban I

- nem alapvető adattípus
- Ábrázolás : a RootOf procedúra segítségével
- Példa

```
alpha := RootOf(x^7-2):  
simplify(alpha^7);  
2
```

- Az α értékkel ugyanúgy dolgozhatunk ezután, mint más értékkel.

Algebrai számok a Sage–ben I

- A megvalósításban itt már komoly különbség van a két rendszer között. A Sage–ban a racionális számkör a default, de a többi szám–struktúrában explicite definiálni kell a \mathbb{Q} testbővítését.
- Példa :

$$K.<a> = \text{NumberField}(x^7-2)$$

$$a^7$$

$$2$$

- Innentől kezdve a K testet is használhatjuk, az összes $a + 1, 2 * a$ alakú kifejezés e test eleme.

Komplex számok a Maple-ban

- Az i helyett I -t használ
Ez felülírható : például j -re `interface(imaginaryunit = j)`
- Megadás literálként $a + b * I$, vagy konstruktorral
`Complex(a, b)`
- A klasszikus adatok `Re()`, `Im()`, `conjugate()`, `abs()`, `argument()`
- Komplex előjel : `csgn()`
- A `csgn(0)` érték a `_Envsignum0` környezeti változóval állítható.
- Különböző kiértékelő függvények : `evalc()` szimbolikusan kezeli a komplex számot, `evalf()`, `evalhf()` a valós és komplex részt lebegőpontosan

Komplex számok a Sage-ben I

- A szigorú algebrai felépítés miatt a \mathbb{R} bővítése a *ComplexField* vagy *CC* test, így konkrét értékekkel kell számolni :
- Példa :

```
a = CC(12 + i); a
12.000000000000000 + 1.000000000000000*I
```

- A komplex számról információk :
a.abs(), *a.arg()*, *a.imag()*, *a.real()*, *a.conjugate()*
- Ugyanúgy, mint a valós számokban, definiálhatók a komplex dupla-pontosnak is.

Komplex számok a Sage-ben II

- A Sage szimbolikus gyűrűje fölött szimbolikusan is számolhatunk komplexekkel

```
a = 1 + i; a
      Symbolic Ring
a.imag()
      1
a*a.conjugate()
      2
```

- A függvények ugyanazok, minden művelet ugyanúgy történik.