Sage Quick Reference: Polynomial ring

NUMATA, Y.

Sage Version 3.4

`http://wiki.sagemath.org/quickref`

Based on work by Peter Jipsen, William Stein

おもな環と体

厳密 (exact)

- `ZZ`　整数 $\mathbb{Z}$, 環
- `QQ`　有理数 $\mathbb{Q}$, 体
- `QQbar`　代数閉包 $\overline{\mathbb{Q}}$
- `GF(2)`　mod 2, 体, specialized implementations
- `GF(p) == FiniteField(p)`　p 素数, $\mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p$, 体
- `Integers(6)`　integers mod 6, $\mathbb{Z}/6\mathbb{Z}$, 環
- `CyclotomicField(7)`　$\mathbb{Q}$ に 1 の 7 乗根を添加した体
- `QuadraticField(-5, 'x')`　$\mathbb{Q}$ に x=$\sqrt{-5}$ を添加した体
- `SR`　ring of symbolic expressions

近似 (inexact)

- `RDF`　倍精度実数
- `RR`　53-bit 精度実数
- `RealField(400)`　400-bit 精度実数
  (`CDF`, `CC`, `ComplexField(400)`　複素数も有)
- `RIF`　実区間演算, 体

整数: $\mathbb{Z}$ = `ZZ`　例 `-2 -1 0 1 10^100`

有理数: $\mathbb{Q}$ = `QQ`　例 `1/2 1/1000 314/100 -2/1`

実数: 例 `.5 0.001 3.14 1.23e10000`

複素数: $\mathbb{C} \approx$ `CC`　例 `CC(1,1) CC(2.5,-3)`

倍精度 (Double): `RDF` and `CDF`　例 `CDF(2.1,3)`

Mod n: $\mathbb{Z}/n\mathbb{Z}$ = `Zmod`　例 `Mod(2,3) Zmod(3)(2)`

有限体: $\mathbb{F}_q$ = `GF`　例 `GF(3)(2) GF(9,"a").0`

多項式: $R[x,y]$　例 `S.<x,y>=QQ[] x+2*y^3`

巾級数: $R[[t]]$　例 `S.<t>=QQ[[]] 1/2+2*t+O(t^2)`

ローラン多項式: $R[t,t^{-1}]$

　例 `S.<t>=LaurentPolynomialRing(QQ,1) 1+t+t^{-2}`

整域 R の商体: 例 `FractionField(R)`

p 進整数: $\mathbb{Z}_p \approx$`Zp`, $\mathbf{Q}_p \approx$`Qp` 例 `2+3*5+O(5^2)`

文字列のリスト

`['x_%d' % i for i in (1..6)]`

　`'x_1','x_2',...,'x_6'` からなるリスト

多項式環の生成

多項式環を生成し使うための方法はいくつかある.

有理数係数二変数多項式環を定義し, 第 1 変数と第 2 変数の 2 次斉次完全対称式を f と置く例:

```
names = ['x','y']
R = PolynomialRing(QQ,names)
(x,y) = R.gens()
f = x^2+x*y+y^2
```

```
names = ['x','y']
R = QQ[names]
(x,y) = R.gens()
f = x^2+x*y+y^2
```

```
R.<x,y> = QQ[]
f = x^2+x*y+y^2
```

```
names = ['t0','t1']
R = PolynomialRing(QQ,names)
t = R.gens()
f = t[0]^2+t[0]*t[1]+t[1]^2
```

```
R = PolynomialRing(QQ,'t',2)
t = R.gens()
f = t[0]^2+t[0]*t[1]+t[1]^2
```

環の操作

- `R.gens()`　生成元達
- `R.term_order()`　使用する term order
- `R.quatient(I)`　環 R/I (I は ideal)
- `R.factional_field()`　商体 (R は整域)
- `R.change_ring(K)`　係数を K に変更
- `R.change_ring(order=t)`　Term order を t に変更
- `R.random_element()`　ランダムに元を返す
- `R.is_ring()`, `R.is_field()`, `R.is_integral_domain()`, `R.is_exact()`, etc.

Ideal の操作

- `S=[f,g];I=R.ideal(S)`　S を生成系とする R の ideal I
- `I+J`　$I+J$,
- `I.intersection(J)`　$I \cap J$,
- `I.quotient(J)`　$(I:J)$,
- `I.weil_restriction()`　Weil restriction
- `I.homogenize()`　変数を増やし斉次化したイデアル.
- `I.elimination_ideal([x,y])`　x, y を含まない環に I を制限した ideal
- `I.change_ring(D)`　(Term Order を変更するなどした) 別の環 D のイデアルとみなす.
- `I.embedded_primes()`,
- `I.associated_primes()`,
- `I.minimal_associated_primes()`,
- `I.primary_decomposition()`,
- `I.complete_primary_decomposition()`
- `I.radical()`
- `I.variety()`
- `I.dimension()`　R/I の Krull 次元
- `I.vector_space_dimension()`　R/I の線形空間としての次元
- `I.hilbert_polynomial()`　Hilbert 多項式
- `I.hilbert_series()`　Hilber 級数
- `I.gens()`　I の生成系
- `I.integral_closure()`,　`I.triangular_decomposition()`,
- `I.syzygy_module()`, etc
- `f in I`　f が I に含まれていれば `True`
- `I.is_trivial()`,　`I.is_zero()`,　`I.is_one()`,
- `I.is_maximal()`,　`I.is_primary()`,　`I.is_prime()`,
- `I.is_principal()`,　`I.is_homogeneous()`,
- `I.is_idempotent()`, etc.

元の操作

- `f+g`　$f+g$,　`f*g`　$f \cdot g$,
- `f.inverse_of_unit()`　(単元なら) f^{-1}
- `D(f)`　f を別の環 D の元に読み替える.
  例: `R=ZZ['x'];D=GF(2)['x'];x=R.gen();f=2*x;d=D(f)`
- `f.monomials()`　f に含まれる単項式 (係数は 1) のリスト
- `f.exponents()`　f に含まれる単項式の冪のリスト
- `f.coefficients()`　係数のリスト
- `f.dict()`　冪に係数を対応させた辞書
- `f.constant_coefficient()`　定数項
- `f.monomial_coefficient(x^2*y)`　f での x^2y の係数
- `f.coefficient({x:2,y:1})`　f での x^2y の係数
- `f.lc()`　先頭単項式の係数
- `f.lm()`　先頭単項式 (係数は 1)
- `f.lt()`　先頭項 (`f.lt()==f.lc()*f.lm()`)
- `f.variables()`　f に含まれる変数のリスト
- `f.variable(i)`　f に含まれる i 番目の変数
- `f.truncate(x,i)`　x の冪が i 未満の項だけの和
- `f.factor()`,　`f.gcd(g)`,　`f.lcm(g)`
- `f.derivative(x)`= $\frac{\partial}{\partial x} f$,　`f.derivative(x,n)`= $\frac{\partial^n}{\partial x^n} f$
- `f.gradient()`= $(\frac{\partial_1}{\partial x_1} f, \frac{\partial_2}{\partial x_2} f, \ldots)$　`f.jacobian_ideal()`
- `f.total_degree()`　f の全次数
- `f.degree(x)`　x を変数とする 1 変数多項式としての次数

`f.degrees()` (f.degree(x1),f.degree(x2),..)
Note: 先頭単項式の次数ではない

`f.subs({x:y+1})` x に $y+1$ を代入.
`f.subs(x=y+1)` でも OK.

`f.quo_rem(g)` f を g で割った商と余り

`f.map_coefficients(phi)`
$f = \sum c_\alpha x^\alpha$ と関数 $\varphi \rightsquigarrow \sum \varphi(c_\alpha) x^\alpha$

`f.homogenize()`, `f.resultant()`, `f.discriminant()`, `f.sylvester_matrix()`, etc.

`f.divides(g)`, `f.is_constant()`, `f.is_generator()`, `f.is_homogeneous()`, `f.is_idempotent()`, `f.is_monomial()`, `f.is_nilpotent()`, `f.is_one()`, `f.is_square()`, `f.is_squarefree()`, `f.is_unit()`, `f.is_univariate()`, `f.is_zero()`, etc

多項式のリストへの操作

`L=[f,g]` 多項式 f と g からなるリスト
`sum(L)`$= \sum_{h \in L} h$
`prod(L)`$= \sum_{h \in L} h$
`L.sort()` Term order でソートする.

Göbner basis

`I.groebner_basis()` I の Gröbner 基底.
`I.reduce(f)` I の Gröbner 基底で f を割った余り.
`I.basis_is_groebner()` $I.gens()$ は Gröbner 基底か

Gröbner fan

S を生成系とする R のイデアルの Gröbner fan のレイを確認:

```
I = R.ideal(S)
F = I.groebner_fan()
P = F.polyheadralfan()
P.rays()
```

`F.dimension_of_homogeneity_space()`, `F.maximal_total_degree_of_a_groebner_basis()`, `F.minimal_total_degree_of_a_groebner_basis()`, `F.number_of_reduced_groebner_bases()`, `F.reduced_groebner_bases()`, `F.tropical_basis()`, `F.tropical_intersection()`, `F.weight_vectors()`, etc.

Nowton polytopes

`N=f.newton_polytope`
`N.is_simple()`, `N.show()`, etc. (See `N.`⟨tab⟩)

Toric ideal

```
A = matrix([[1,1,1],[0,1,2]])
T = ToricIdeal(A)
T.ker()
```

環を指定するには `ToricIdeal(A,polynomial_ring=R)`

Boolean Polynomial Ring

$\mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$
`R.<x, y, z> = BooleanPolynomialRing()`

Term Order

Term Order を指定して多項式環を定義.
`t=TermOrder('lex');`
`R=PolynomialRing(QQ,['x','y'],order=t)`
Note: `PolynomialRing(QQ,['x','y'],order='lex')` も可

主な順序:
`degrevlex` (次数逆辞書式), `deglex` (次数辞書式), `lex` (純辞書式), `invlex` (Inverse lexicographic), `degneglex` (Degree negative lexicographic).
重みを指定するときは `TermOrder('wdeglex',(1,2,3))'`
重みを必要とする主な順序:
`wdegrevlex` (Weighted degree reverse lexicographic), `wdeglex` (Weighted degree lexicographic), `negwdegrevlex` (Negative weighted degree reverse lex.)
TermOrder 同士の和はブロック順序.
Note: 変数の数を指定しなければならない.
例: `TermOrder('deglex',2)+TermOrder('deglex',3)`