

# Contents

<b>1</b>	<b>Classes</b>	<b>2</b>
1.1	poly.ring – polynomial rings	2
1.1.1	PolynomialRing – ring of polynomials	3
1.1.1.1	getInstance – classmethod	4
1.1.1.2	getCoefficientRing	4
1.1.1.3	getQuotientField	4
1.1.1.4	issubring	4
1.1.1.5	issuperring	4
1.1.1.6	getCharacteristic	4
1.1.1.7	createElement	4
1.1.1.8	gcd	4
1.1.1.9	isdomain	5
1.1.1.10	iseuclidean	5
1.1.1.11	isnoetherian	5
1.1.1.12	ispid	5
1.1.1.13	isufd	5
1.1.2	RationalFunctionField – field of rational functions	5
1.1.2.1	getInstance – classmethod	6
1.1.2.2	createElement	6
1.1.2.3	getQuotientField	6
1.1.2.4	issubring	6
1.1.2.5	issuperring	6
1.1.2.6	unnest	6
1.1.2.7	gcd	6
1.1.2.8	isdomain	7
1.1.2.9	iseuclidean	7
1.1.2.10	isnoetherian	7
1.1.2.11	ispid	7
1.1.2.12	isufd	7
1.1.3	PolynomialIdeal – ideal of polynomial ring	7
1.1.3.1	reduce	8
1.1.3.2	issubset	8
1.1.3.3	issuperset	8

# Chapter 1

## Classes

### 1.1 poly.ring – polynomial rings

- Classes
  - **PolynomialRing**
  - **RationalFunctionField**
  - **PolynomialIdeal**

### 1.1.1 PolynomialRing – ring of polynomials

A class for uni-/multivariate polynomial rings. A subclass of **CommutativeRing**.

#### Initialize (Constructor)

```
PolynomialRing(coeffring: CommutativeRing, number_of_variables:  
integer=1)  
→ PolynomialRing
```

`coeffring` is the ring of coefficients. `number_of_variables` is the number of variables. If its value is greater than 1, the ring is for multivariate polynomials.

#### Attribute

**zero** :  
zero of the ring.

**one** :  
one of the ring.

## Methods

### 1.1.1.1 getInstance – classmethod

**getInstance**(coeffring: *CommutativeRing*, number\_of\_variables: *integer*)  
→ *PolynomialRing*

return the instance of polynomial ring with coefficient ring `coeffring` and number of variables `number_of_variables`.

### 1.1.1.2 getCoefficientRing

**getCoefficientRing**() → *CommutativeRing*

### 1.1.1.3 getQuotientField

**getQuotientField**() → *Field*

### 1.1.1.4 issubring

**issubring**(other: *Ring*) → *bool*

### 1.1.1.5 issuperring

**issuperring**(other: *Ring*) → *bool*

### 1.1.1.6 getCharacteristic

**getCharacteristic**() → *integer*

### 1.1.1.7 createElement

**createElement**(seed) → *polynomial*

Return a polynomial. `seed` can be a polynomial, an element of coefficient ring, or any other data suited for the first argument of uni-/multi-variate polynomials.

### 1.1.1.8 gcd

**gcd**(a, b) → *polynomial*

Return the greatest common divisor of given polynomials (if possible). The polynomials must be in the polynomial ring. If the coefficient ring is a field, the result is monic.

**1.1.1.9 isdomain**

**1.1.1.10 iseclidean**

**1.1.1.11 isnoetherian**

**1.1.1.12 ispid**

**1.1.1.13 isufd**

Inherited from **CommutativeRing**.

## **1.1.2 RationalFunctionField – field of rational functions**

### **Initialize (Constructor)**

**RationalFunctionField**(field: *Field*, number\_of\_variables: *integer*)  
→ *RationalFunctionField*

A class for fields of rational functions. It is a subclass of **QuotientField**.

`field` is the field of coefficients, which should be a **Field** object. `number_of_variables` is the number of variables.

### **Attribute**

**zero** :  
zero of the field.

**one** :  
one of the field.

## Methods

### 1.1.2.1 getInstance – classmethod

```
getInstance(coefffield: Field, number_of_variables: integer)  
→ RationalFunctionField
```

return the instance of `RationalFunctionField` with coefficient field `coefffield` and number of variables `number_of_variables`.

### 1.1.2.2 createElement

```
createElement(*seedarg: list, **seedkwd: dict) → RationalFunction
```

### 1.1.2.3 getQuotientField

```
getQuotientField() → Field
```

### 1.1.2.4 issubring

```
issubring(other: Ring) → bool
```

### 1.1.2.5 issuperring

```
issuperring(other: Ring) → bool
```

### 1.1.2.6 unnest

```
unnest() → RationalFunctionField
```

If `self` is a nested `RationalFunctionField` i.e. its coefficient field is also a `RationalFunctionField`, then the method returns one level unnested `RationalFunctionField`.  
For example:

## Examples

```
>>> RationalFunctionField(RationalFunctionField(Q, 1), 1).unnest()  
RationalFunctionField(Q, 2)
```

### 1.1.2.7 gcd

```
gcd(a: RationalFunction, b: RationalFunction) → RationalFunction
```

Inherited from `Field`.

**1.1.2.8 isdomain**

**1.1.2.9 iseclidean**

**1.1.2.10 isnoetherian**

**1.1.2.11 ispid**

**1.1.2.12 isufd**

Inherited from **CommutativeRing**.

### 1.1.3 PolynomialIdeal – ideal of polynomial ring

A subclass of **Ideal** represents ideals of polynomial rings.

#### Initialize (Constructor)

**PolynomialIdeal**(generators: *list*, polyring: *PolynomialRing*)  
→ *PolynomialIdeal*

Create an object represents an ideal in a polynomial ring `polyring` generated by `generators`.

#### Operations

operator	explanation
<code>in</code>	membership test
<code>==</code>	same ideal?
<code>!=</code>	different ideal?
<code>+</code>	addition
<code>*</code>	multiplication

## Methods

### 1.1.3.1 reduce

`reduce(element: polynomial) → polynomial`

Modulo `element` by the ideal.

### 1.1.3.2 issubset

`issubset(other: set) → bool`

### 1.1.3.3 issuperset

`issuperset(other: set) → bool`