

# Contents

<b>1</b>	<b>Classes</b>	<b>2</b>
1.1	poly.multivar – multivariate polynomial . . . . .	2
1.1.1	PolynomialInterface – base class for all multivariate polynomials . . . . .	3
1.1.2	BasicPolynomial – basic implementation of polynomial . . . . .	3
1.1.3	TermIndices – Indices of terms of multivariate polynomials . . . . .	3
1.1.3.1	pop . . . . .	4
1.1.3.2	gcd . . . . .	4
1.1.3.3	lcm . . . . .	4

# Chapter 1

## Classes

### 1.1 poly.multivar – multivariate polynomial

- Classes
  - †**PolynomialInterface**
  - †**BasicPolynomial**
  - **TermIndices**

### 1.1.1 PolynomialInterface – base class for all multivariate polynomials

Since the interface is an abstract class, do not instantiate.

### 1.1.2 BasicPolynomial – basic implementation of polynomial

Basic polynomial data type.

### 1.1.3 TermIndices – Indices of terms of multivariate polynomials

It is a tuple-like object.

#### Initialize (Constructor)

`TermIndices(indices: tuple) → TermIndices`

The constructor does not check the validity of indices, such as integerness, nonnegativity, etc.

#### Operations

operator	explanation
<code>t == u</code>	equality
<code>t != u</code>	inequality
<code>t + u</code>	(componentwise) addition
<code>t - u</code>	(componentwise) subtraction
<code>t * a</code>	(componentwise) multiplication by scalar <code>a</code>
<code>t &lt;= u, t &lt; u, t &gt;= u, t &gt; u</code>	ordering
<code>t[k]</code>	k-th index
<code>len(t)</code>	length
<code>iter(t)</code>	iterator
<code>hash(t)</code>	hash

## Methods

### 1.1.3.1 pop

**pop(self, pos: *integer*) → (*integer*, *TermIndices*)**

Return the index at pos and a new TermIndices object as the omitting-the-pos indices.

### 1.1.3.2 gcd

**gcd(self, other: *TermIndices*) → *TermIndices***

Return the “gcd” of two indices.

### 1.1.3.3 lcm

**lcm(self, other: *TermIndices*) → *TermIndices***

Return the “lcm” of two indices.

# Bibliography