Contents

1 Classes **2**

1.1 round2 – the round 2 method 2

1.1.1 ModuleWithDenominator – bases of $\mathbb{Z}$-module with denominator. 3

1.1.1.1 get_rationals – get the bases as a list of rationals 4

1.1.1.2 get_polynomials – get the bases as a list of polynomials 4

1.1.1.3 determinant – determinant of the bases 4

1.1.2 round2(function) 5

1.1.3 Dedekind(function) 5

Chapter 1

Classes

1.1 round2 – the round 2 method

- **Classes**
  - **ModuleWithDenominator**
- **Functions**
  - **round2**
  - **Dedekind**

The round 2 method is for obtaining the maximal order of a number field from an order generated by a root of a defining polynomial of the field.

This implementation of the method is based on [**?**](Algorithm 6.1.8) and [**?**](Chapter 3).

1.1.1 ModuleWithDenominator – bases of $\mathbb{Z}$-module with denominator.

Initialize (Constructor)

ModuleWithDenominator(`basis`: ***list***, `denominator`: ***integer***, `**hints`: ***dict***)
→ ***ModuleWithDenominator***

This class represents bases of $\mathbb{Z}$-module with denominator. It is not a general purpose $\mathbb{Z}$-module, you are warned. `basis` is a list of integer sequences.
`denominator` is a common denominator of all bases.
†Optionally you can supply keyword argument `dimension` if you would like to postpone the initialization of `basis`.

Operations

operator	explanation
`A + B`	sum of two modules
`a * B`	scalar multiplication
`B / d`	divide by an integer

Methods

1.1.1.1 get_rationals – get the bases as a list of rationals

get_rationals(`self`) → ***list***

Return a list of lists of rational numbers, which is bases divided by denominator.

1.1.1.2 get_polynomials – get the bases as a list of polynomials

get_polynomials(`self`) → ***list***

Return a list of rational polynomials, which is made from bases divided by denominator.

1.1.1.3 determinant – determinant of the bases

determinant(`self`) → ***list***

Return determinant of the bases (bases ought to be of full rank and in Hermite normal form).

1.1.2 round2(function)

round2(`minpoly_coeff`: ***list***) → (***list***, ***integer***)

Return integral basis of the ring of integers of a field with its discriminant. The field is given by a list of integers, which is a polynomial of generating element θ. The polynomial ought to be monic, in other word, the generating element ought to be an algebraic integer.

The integral basis will be given as a list of rational vectors with respect to θ.

1.1.3 Dedekind(function)

Dedekind(`minpoly_coeff`: ***list***, `p`: ***integer***, `e`: ***integer***)
→ (***bool***, ***ModuleWithDenominator***)

This is the Dedekind criterion.

`minpoly_coeff` is an integer list of the minimal polynomial of θ.
`p**e` divides the discriminant of the minimal.
The first element of the returned tuple is whether the computation about `p` is finished or not.