Contents

Chapter 1

Functions

1.1 poly.groebner – Gröbner Basis

The groebner module is for computing Gröbner bases for multivariate polynomial ideals.

This module uses the following types:

polynomial :
`polynomial` is the polynomial generated by function poly.multiutil.polynomial.

order :
`order` is the order on terms of polynomials.

1.1.1 buchberger – naïve algorithm for obtaining Gröbner basis

buchberger(`generating`: *list*, `order`: *order*) → *[polynomials]*

Return a Gröbner basis of the ideal generated by given generating set of polynomials with respect to the order.
The argument `generating` is a list of poly.multiutil.Polynomial; the argument `order` is an order.
Be careful, this implementation is very naive.

1.1.2 normal_strategy – normal algorithm for obtaining Gröbner basis

normal_strategy(`generating`: *list*, `order`: *order*) → *[polynomials]*

Return a Gröbner basis of the ideal generated by given generating set of polynomials with respect to the order. This function uses the 'normal strategy'.
The argument `generating` is a list of poly.multiutil.Polynomial; the argument

`order` is an order.

1.1.3 reduce_groebner – reduce Gröbner basis

reduce_groebner(`gbasis`: ***list***, `order`: ***order***) → ***[polynomials]***

Return the reduced Gröbner basis constructed from a Gröbner basis. It satisfies that:

- $\mathrm{lb}(f)$ divides $\mathrm{lb}(g) \Rightarrow g$ is not in reduced Gröbner basis, and
- monic.

The argument `gbasis` is a list of polynomials, a Gröbner basis.

1.1.4 s_polynomial – S-polynomial

s_polynomial(`f`: ***polynomial***, `g`: ***polynomial***, `order`: ***order***)
→ ***[polynomials]***

Return S-polynomial of `f` and `g` with respect to the `order`.

$$S(f,g) = (\mathrm{lc}(g) * T/\mathrm{lb}(f)) * f - (\mathrm{lc}(f) * T/\mathrm{lb}(g)) * g,$$

where $T = \mathrm{lcm}(\mathrm{lb}(f), \mathrm{lb}(g))$.