Contents

Chapter 1

Functions

1.1 poly.factor – 多項式の因数分解

factor モジュールは整数係数一変数多項式の因数分解のためのもの.
　このモジュールは以下に示す型を使用:

polynomial :
　　polynomial は poly.uniutil.polynomial によって生成された多項式.

1.1.1 brute_force_search – 総当たりで因数分解を探す

brute_force_search(f: *poly.uniutil.IntegerPolynomial*, fp_factors: *list*, q: *integer*)
　　→ *[factors]*

　fp_factors 上でいくつかの積の組み合わせである因数を探すことにより f の因数分解を見つける. この組み合わせは総当たりで探される.
引数 fp_factors は poly.uniutil.FinitePrimeFieldPolynomial のリストです.

1.1.2 divisibility_test – 可除性テスト

divisibility_test(f: *polynomial*, g: *polynomial*) → *bool*

　多項式において,f が g で割り切れるかどうか,Boolean 値を返す.

1.1.3 minimum_absolute_injection – 係数を絶対値最小表現に渡す

minimum_absolute_injection(f: *polynomial*) → *F*

各係数を絶対値最小表現に渡す $\mathbf{Z}/p\mathbf{Z}$ 係数多項式 f の単射により整数係数多項式 F を返す.

与えられた多項式 f の係数環は **IntegerResidueClassRing** または **FinitePrimeField** でなければならない.

1.1.4 padic_factorization – p 進分解

padic_factorization(f: *polynomial*) → *p*, *factors*

素数 p と, 与えられた平方因子を含まない整数係数多項式 f の p 進分解を返す. 結果である factors は整数係数を持ち,$\mathbb{F}_p$ からその絶対値最小表現に写されている.

† 素数は以下のように選ばれる:

1. f mod p でも平方因子を持たない,
2. 因数の数は次の素数を超えない.

与えられた多項式 f は poly.uniutil.IntegerPolynomial でなければならない.

1.1.5 upper_bound_of_coefficient –Landau-Mignotte の係数の上界

upper_bound_of_coefficient(f: *polynomial*) → *long*

次数は与えられた f の次数の半分を超えない大きさである Landau-Mignotte の因数の係数の上界を計算.

与えられた多項式 f は整数係数多項式でなければならない.

1.1.6 zassenhaus – Zassenhaus 法による平方因子のない整数係数多項式の因数分解

zassenhaus(f: *polynomial*) → *list of factors f*

Berlekamp-Zassenhaus 法による平方数のない整数係数の多項式 f の因数.

1.1.7 integerpolynomialfactorization – 整数多項式の因数分解

integerpolynomialfactorization(f: *polynomial*) → *factor*

Berlekamp-Zassenhaus 法により整数係数多項式 f を因数分解.

因数は (factor, index) という形式のタプルのリストの形式で出力される.

Bibliography