Using LLL's Algorithm to factor Polynomials in Q[X]

Hieu Ha Van, Lambert Rosique and Thieu Vo Ngoc

Arithm. Algo. course's Presentation

22 November 2012

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

- Calculate $R(f, f')$ *using "subresultant algorithm"*

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

- Calculate $R(f, f')$ *using "subresultant algorithm"*

a) If $R(f, f') = 0$ *then let* $g = gcd(f, f')$

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

- Calculate $R(f, f')$ *using "subresultant algorithm"*

a) If $R(f, f') = 0$ *then let* $g = gcd(f, f')$

Replace $f \leftarrow \frac{f}{g}$

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

- Calculate $R(f, f')$ *using "subresultant algorithm"*

a) If $R(f, f') = 0$ *then let* $g = gcd(f, f')$

Replace $f \leftarrow \frac{f}{g}$ *so that* $R(f, f') \neq 0$

Main Algorithm

Algorithm

INPUT: *Primitive polynomial* $f \in Z[X]$ *of degree* $n > 0$

OUTPUT: *Irreducible factors of* f *in* $Z[X]$

1st step :

- Calculate $R(f, f')$ *using "subresultant algorithm"*

a) If $R(f, f') = 0$ *then let* $g = gcd(f, f')$

Replace $f \leftarrow \frac{f}{g}$ *so that* $R(f, f') \neq 0$

b) Else, go to step 2.

Algorithm

2nd step :

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

b) Apply Berlekamp's algo. $\Longrightarrow$ *irreducible factors of* $(fmodp)$ *in* F_p

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

b) Apply Berlekamp's algo. $\Longrightarrow$ *irreducible factors of* $(fmodp)$ *in* F_p

Remark

$R(f, f') = \pm a_n(f) \cdot \Delta(f) \neq 0 modp$

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

b) Apply Berlekamp's algo. $\Longrightarrow$ *irreducible factors of* $(fmodp)$ *in* F_p

Remark

$R(f, f') = \pm a_n(f) \cdot \Delta(f) \neq 0 modp$ *so* $deg(fmodp) = n$

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

b) Apply Berlekamp's algo. $\Longrightarrow$ *irreducible factors of* $(fmodp)$ *in* F_p

Remark

$R(f, f') = \pm a_n(f) \cdot \Delta(f) \neq 0 modp$ *so* $deg(fmodp) = n$ *and* f *has no multiple factors in* F_p

Algorithm

2nd step :

a) Find $p \nmid R(f, f')$ *smallest in* P.

b) Apply Berlekamp's algo. $\Longrightarrow$ *irreducible factors of* $(fmodp)$ *in* F_p

Remark

$R(f, f') = \pm a_n(f) \cdot \Delta(f) \neq 0 modp$ *so* $deg(fmodp) = n$ *and* f *has no multiple factors in* F_p *Consequence :* (2.4) *holds for all irreducible factor of* $(fmodp)$ *in* $F_p[X]$

Algorithm

3rd step :

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

a) Start with $f_1 = 1$ *and* $f_2 = f$

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

a) Start with $f_1 = 1$ *and* $f_2 = f$

b) While $f_2 \neq \pm 1$ *do :*

α. *Choose an irreducible factor* $(h mod p)$ *of* $(f_2 mod p)$, *reduced,* $a_{deg(h)}(h) = 1$

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

a) Start with $f_1 = 1$ *and* $f_2 = f$

b) While $f_2 \neq \pm 1$ *do :*

α. *Choose an irreducible factor* $(hmodp)$ *of* $(f_2 modp)$, *reduced,* $a_{deg(h)}(h) = 1$

β. *Apply Sub-Algorithm 1 to find irred. factor* h_0 *of* f_2 *in* $Z[X]$ *st :*

$(hmodp) \mid (h_0 modp)$

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

a) Start with $f_1 = 1$ *and* $f_2 = f$

b) While $f_2 \neq \pm 1$ *do :*

α. *Choose an irreducible factor* $(hmodp)$ *of* $(f_2 modp)$, *reduced,* $a_{deg(h)}(h) = 1$

β. *Apply Sub-Algorithm 1 to find irred. factor* h_0 *of* f_2 *in* $Z[X]$ *st :*
$(hmodp) \mid (h_0 modp)$

γ. $f_1 \longleftarrow f_1 \cdot h_0$ *and* $f_2 \longleftarrow \frac{f_2}{h_0}$

Algorithm

3rd step :

Let $f = f_1 \cdot f_2$ *in* $Z[X]$ *with* f_1 *in* $Z[X]$ *and* $f_2 mod p$ *in* $F_p[X]$ *completely factorized*

a) Start with $f_1 = 1$ *and* $f_2 = f$

b) While $f_2 \neq \pm 1$ *do :*

α. *Choose an irreducible factor* $(h mod p)$ *of* $(f_2 mod p)$, *reduced,* $a_{deg(h)}(h) = 1$

β. *Apply Sub-Algorithm 1 to find irred. factor* h_0 *of* f_2 *in* $Z[X]$ *st :*
$(h mod p) \mid (h_0 mod p)$

γ. $f_1 \longleftarrow f_1 \cdot h_0$ *and* $f_2 \longleftarrow \frac{f_2}{h_0}$

δ. *Delete irreducible factors of* $(f_2 mod p)$ *dividing* $(h_0 mod p)$

Sub-Algorithm 1

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced and satisfying hypothesis for* $k = 1$

Sub-Algorithm 1

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$, *of degree* $n > 0$, p *prime,* $h \in Z[X]$ *reduced and satisfying hypothesis for* $k = 1$

OUTPUT: h_0 *irreducible factor of f st* $(h \bmod p) \mid (h_0 \bmod p)$

Sub-Algorithm 1

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$, *of degree* $n > 0$, p *prime,* $h \in Z[X]$ *reduced and satisfying hypothesis for* $k = 1$

OUTPUT: h_0 *irreducible factor of f st* $(h \bmod p) \mid (h_0 \bmod p)$

1st step :

Sub-Algorithm 1

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced and satisfying hypothesis for* $k = 1$

OUTPUT: h_0 *irreducible factor of f st* $(h \bmod p) \mid (h_0 \bmod p)$

1st step :

a) While $h_0 \neq f$ *do*

α. *Find least* $k \in N$ *st :* $p^{kl} > 2^{(n-1)n/2} \cdot \binom{2(n-1)}{n-1}^{n/2} \cdot |f|^{2n-1}$

Sub-Algorithm 1

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced and satisfying hypothesis for* $k = 1$

OUTPUT: h_0 *irreducible factor of f st* $(h \bmod p) \mid (h_0 \bmod p)$

1st step :

a) While $h_0 \neq f$ *do*

α. *Find least* $k \in N$ *st :* $p^{kl} > 2^{(n-1)n/2} \cdot \binom{2(n-1)}{n-1}^{n/2} \cdot |f|^{2n-1}$

β. *Use Hensel's lemma to modify h without changing* $(h \bmod p)$ *st* (2.2) *still holds for this k*

Algorithm

γ. *Let* $u \in N$ *greatest st :* $l \leq (n-1)/2^u$

Algorithm

γ. *Let* $u \in N$ *greatest st :* $l \leq (n-1)/2^u$
For i *from* u *down to* 0 *do :*

Algorithm

γ. *Let* $u \in N$ *greatest st :* $l \leq (n-1)/2^u$
For i *from* u *down to* 0 *do :*
Apply Sub-Algo 2 with $m = \lfloor (n-1)/2^i \rfloor$

Algorithm

γ. *Let* $u \in N$ *greatest st :* $l \leq (n-1)/2^u$
For i *from* u *down to* 0 *do :*
Apply Sub-Algo 2 with $m = \lfloor (n-1)/2^i \rfloor$
If it succeeds in determining h_0, *break*

Algorithm

γ. *Let* $u \in N$ *greatest st :* $l \leq (n-1)/2^u$
For i *from* u *down to* 0 *do :*
Apply Sub-Algo 2 with $m = \lfloor (n-1)/2^i \rfloor$
If it succeeds in determining h_0, *break*
Else, $deg(h_0) > n-1$ *so* $h_0 = f$.

Sub-Algorithm 2

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced (mod* p^k*) and satisfying hypothesis*

Sub-Algorithm 2

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced (mod* p^k*) and satisfying hypothesis*

OUTPUT: *Answer to "Is* $\deg(h_0) \leq m$*?" plus* h_0 *if yes,*

Sub-Algorithm 2

Algorithm

INPUTS: *f primitive polynomial in* $Z[X]$*, of degree* $n > 0$*, p prime,* $h \in Z[X]$ *reduced (mod* p^k*) and satisfying hypothesis*

OUTPUT: *Answer to "Is* $deg(h_0) \leq m$*?" plus* h_0 *if yes,*

for h_0 *as in proposition, an* $m \geq deg(h) = l$ *given st :*
$p^{kl} > 2^{mn/2} \cdot \binom{2m}{m}^{n/2} \cdot |f|^{m+n}$

Algorithm

a) Let L lattice as before, of basis
$\{p^k X^i : 0 \leq i < l\} \cup \{hX^j : 0 \leq j \leq m - l\}$
Find a reduced basis $b_1, b_2, ..., b_{m+1}$

Algorithm

a) Let L lattice as before, of basis
$\{p^k X^i : 0 \leq i < l\} \cup \{hX^j : 0 \leq j \leq m - l\}$
Find a reduced basis $b_1, b_2, ..., b_{m+1}$

b) If $|b_1| \geq (p^{kl}/|f|^m)^{1/n}$ *then* $deg(h_0) > m$, *break*

Algorithm

a) Let L lattice as before, of basis $\{p^k X^i : 0 \leq i < l\} \cup \{hX^j : 0 \leq j \leq m - l\}$
Find a reduced basis $b_1, b_2, ..., b_{m+1}$

b) If $|b_1| \geq (p^{kl}/|f|^m)^{1/n}$ *then* $deg(h_0) > m$, *break*

c) Else then $deg(h_0) \leq m$ *and* $h_0 = gcd(b_1, ..., b_t)$ *(with t as seen before)*
Just calculate h_0 *by repetedly applying Subresultant algorithm.*