Multiplication of Univariate Polynomials Over Algebraic Number Fields

Cory Ahn and Michael Monagan, Simon Fraser University

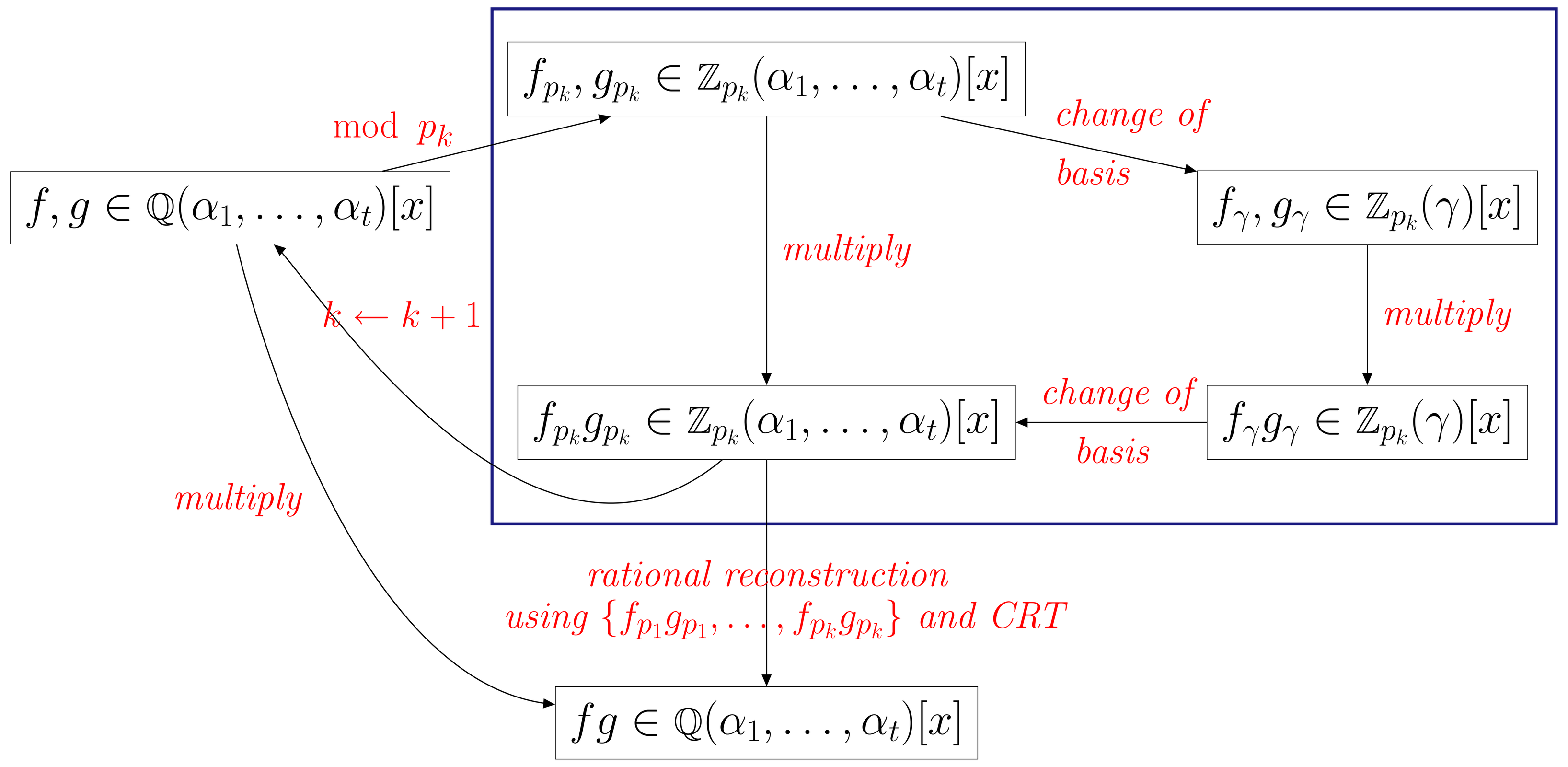
Motivation

Let $f(x)$ and $g(x)$ be dense polynomials in $K[x]$ where $K = \mathbb{Q}(\alpha_1, \ldots, \alpha_t)$ is an algebraic number field and each $\alpha_i \notin \mathbb{Q}(\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_t)$.

How can we compute $h(x) = f(x) \cdot g(x)$ efficiently?

Overview of Strategy

Initialize k to 1.



In this poster we will only be concerned with the algorithms performed within the blue box above.

Representing $f \in K(\alpha_1, \ldots, \alpha_t)[x]$

Fact. $K(\alpha_1, \ldots, \alpha_t) \cong K[u_1, \ldots, u_t]/\langle m_1, \ldots, m_t \rangle$, where $m_i := m_i(u_i)$ is the minimal polynomial for α_i over K for each $i = 1, \ldots, t$.

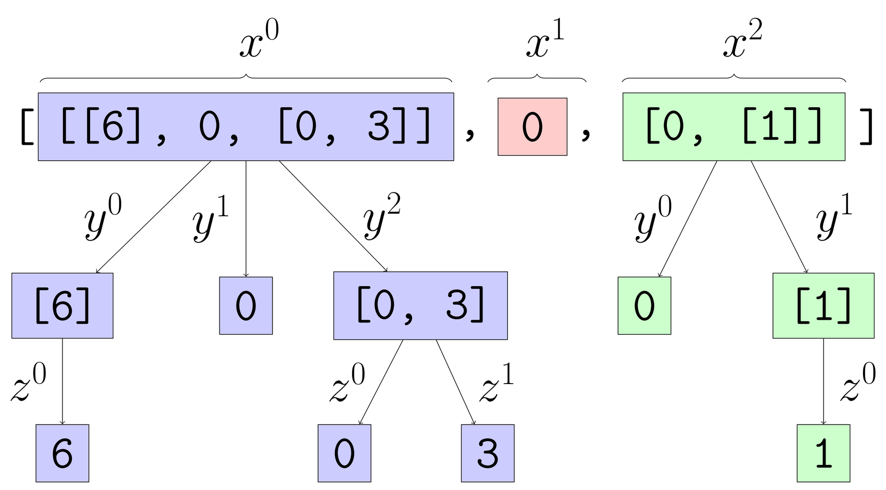
So we consider f and g as $(t+1)$-variate polynomials in $K[u_1, \ldots, u_t][x]/\langle m_1, \ldots, m_t \rangle$.

We also need to choose a data structure to represent the polynomials. We will use a **recursive dense** data structure (`recden` in Maple).

Example.

$$f(x, y) = 13 - 4y^2 z + 8x^2 y \in \mathbb{Z}_7[z][x, y]/\langle z^2 + 5 \rangle \text{ with } x >_{lex} y >_{lex} z$$

$$\equiv \left(6y^0 + 0y^1 + (0z^0 + 3z^1)\, y^2\right) x^0 + 0\, x^1 + \left(0y^0 + 1y^1\right) x^2 \mod 7$$



Naïve Multiplication Strategy

- Convert $f, g \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$ to `recden` polynomials $F, G \in \mathbb{Q}[u_1, \ldots, u_t][x]/\langle m_1, \ldots, m_t \rangle$ and multiply F and G "naively".

Let $d = \deg(\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t))$, and let $\deg_x(f), \deg_x(g) \le n$.
This takes $\mathcal{O}(n^2 d^2)$ arithmetic operations in $\mathbb{Q}$. Slow!

There are efficiency problems associated with this strategy.

Problem 1:
More variables in polynomial = more complicated `recden` data structure

Example. $f := a + b + c + d + e \in \mathbb{Z}_7[a, b, c, d, e]$ in `recden` data structure is:
`[[[[[0, 1], [1]], [[1]]], [[[1]]]], [[[[1]]]]]`.

Solution: Map $K(\alpha_1, \ldots, \alpha_t)$ to $K(\gamma)$.

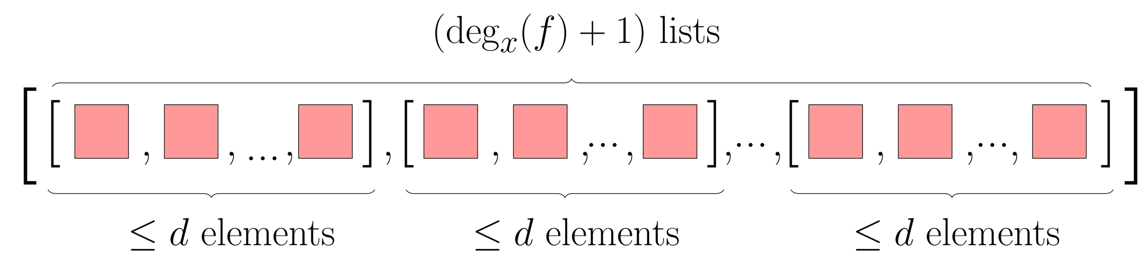
How to find γ? Let $\alpha_2, \ldots, \alpha_m$ be the conjugates of $\alpha (= \alpha_1)$ and let $\beta_2, \ldots, \beta_n$ be the conjugates of $\beta (= \beta_1)$.

$$\text{Let } S = \left\{ \frac{\alpha_r - \alpha_s}{\beta_t - \beta_u} : r, s \in \{1, \ldots, m\}, t, u \in \{1, \ldots, n\}, t \neq u \right\}.$$

Pick $c \in K \backslash S$. Then $K(\alpha, \beta) \cong K(\gamma := \alpha + c\beta)$.

We can generalize this to express the (t+1)-variate polynomial f in $\mathbb{Z}_p[u_1, \ldots, u_t][x]/\langle m_1, \ldots, m_t \rangle$ as a bivariate polynomial in $\mathbb{Z}_p[z][x]/\langle m_\gamma(z) \rangle$.
$\Rightarrow$ **simpler `recden` data structure!**

In fact, f has the form:



Problem 2: The multiplication is too slow: $\mathcal{O}(n^2 d^2)$

Solution: Use the fast Fourier Transform (FFT) to multiply the two bivariate polynomials: $\mathcal{O}(nd^2 + dn \log_2 n)$

A Faster Multiplication Strategy

1. Choose p, a prime.
2. Convert $f_p, g_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$ to $f_\gamma, g_\gamma \in \mathbb{Z}_p(\gamma)[x]$. $\longleftarrow \mathcal{O}(d^3 + nd^2)$
3. Multiply f_γ and $g_\gamma \in \mathbb{Z}_p[z][x]/\langle m_\gamma(z) \rangle$ using the FFT. $\longleftarrow \mathcal{O}(nd^2 + dn \log_2 n)$
4. Convert the product to a polynomial in $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$. $\longleftarrow \mathcal{O}(nd^2)$

Thus the overall cost of this strategy is $\mathcal{O}(d^3 + nd^2 + dn \log_2 n)$.
This is a considerable improvement over the naive strategy, especially for large n.

$\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t) \longrightarrow \mathbb{Z}_p(\gamma)$

We must find $\{c_2, \ldots, c_t\} \subset \mathbb{Z}_p$ so that $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t) \cong \mathbb{Z}_p(\alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)$.

Theorem. If we randomly choose a set of numbers $\chi := \{c_2, \ldots, c_t\} \subset \mathbb{Z}_p$, then the probability of choosing the "unlucky" χ such that $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t) \ncong \mathbb{Z}_p(\gamma = \alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)$ is less than $\frac{td^2}{p}$.

We use large p, so $\frac{td^2}{p}$ will be small. As such, we will pick $c_2, \ldots, c_t \in \mathbb{Z}_p$ at random.

Lemma. Let $\gamma = \alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t$ be such that $\mathbb{Z}_p(\gamma) \cong \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)$. Then

$$B_1 := \{1, \gamma, \gamma^2, \ldots, \gamma^{d-1}\} \text{ is a basis for } \mathbb{Z}_p(\gamma) \text{ and}$$

$$B_2 := \{\alpha_1^{j_1} \alpha_2^{j_2} \cdots \alpha_t^{j_t}, j_i = 0, 1, \ldots, d_i - 1\} \text{ is a basis for } \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t).$$

Thus we will build a $d \times d$ matrix C so that C is a change-of-basis matrix from B_1 to B_2 and C^{-1} is a change-of-basis matrix from B_2 to B_1.

Choosing the "Right" Prime

We choose our prime p such that

- C is invertible in $\mathbb{Z}_p$.
- p is between 2^{30} and $2^{31.5}$, so that all numbers arising from our algorithm can be mltiplied on a 64-bit machine without overflow.
- p is a Fourier prime (i.e. prime of form $k \cdot 2^r + 1$, k odd and $r \ge R$, where 2^R is the smallest power of two greater than $\deg_x(f) + \deg_x(g)$).

Lemma. Of all Fourier primes between 2^{30} and $2^{31.5}$ for a given $N = 2^R > \deg_x(f) + \deg_x(g)$, the probability that a Fourier prime divides $\det(C)$ is at most

$$\frac{(d/2 + R \cdot d)\, 2^R}{8.7459 \times 10^8}.$$

Since $d, 2^R \ll 8.7459 \times 10^8$ we pick a random Fourier prime in $(2^{30}, 2^{31.5})$ as our p.

Benchmarks

Let $f(x), g(x) \in \mathbb{Z}_p(\alpha_1, \alpha_2, \alpha_3)[x]$ where $n = \deg_x(f) = \deg_x(g)$, and the coefficients of f and g are chosen at random from $\mathbb{Z}_p$. $\alpha_1 = \sqrt{111}, \alpha_2 = \sqrt{131}$ and $\alpha_3 = \sqrt{171}$.

	$\mathbb{Z}_p(\alpha_1, \alpha_2, \alpha_3)[x]$			$\mathbb{Z}_p(\gamma)[x]$		
n	naive mult	FFT mult.	conversion 1	naive mult.	FFT mult.	conversion 2
12	0.146	0.074	0.013	0.003	0.010	0.003
24	0.541	0.152	0.017	0.008	0.024	0.005
48	2.096	0.344	0.024	0.032	0.054	0.010
96	8.207	0.770	0.045	0.128	0.123	0.019
192	32.533	1.704	0.096	0.471	0.293	0.039
384	129.620	3.767	0.252	1.908	0.693	0.078

Here $\alpha_1, \alpha_2, \alpha_3$ be algebraic numbers of degree 4 each.

	$\mathbb{Z}_p(\alpha_1, \alpha_2, \alpha_3)[x]$			$\mathbb{Z}_p(\gamma)[x]$		
n	naive mult.	FFT mult.	conversion 1	naive mult.	FFT mult.	conversion 2
12	1.408	0.562	0.390	0.058	0.028	0.083
24	4.987	1.191	0.505	0.207	0.067	0.224
48	18.262	2.506	0.881	0.782	0.151	0.386
96	70.710	5.279	1.626	3.119	0.367	0.778
192	280.522	11.258	4.037	12.350	0.836	1.566