9 The resultant and a modular gcd algorithm in $\mathbb{Z}[x]$

Let F be a field. Then the ring $\mathsf{F}[x]$ of polynomials is a unique factorization domain (UFD), so greatest common divisors exist. Not only is $\mathsf{F}[x]$ a UFD, it a Euclidean domain, so gcds can be computed with the Euclidean algorithm.

But what about $\mathbb{Z}[x]$? Because $\mathbb{Z}[x]$ is not a Euclidean domain the Euclidean algorithm cannot be applied directly. Do gcds over $\mathbb{Z}[x]$ even exist? It turns out that the answer is yes. But then some natural questions arise. How can we compute gcds over $\mathbb{Z}[x]$? What is the relationship of gcds over $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$? This script gives answers to these questions.

Subsection 9.1 and 9.2 develop some necessary mathematical background. The last subsection gives an efficient modular algorithms for computing gcds over $\mathbb{Z}[x]$. Because of the established relationship between factorization over $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$ in §9.1, the modular algorithm for gcd over $\mathbb{Z}[x]$ will also be useful for gcd computation over $\mathbb{Q}[x]$.

9.1 Gauss' lemma and theorem

To begin we need to define some notation. Let R be a UFD. Recall that a unit of R is an invertible element, and that two elements $a, b \in \mathsf{R}$ are associates if $a = ub$ for $u \in \mathsf{R}$ a unit. Over $\mathbb{Z}$ the only units are ± 1, while over $\mathsf{F}[x]$ the units are the nonzero constant polynomials, that is, elements of $\mathsf{F} \setminus \{0\}$. Gcds over R and $\mathsf{F}[x]$ are unique, but only up to units. To make gcds unique, we define a function lu and normal over R such that for any $a \in \mathsf{R}$ we have $a = \mathrm{lu}(a) \times \mathrm{normal}(a)$. An element $a \in \mathsf{R}$ is normalized if $a = \mathrm{normal}(a)$, or equivalently, if $\mathrm{lu}(a) = 1$. For all $a, b \in \mathsf{R}$, by $\gcd(a, b)$ we mean the unique normalized gcd of a and b. Over $\mathbb{Z}$ we define $\mathrm{lu}(a) = \mathrm{sign}(a)$, so gcds over $\mathbb{Z}$ are positive; while over $\mathsf{F}[x]$ we define $\mathrm{lu}(a) = \mathrm{lc}(a)$, so gcds over $\mathsf{F}[x]$ are monic. By convention, $\mathrm{lu}(0) = 1$ and $\mathrm{normal}(0) = 0$.

Now let $f = f_0 + f_1 x + \cdots + f_n x^n \in \mathsf{R}[x]$, R a UFD. The *content* $\mathrm{cont}(f)$ is defined as $\mathrm{cont}(f) = \gcd(f_0, \ldots, f_n) \in \mathsf{R}$. By convention, $\mathrm{cont}(f_0) = \gcd(f_0) = \mathrm{normal}(f_0)$. The *primitive part* $\mathrm{pp}(f)$ of f is defined by $f = \mathrm{cont}(f) \cdot \mathrm{pp}(f)$. A polynomial $f \in \mathsf{R}[x]$ is primitive if $\mathrm{cont}(f) = 1$

Example 9.1. *Let* $f = 18x^3 - 42x^2 + 30x - 6$. *Then cont*$(f) = \gcd(18, -42, 30, -6) = 6$ *and pp*$(f) = 3x^2 - 7x^2 + 30x - 6$.

It is useful to extend the notion of content to polynomials in $\mathsf{F}[x]$. If $f = (a_0/b) + (a_1/b)x + \cdots + (a_n/b)x^n \in \mathsf{F}[x]$ for a common denominator b, then $\mathrm{cont}(f) = \gcd(a_0, \ldots, a_n)/\mathrm{cont}(b) \in \mathsf{F}$, and $\mathrm{pp}(f) = f/\mathrm{cont}(f)$. With this definition, $\mathrm{pp}(f)$ will be a primitive polynomial in $\mathsf{R}[x]$.

Example 9.2. *cont*$((2/3)x + 1/2) = 1/6$ *and pp*$((2/3)x + 1/2) = 4x + 3$.

If R is a UFD, the following fundamental theorem guarantees that $\mathsf{R}[x]$ is also a UFD, and fully exposes the relationship between the factorization of polynomials in $\mathsf{R}[x]$ and $\mathsf{F}[x]$, where F is the fraction field of R.

Theorem 9.3. Gauss *Let* R *be a UFD. Then the following hold.*

- *The product of two primitive polynomials in* $\mathsf{R}[x]$ *is primitive.*
- *For* $f,g \in \mathsf{R}[x]$, $cont(fg) = cont(f)\cdot cont(g)$ *and* $pp(fg) = pp(f)\cdot pp(g)$.
- $\mathsf{R}[x]$ *is UFD, and the unique factorization (up to units and ordering) of an* $f \in \mathsf{R}[x]$ *is given by*

$$f = \overbrace{p_1 p_2 \cdots p_k}^{cont(f)} \cdot \overbrace{pp(f_1)\, pp(f_2) \cdots pp(f_r)}^{pp(f)},$$

where $p_1 p_2 \cdots p_k$ *is the factorization over* R *of the content of* f*, and* $f_1 f_2 \cdots f_r$ *is the factorization over* $\mathsf{F}[x]$ *of the primitive part of* f.

As a corollary of Theorem 9.3, since $\mathsf{R}[x]$ is a UFD, any two elements of $\mathsf{R}[x]$ have a gcd. To make gcds in $\mathsf{R}[x]$ unique, we extend lu to $f \in \mathsf{R}[x]$ by $\mathrm{lu}(f) = \mathrm{lu}(\mathrm{lc}(f))$. Then $f = \mathrm{lu}(f)\cdot \mathrm{normal}(f)$, where $\mathrm{normal}(f)$ has a normalized leading coefficient from R. As a corollary of Theorem 9.3, given primitive polynomials $f,g \in \mathbb{Z}[x]$, we know the their gcd h over $\mathbb{Z}[x]$ will also be primitive, and we can compute h by passing over $\mathbb{Q}[x]$ as follows:

$$h := \underset{\mathbb{Z}[x]}{\gcd}(f,g) = \mathrm{pp}(\underset{\mathbb{Q}[x]}{\gcd}(f,g)) \tag{1}$$

The following algorithm modifies this recipe slightly by first scaling the gcd over $\mathbb{Q}[x]$, which may have rational number coefficients, by $\gcd(\mathrm{lc}(f),\mathrm{lc}(g))$, which is guaranteed to clear the denominators.

$$h := \underset{\mathbb{Z}[x]}{\gcd}(f,g) = \mathrm{pp}(\overbrace{\underset{\mathbb{Z}}{\gcd}(\mathrm{lc}(f),\mathrm{lc}(g)) \cdot \underset{\mathbb{Q}[x]}{\gcd}(f,g)}^{\in \mathbb{Z}[x]}) \tag{2}$$

Note that (1) and (2) only hold when $\gcd(f,g)$ is primitive. (A sufficient condition for $\gcd(f,g)$ to be primitive is that at least one of f and g be primitive.) Also, since $\gcd(\mathrm{lc}(f),\mathrm{lc}(g))$ may actually be a proper multiple of $\mathrm{lc}(h)$, we still need to take the primitive part in (2).

Algorithm: `PrimitiveGCD`
Input: ▸ $f,g \in \mathsf{R}[x]$ where R is a UFD and at least one of f and g is primitive.
Output: ▸ $\gcd(f,g) \in \mathsf{R}[x]$
(1) Compute the monic gcd $v \in \mathsf{F}[x]$ of f and g over $\mathsf{F}[x]$, where F is the field of fractions of R.
(2) $b \leftarrow \gcd(\mathrm{lc}(f),\mathrm{lc}(g))$
(3) Return $\mathrm{pp}(bv) \in \mathsf{R}[x]$

Example 9.4. *Let* $f = 18x^3 - 42x^2 + 30x - 6 \in \mathbb{Z}[x]$ *and* $g = -12x^2 + 10x - 2 \in \mathbb{Z}[x]$. *Then*

$$f = cont(f)\cdot pp(f) = 6\cdot(3x^3 - 7x^2 + 5x - 1)$$

and

$$g = cont(g)\cdot pp(f) = 2\cdot(-6x^2 + 5x - 1).$$

Over $\mathbb{Q}[x]$ we have

$$\gcd_{\mathbb{Q}[x]}(f,g) = \gcd_{\mathbb{Q}[x]}(pp(f),pp(g)) = x - 1/3 \in \mathbb{Q}[x].$$

Over $\mathbb{Z}[x]$ we have

$$\gcd_{\mathbb{Z}[x]}(f,g) = \gcd_{\mathbb{Z}}(cont(f),cont(g)) \cdot \gcd_{\mathbb{Z}[x]}(pp(f),pp(g)) = 2\cdot(3x-1) \in \mathbb{Z}[x].$$

Note that $\gcd_{\mathbb{Z}[x]}(pp(f),pp(g))$ is equal to $pp(\gcd_{\mathbb{Q}[x]}(f,g))$.

9.2 The resultant

Our goal will be to develop a modular algorithm for computing gcds over $\mathbb{Z}[x]$. The approach will be to choose a prime p and compute the gcd over $\mathbb{Z}_p[x]$ of the modular images of the polynomials. If the modular gcd is indeed an image of the gcd over $\mathbb{Z}[x]$, then the gcd over $\mathbb{Z}[x]$ can be recovered provided the prime p is large enough to capture the coefficients. But some primes are *bad*. The following example illustrates some subtleties with the approach.

Example 9.5. *Consider $f = 3x^3 + 3x - x^2 - 1$ and $g = 3x^2 + 5x - 2$ over $\mathbb{Z}[x]$. These are primitive polynomials with $h = \gcd(f,g) = 3x - 1 \in \mathbb{Z}[x]$. Consider the gcd of the modular images of f and g for the primes 3, 5 and 7.*

$$\begin{aligned}
\gcd(f \bmod 3, g \bmod 3) &= 1 \quad \textit{degree is too small} \\
\gcd(f \bmod 5, g \bmod 5) &= x^2+1 \quad \textit{degree is too large} \\
\gcd(f \bmod 7, g \bmod 7) &= x+2 \quad \textit{degree is correct}
\end{aligned}$$

If we multiply the monic gcd modulo 7 by the leading coefficient of the gcd over $\mathbb{Z}[x]$, and reduce in the symmetric range modulo 7, we obtain $3x + 6 \equiv 3x - 1 \bmod 7$.

As the last example illustrated, not all primes p are good primes in the sense that the gcd of the modular images of the polynomials may not be equal to the modular image of $h/\operatorname{lc}(h)$, where h is the gcd over $\mathbb{Z}$.

To get a handle on the bad primes we need to introduce the concept of the resultant. Let $f, g \in \mathsf{F}[x]$ be nonzero, $n = \deg f$, $m = \deg g$. Then $(-g)f + (f)g = 0$, but if we restrict the degrees of s and t in the equation $(s)f + (t)g = 0$, then the following lemma gives an interesting relationship between the existence of a solution to $sf + tg = 0$ and the existence of a nontrivial gcd of f and g.

Lemma 9.6. $\gcd(f,g) \neq 1$ *iff there exist nonzero $s,t \in \mathsf{F}[x]$ such that $sf + tg = 0$ with $\deg s < \deg g$ and $\deg t < \deg f$.*

Proof. (Only If) Suppose $\deg h = \deg \gcd(f,g) > 1$. Then we can choose $s = -g/h$ and $t = f/h$. (If) Assume $sf + tg = 0$ with $\gcd(f,g) = 1$ and $\deg t < \deg f$. Then $sf = -tg$ and $f \mid t$, which is impossible if $\deg t < \deg f$. □

Next, notice that polynomial multiplication is a linear map. For example, if $f = f_0 + f_1x + f_2x^2$ and $s = s_0 + s_1 + s_2x^2$, then the coefficient of the product $sf = u_0 + u_1x + \cdots + u_4x^4$ can be computed by a matrix$\times$vector product:

$$\begin{bmatrix} f_2 & & \\ f_1 & f_2 & \\ f_0 & f_1 & f_2 \\ & f_0 & f_1 \\ & & f_0 \end{bmatrix} \begin{bmatrix} s_2 \\ s_1 \\ s_0 \end{bmatrix} = \begin{bmatrix} u_4 \\ u_3 \\ u_2 \\ u_1 \\ u_0 \end{bmatrix}.$$

By extension, we can view the multiplication

$$\begin{bmatrix} f & g \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix}$$

in Lemma 9.6 as a linear map. We content ourselves with an explicit example.

Example 9.7. *Let* $f = 3x^3 - x^2 + 3x - 1$ *and* $g = 3x^2 + 5x - 2$. *Define* $s := s_1x + s_0$ *and* $t := t_2x^2 + t_1x + t_0$, *so that* $\deg s < \deg g$ *and* $\deg t < \deg f$. *The coefficient vector of* $sf + tg$ *is given by*

$$\overbrace{\left[\begin{array}{cc|ccc} 3 & & 3 & & \\ -1 & 3 & 5 & 3 & \\ 3 & -1 & -2 & 5 & 3 \\ -1 & 3 & & -2 & 5 \\ & -1 & & & -2 \end{array}\right]}^{Syl(f,g)} \begin{bmatrix} s_1 \\ s_0 \\ t_2 \\ t_1 \\ t_0 \end{bmatrix}.$$

In the above example, the matrix defining the linear map is square of dimension 5. In general, if $f, g \in \mathsf{R}[x]$ with $\deg f = n$ and $\deg g = m$, the Sylvester matrix $\mathrm{Syl}(f,g)$ of f and g is the square $(n+m) \times (n+m)$ matrix with first $\deg m$ columns comprised of shifts of the coefficient vector of f, and last n columns comprised of shifts of the coefficient vector of g.

Theorem 9.8. *Let* $f, g \in \mathsf{F}[x]$ *be nonzero.*

- $\gcd(f,g) = 1$ *iff* $Syl(f,g)$ *is invertible.*
- *If* $\gcd(f,g) = 1$ *and* $n + m \geq 1$, *then the EEA computes* $v \in \mathsf{F}^{n+m}$ *such that* $Syl(f,g)v$ *corresponds to the coefficient vector of the constant polynomial* 1.

Proof. The first part of the theorem follows as a corollary of Lemma 9.6. In particular, $\mathrm{Syl}(f,g)$ is invertible iff there does not exist a vector in the right nullspace of $\mathrm{Syl}(f,g)$; this is true iff there does not exist a solution to $sf + tg = 0$ with $\deg s < \deg g$ and $\deg t < \deg f$. For the second part, note that if $\mathrm{Syl}(f,g)$ is invertible, then the solution to $sf + tg = 1$ with $\deg s < \deg g$ and $\deg t < \deg f$ is unique. □

Definition 9.9. $res(f,g) = \det Syl(f,g)$.

By convention, if $n = m = 0$ then $\mathrm{Syl}(f,g)$ is the 0×0 matrix and $\mathrm{res}(f,g) = 1$. Also, $\mathrm{res}(f,0) = res(0,f) = 0$ if $f = 0$ or f is nonconstant.

Corollary 9.10. *Let $f,g \in \mathsf{F}[x]$. Then $\gcd(f,g) = 1$ iff $res(f,g) \neq 0$.*

Example 9.11. *Let $f = 3x^3 - x^2 + 3x - 1$ and $g = 3x^2 + 5x - 2$. Then $h = gcd(f,g) = 3x - 1 \in \mathbb{Z}[x]$. Since $\deg h > 0$ we have $res(f,g) \neq 0$, but*

$$res(f/h, g/h) = res(x^2+1, x+2) = \det Syl(f,g) = \begin{vmatrix} 1 & 1 & \\ 0 & 2 & 1 \\ 1 & & 2 \end{vmatrix} = 5.$$

So far, all discussion regarding $\mathrm{Syl}(f,g)$ and $\mathrm{res}(f,g)$ assumed f and g had coefficient from a field F. The case $\mathsf{F}[x]$ is mathematically simpler because we can use the language of vector spaces over fields for the description of the linear map given by $\mathrm{Syl}(f,g)$. In particular, $\mathrm{Syl}(f,g)$ is an isomorphism iff $\mathrm{Syl}(f,g)$ is invertible iff $\det \mathrm{Syl}(f,g) = \mathrm{res}(f,g) \neq 0$ iff there exist unique s and t in $\mathsf{F}[x]$ with $sf + tg = 1$, $\deg s < \deg g$, $\deg t < \deg f$. The following is a continuation of the previous example.

Example 9.12. *Let $f = x^2 + 1$ and $g = x + 2$. Define $s := s_0$ and $t := t_1 x + t_0$, so that $\deg s < \deg g$ and $\deg t < \deg f$. Considering f and g to live over $\mathbb{Q}[x]$, then the unique solution to $sf + tg = 1$ is given by*

$$\begin{bmatrix} s_0 \\ t_1 \\ t_0 \end{bmatrix} = \overbrace{\begin{bmatrix} 4/5 & -2/5 & 1/5 \\ 1/5 & 2/5 & -1/5 \\ -2/5 & 1/5 & 2/5 \end{bmatrix}}^{Syl(f,g)^{-1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Indeed, we have

$$\overbrace{\left(\frac{1}{5}\right)}^{s}(x^2+1) + \overbrace{\left(\frac{-1}{5}x + \frac{2}{5}\right)}^{t}(x+2) = 1.$$

But if $f,g \in \mathsf{R}[x]$, R a UFD, then $\mathrm{Syl}(f,g)$ and $\mathrm{res}(f,g)$ are well defined over R, and $\mathrm{res}(f,g)$ can tell us something about the degree of $\gcd(f,g)$ over R.

Corollary 9.13. *Let $f,g \in \mathsf{R}[x]$ be nonzero, R a UFD. Then $\gcd(f,g)$ is nonconstant in $\mathsf{R}[x]$ iff $res(f,g) = 0 \in \mathsf{R}$.*

The following theorem will provide the basis for our modular gcd algorithm over $\mathbb{Z}[x]$.

Theorem 9.14. *Let $f,g \in \mathbb{Z}[x]$. Suppose a prime p does not divide $b := \gcd(lc(f), lc(g))$. Then*

(i) $lc(\gcd_{\mathbb{Z}}(f,g)) \mid b$

(ii) $\deg(\gcd_{\mathbb{Z}_p[x]}(f \bmod p, g \bmod p)) \geq \gcd_{\mathbb{Z}[x]}(f,g)$

(iii) $\deg \gcd_{\mathbb{Z}_p[x]}(f \bmod p, g \bmod p)) = \deg(\gcd_{\mathbb{Z}[x]}(f,g))$

$$\iff lc(\underset{\mathbb{Z}[x]}{\gcd}(f,g)) \cdot \underset{\mathbb{Z}_p[x]}{\gcd}(f \bmod p, g \bmod p) \equiv \underset{\mathbb{Z}[x]}{\gcd}(f,g) \pmod p$$

$$\iff p \textit{ does not divide } res(f/h, g/h) \in \mathbb{Z}.$$

Example 9.15. *Consider* $f = 3x^3 + 3x - x^2 - 1$, $g = 3x^2 + 5x - 2$ *and* $h = \gcd(f,g) = 3x - 1 \in \mathbb{Z}[x]$ *from Example 9.5. We have* $b := \gcd(lc(f), lc(g)) = 3$*, so a priori we can infer nothing about* $\deg \gcd(f \bmod 3, g \bmod 3)$ *relative to* $\deg h$*. Since* $res(f/h, g/h) = 5$*, we know that* $\deg \gcd(f \bmod 5, g \bmod 5) > \deg h$*. Since* 7 *does not divide* $res(f,g)$*, we know that* $\deg \gcd(f \bmod 7, g \bmod 7) = \deg \gcd(f,g)$*, and, moreover, that* $\gcd(f \bmod 7, g \bmod 7) \in \mathbb{Z}_p[x]$ *will be the image of* $h/lc(h)$ *modulo 7.*

The idea for a modular algorithm to compute $\gcd(f,g)$ is now clear. Choose a prime p such that

- p does not divide $b := \gcd(\mathrm{lc}(f), \mathrm{lc}(g))$,
- p hopefully does not divide $\mathrm{res}(f/h, g/h)$, and
- coefficients of $(b/\alpha)\gcd(f,g)$ can be captured in the symmetric range modulo p.

To fill in the details we need to have a handle on the size of coefficients of factors of a polynomial over $\mathbb{Z}[x]$. Recall that $f = f_0 + f_1 x + \cdots + f_n x^n \in \mathbb{Z}[x]$ we have the following norms:

- $||f||_\infty = \max_i |f_i|$,
- $||f||_1 = \sum_i |f_i|$.

Theorem 9.16. *Suppose* $f, g, h \in \mathbb{Z}[x]$ *with* $f = gh$ *and* $\deg f = n$*. Then*

(i) $||h||_\infty \le (n+1)^{1/2} 2^n ||f||_\infty$

(ii) $||g||_\infty ||h||_\infty \le ||g||_1 ||h||_1 \le (n+1)^{1/2} 2^n ||f||_\infty$

What about the size of $\mathrm{res}(f/h, g/h)$? The following bound, based on the above bound for the magnitudes of coefficients of an integeer polynomial, and Hadamard's bound for the determinant, but taking into account the structure of $\mathrm{Syl}(f/h, g/h)$, at least gives us a bound on the magnitude of the product of all bad primes, that is, those primes that divide $\mathrm{res}(f/h, g/h)$.

Lemma 9.17. *Let* $f, g \in \mathbb{Z}[x]$, $n = \deg f \ge \deg g \ge 1$*. Let* $||f||_\infty, ||g||_\infty \le A$*. Then*

$$|res(f/h, g/h)| \le (n+1)^n A^{2n}.$$

The following example illustrates that it would be too expensive to choose primes that are large enough to guarantee they don't divide $\mathrm{res}(f,g)$.

Example 9.18. *Let* $f, g \in \mathbb{Z}[x]$ *have degrees bounded by* $n = 1000$ *and max-norm bounded by* 10^3*. Then*

- *Theorem 9.16 gives the a priori bound* $||\gcd(f,g)||_\infty \le 10^{305}$.
- *Lemma 9.17 gives the bound* $|res(f/h, g/h)| \le 10^{9001}$

9.3 A big prime modular gcd algorithm

Instead, the following algorithm chooses a random prime that is large enough to capture the coefficient of $\gcd(f,g)$, but then checks that a correct image was computed in step (4).

Algorithm: `ModularGCD`
Input: ▸ Primitive $f,g \in \mathbb{Z}[x]$, $n = \max(\deg f, \deg g)$, $A = \max(||f||_\infty, ||g||_\infty)$
Output: ▸ $\gcd(f,g) \in \mathbb{Z}[x]$
(1) $b \leftarrow \gcd(\mathrm{lc}(f), \mathrm{lc}(g))$
$B \leftarrow (n+1)^{1/2} 2^n A b$
(2) Choose a random prime p with $2B < p \le 4B$.
$v \leftarrow \gcd(f \bmod p, g \bmod p)$
(3) Compute $w, f^*, g^* \in \mathbb{Z}[x]$ with max-norm $< p/2$ such that

$$w \equiv bv \bmod p, \quad f^* w \equiv bf \bmod p, \quad g^* w = bg \bmod p$$

(4) If $||f^*||_1 ||w||_1 \le B$ and $||g^*||_1 ||w||_1 \le B$ then return $\mathrm{pp}(w)$
Else goto (2)

We will not prove it here, but mention that it can be shown rigourously that the random prime chosen in step (2) will divide $\mathrm{res}(f/h, g/h)$ with probably at most $1/2$. In other words, less than half the primes (in the worst case) in the range $2B < p \le 4B$ will divide $\mathrm{res}(f/h, g/h)$. It follows that the expected running ime of the algorihm is at most two iterations.

Example 9.19. *Consider $f = 3x^3 - x^2 + 3x - 1$ and $g = 3x^2 + 5x - 2$, both primitive polynomials.*

1. *We get $b = 3$ and $B = 240$. Note that B will always be large enough that any prime $> 2B$ will necessarily not divide either of the leading coefficients of f or g.*

2. *We choose the prime $p = 487$ and compute*

$$v = \gcd(f \bmod p, g \bmod p) = x + 162.$$

3. *Here we obtain $w = 3x - 1$ and*

$$\overbrace{(3x^2+3)}^{f^*}\overbrace{(3x-1)}^{w} \equiv \overbrace{9x^3 - 3x^2 + 9x - 3}^{bf} \pmod{p}, \qquad \overbrace{(3x+6)}^{g^*}\overbrace{(3x-1)}^{w} \equiv \overbrace{9x^2 + 15x - 6}^{bg} \pmod{p}$$

4. *Now, to verify correctness of the computed image w, we need to check that the congruences in step (3) actually hold without the mod. One way to do this is to do a multiplication over $\mathbb{Z}[x]$. Instead, the algorithm computes the a priori bound $||f^* w||_\infty \le ||f^*||_1 ||w||_1$ to check if the product $f^* g$ over $\mathbb{Z}[x]$ is such that all coefficients of $f^* g$ don't change when reduced modulo p in the symmetric range; if this is the case, then $f^* w = bf$ over $\mathbb{Z}[x]$ and w is verified to be a factor of bf. Similar for bg.*