Advanced Computer Algebra: From Newton to Hensel

Marc Moreno Maza
University of Western Ontario
AM583 – Fall 2003

April 27, 2004

1 Fast Polynomial Multiplication based on the Discrete Fourier Transform

In the whole section we consider a commutative ring R with units. Let us consider two univariate polynomials $f, g \in R[x]$ of degree less than n. Then their product has degree less than $2n - 1$. Assume that we are given a finite subset P of R with $2\,n - 1$ elements such that $f(r)$ and $g(r)$ are known for every $r \in R$. Then the values $f(r)\,g(r)$ define the polynomial $f\,g$ completely. Indeed, by Lagrange interpolation we construct the polynomial $f\,g$ (which has at most $2n - 1$ coefficients) from the $2\,n - 1$ values $f(r)\,g(r)$. Observe that the cost of defining $f\,g$ by the values $f(r)\,g(r)$ is $2\,n - 1$, which is cheap!

However if we need information about $f\,g$ such as its leading coefficient or its number of terms we need to compute the coefficients of $f\,g$. The cost of this construction is in $\mathcal{O}(n^2)$.

The underlying idea of the fast polynomial multiplication based on the discrete Fourier transform is the following. Assume that there exists a subset P of R with $2\,n - 1$ elements such that

- evaluating f and g at every $r \in P$ can be done at *nearly linear time cost* (such as $\mathcal{O}(n \log(n))$),
- interpolating $f(r)\,g(r)$ for $r \in P$ can be done at *nearly linear time cost.*

Then the multiplication of f by g, represented by their coefficients, can be done in $\mathcal{O}(n \log(n))$.

Such sets P do not always exist. This depends on the ring R and the degrees of the polynomials f and g. However many finite fields possess such sets for small enough degrees of f and g. For the arbitrary ring R, it can be shown that the multiplication of f nad g, represented by their coefficients, can be done in $\mathcal{O}(n \log(n) \log(\log(n)))$.

1.1 Primitive roots of unity

Let us recall that an element $a \in R$ is a zero divisor if $a \neq 0$ and there exists $b \in R$ such that $a\,b = 0$ and $b \neq 0$.

Let us recall also that because the ring R has a unit 1 then there is a map

$$\begin{array}{ccc} \mathbb{Z} & \longmapsto & R \\ n & \longmapsto & n\,1_R \end{array} \qquad (1)$$

which allows to talk of an integer as an element of R. However a nonzero integer n may become 0_R. For instance every multiple of p becomes 0 in $\mathbb{Z}/p\mathbb{Z}$.

Observe also the ring R may contain nonzero elements that are neither zero divisor, nor units. For instance in the ring of square matrices of order 2 with integer coefficients the matrix $\left(\begin{array}{cc} 1 & 2 \\ 0 & 3 \end{array}\right)$ is nonsingular but has no inverse (right or left). But this is not a good example since we limit ourselves to commutative rings. So consider instead the following example.

Example 1 Let $R = \mathbb{Z}/6\mathbb{Z}$ and let $U = R[x]$ be the ring of univariate polynomials over R. The ring U has units (for instance the constant polynomial 5), it has zero divisors (for instance the polynomial $2x+4$ since $3(2x+4) = 0$) and also elements like $x+1$ which is not a unit nor a zero divisor.

Why do we need to take such exotic examples? Because among the rings you are familiar with, this is the simplest example of ring having zero divisors, units and nonzero elements that are not zero divisors or units. Indeed our usual rings are $\mathbf{k}[x]$ where $\mathbf{k}$ is a field and the residue class ring $\mathbb{Z}/m\mathbb{Z}$ (with $m > 1$ not necessarily prime). The ring $\mathbf{k}[x]$ has no zero divisors whereas $\mathbb{Z}/m\mathbb{Z}$ has zero divisors iff m is not prime. But for every non prime m the ring $\mathbb{Z}/m\mathbb{Z}$ consists only of zero, units and zero divisors. This follows from the proposition below. This is why we were led to the $\mathbb{Z}/6\mathbb{Z}[x]$ example.

Proposition 1 *Let E be an Euclidean domain and m an element of E with $m \neq 0$ and $m \neq 1$. Consider the residue class ring $R = E/\langle m \rangle$ where $\langle m \rangle$ denotes the ideal generated by m. Then every element of R is either zero, a zero divisor or a unit.*

PROOF $\triangleright$ Studying the elements of R (for deciding if they are null, zero divisors or units) can be done by studying the $a \mod m$ for every $a \in E$. So let a be in E. If a is a multiple of m then $a \equiv 0 \mod m$. Assume from now on that a is not a a multiple of m. Consider g the gcd of a and m together with the Bezout relation

$$u\,a + v\,m \;=\; g \tag{2}$$

Observe that $u \mod m$ cannot be zero (otherwise $g \mod m$ would be zero and m would divide g and thus a). If $g = 1$ then $u\,a \equiv 1 \mod m$ and this shows $a \mod m$ is a unit of R. The same result holds if g is a unit. Assume from now on that g is not a unit. Since g divides a and m let a' and m' be such that

$$a \;=\; g\,a' \quad \text{and} \quad m \;=\; g\,m' \tag{3}$$

Observe that $m' \mod m$ is not zero (otheriwse m' would write $m' = m''m$ leading to the contradiction $g\,m'' = 1$). We have

$$m'\,a \;=\; m'\,g\,a' \;=\; m\,a' \equiv 0 \mod m \tag{4}$$

This shows that $a \mod m$ is a zero divisor of R. Yes, this was that simple! $\triangleleft$

Definition 1 *Let n be a positive integer and $\omega \in R$.*

1. *ω is a* n-th root of unity *if $\omega^n = 1$.*
2. *ω is a* primitive n-th root of unity *if*
   (a) *$\omega^n = 1$,*
   (b) *n is a unit in R,*
   (c) *for every prime divisor t of n the element $\omega^{n/t} - 1$ is neither zero nor a zero divisor.*

Remark 1 When R is an integral domain the last condition of Definition 1 becomes: for every prime divisor t of n the element $\omega^{n/t} \neq 1$. Observe also that a n-th root of unity is necessarily a unit.

Example 2 Consider that R is the field $\mathbb{C}$ of complex numbers. The number $\omega = e^{2\,\imath\pi/8}$ is a a primitive 8-th root of unity.

Example 3 In $R = \mathbb{Z}/8\mathbb{Z}$ we have $3^2 \equiv 1$. However 3 is not a primitive 2-th of unity, since $n = 2$ is not a unit in R.

Example 4 In $R = \mathbb{Z}/17\mathbb{Z}$ we have the following computation in AXIOM

```
(1) -> R := PF(17)

   (1)  PrimeField 17
                                                 Type: Domain
(2) -> w: R := 3

   (2)  3
                                           Type: PrimeField 17
(3) -> [w^i for i in 0..16]

   (3)  [1,3,9,10,13,5,15,11,16,14,8,7,4,12,2,6,1]
                                        Type: List PrimeField 17
(4) -> u: R := 2

   (4)  2
                                             Type: PrimeField 17
(5) -> [u^i for i in 0..16]

   (5)  [1,2,4,8,16,15,13,9,1,2,4,8,16,15,13,9,1]
                                         Type: List PrimeField 17
```

The first list shows that 3 is a primitive 16-th root of unity. However with $\omega = 2$ we have $\omega^8 - 1 = 0$ since $(2^4 - 1)(2^4 + 1) = 15 \times 17 \equiv 0$.

Lemma 1 *Let $1 < \ell < n$ be integers and let ω be a primitive n-th root of unity. Then we have*

1. *$\omega^\ell - 1$ is neither zero nor a zero divisor in R,*
2. *$\Sigma_{0 \leq j < n}\, \omega^{j\,\ell} = 0$.*

Proof $\triangleright$ It relies on the formula

$$(c-1)\ \Sigma_{0 \le j < m}\ c^j \quad = \quad c^m - 1 \tag{5}$$

which holds for every $c \in R$ and every positive integer m.

Let us prove the first statement of the lemma. Let g be the gcd of ℓ and n. Let $u, v \in \mathbb{Z}$ be such that

$$u\,\ell + v\,n \ = \ g \tag{6}$$

Since $\ell < n$ we have $1 \le g < n$. Hence we can cancel a prime factor t of n such that

$$g \ \mid \ (n/t) \tag{7}$$

Let

$$c \ = \ \omega^g \ \ \text{and} \ \ m = n/(tg) \tag{8}$$

in Relation (5) leading to

$$(\omega^g - 1)\ a \quad = \quad (\omega^{n/t} - 1) \tag{9}$$

for some $a \in R$. Hence if $(\omega^g - 1)$ would be zero or a zero divisor then so would be $(\omega^{n/t} - 1)$ which is false. Now applying Relation (5) with $c = \omega^\ell$ and $m = u$ implies that $(\omega^\ell - 1)$ divides $(\omega^{u\,\ell} - 1)$. But with Relation (6) we obtain

$$(\omega^{u\,\ell} - 1) \ = \ (\omega^{u\,\ell}\,\omega^{v\,n} - 1) \ = \ (\omega^g - 1) \tag{10}$$

Hence

$$(\omega^\ell - 1) \ \mid \ (\omega^g - 1) \tag{11}$$

Therefore $(\omega^\ell - 1)$ cannot be zero or a zero divisor.

Now let us prove the second statement of the lemma. By applying Relation (5) with $c = \omega^\ell$ and $m = n$ we have

$$(\omega^\ell - 1)\ \Sigma_{0 \le j < n}\ \omega^{\ell j} \quad = \quad \omega^{\ell n} - 1 \quad = 0 \tag{12}$$

Since $(\omega^\ell - 1)$ is neither zero nor a zero divisor we otain the desired formula. $\triangleleft$

1.2 The Discrete Fourier Transform

Let n be a positive integer and $\omega \in R$ be a primitive n-th root of unity. In what follows we identify every univariate polynomial

$$f \; = \; \Sigma_{0 \leq i < n} \, f_i \, x^i \; \in R[x] \tag{13}$$

of degree less than n with its coefficient vector $(f_0, \ldots, f_{n-1}) \in R^n$.

Definition 2 *The R-linear map*

$$DFT_\omega \; : \; \left\{ \begin{array}{rcl} R^n & \longmapsto & R^n \\ f & \longmapsto & (f(1), f(\omega), f(\omega^2), \ldots, f(\omega^{n-1})) \end{array} \right. \tag{14}$$

which evaluates a polynomial at the powers of ω is called the Discrete Fourier Transform *(DFT).*

Proposition 2 *The R-linear map DFT_ω is an isomorphism.*

PROOF $\triangleright$ Since the R-linear map DFT_ω is an endomorphism (the source and target spaces are the same) we only need to prove that DFT_ω is bijective. Observe that the Vandermonde matrix $VDM(1, \omega, \omega^2, \ldots, \omega^{n-1})$ is the matrix of the R-linear map DFT_ω. Then for proving that DFT_ω is bijective we need only to prove that $VDM(1, \omega, \omega^2, \ldots, \omega^{n-1})$ is invertible which holds iff the values $1, \omega, \omega^2, \ldots, \omega^{n-1}$ are pairwise different. A relation $\omega^i = \omega^j$ for $0 \leq i < j < n$ would imply $\omega^i (1 - \omega^{j-i}) = 0$. Since $(1 - \omega^{j-i})$ cannot be zero or a zero divisor then ω^i and thus ω must be zero. Then ω cannot be a root of unity. A contradiction. Therefore the values $1, \omega, \omega^2, \ldots, \omega^{n-1}$ are pairwise different and DFT_ω is an isomorphism. $\triangleleft$

Proposition 3 *Let V_ω denote the matrix of the isomorphism DFT_ω. Then ω^{-1} the inverse of ω is also a primitive n-th root of unity and we have*

$$V_\omega \, V_{\omega^{-1}} \; = \; n I_n \tag{15}$$

where I_n denotes the unit matrix of order n.

PROOF $\triangleright$ Define $\omega' = \omega^{-1}$. Observe that $\omega' = \omega^{n-1}$. Thus ω' is a root of unity. We leave to the reader the proof that this a primitive n-th root of unity, Let us consider the product of the matrix V_ω and $V_{\omega'}$. The element at row i and column k is

$$\begin{array}{rcl} (V_\omega\, V_{\omega'})_{ik} & = & \Sigma_{0\,\leq\,j\,<\,n}\ (V_\omega)_{ij}(V_{\omega'})_{jk} \\ & = & \Sigma_{0\,\leq\,j\,<\,n}\ \omega^{ij}\,(\omega')^{jk} \\ & = & \Sigma_{0\,\leq\,j\,<\,n}\ \omega^{ij}\,\omega^{-jk} \\ & = & \Sigma_{0\,\leq\,j\,<\,n}\ (\omega^{i-k})^{j} \end{array} \tag{16}$$

Observe that ω^{i-k} is either a power of ω or a power of its inverse. Thus, in any case this is a power of ω. If $i = k$ this power is 1 and $(V_\omega\, V_{\omega'})_{ik}$ is equal to n. If $i \neq k$ then the conclusion follows by applying the second statement of Lemma 1 which shows that $(V_\omega\, V_{\omega'})_{ik} = 0$. $\lhd$

1.3 Convolution of polynomials

Let n be a positive integer and $\omega \in R$ be a primitive n-th root of unity.

Definition 3 *The* convolution *w.r.t. n of the polynomials $f = \Sigma_{0\,\leq\,i\,<\,n}\ f_i\, x^i$ and $g = \Sigma_{0\,\leq\,i\,<\,n}\ g_i\, x^i$ in $R[x]$ is the polynomial*

$$h \ = \ \Sigma_{0\,\leq\,k\,<\,n}\ h_k\, x^k \tag{17}$$

such that for every $k = 0 \cdots n-1$ the coefficient h_k is given by

$$h_k \ = \ \Sigma_{i+j\equiv k \mod n}\ \ f_i\, g_j \tag{18}$$

The polynomial h is also denoted by $f *_n g$ or simply by $f * g$ if not ambiguous.

Remark 2 Observe that the product of f by g is

$$p \ = \ \Sigma_{0\,\leq\,k\,<\,2n-1}\ \ p_k\, x^k \tag{19}$$

where for every $k = 0 \cdots 2n-2$ the coefficient p_k is given by

$$p_k \ = \ \Sigma_{i+j=k}\ \ f_i\, g_j \tag{20}$$

We can rearrange the polynomial p as follows.

$$\begin{array}{rcl} p & = & \Sigma_{0\,\leq\,k\,<\,n}\ (p_k\, x^k) \ \ + \ \ x^n\ \Sigma_{0\,\leq\,k\,<\,n-1}\ (p_{k+n}\, x^k) \\ & = & \Sigma_{0\,\leq\,k\,<\,n}\ (p_k \ + \ p_{k+n}\, x^n)\, x^k \end{array} \tag{21}$$

Therefore we have

$$f\, *\, g \ \equiv \ fg \mod x^n - 1 \tag{22}$$

Example 5 Let $R = \mathbb{Z}/5\mathbb{Z}$, $n = 4$, $u \in R$ and consider the polynomials

$$f \; = \; x^3 + 1 \quad \text{and} \quad g \; = \; 2x^3 + 3x^2 + x + 1. \tag{23}$$

Assume that we want to multiply f and g modulo $x^4 = u$. Then we have to arrange the product fg in a form $q(x^4 - u) + r$ where r has degree less than 4. We obtain

$$\begin{array}{rcl} f\,g & = & 2\,x^6 + 3\,x^5 + x^4 + 3\,x^3 + 3\,x^2 + x + 1 \\ & = & (2x^2 + 3x + 1)(x^4 - u) + 3x^3 + 3x^2 + x + 1 + (2x^2 + 3x + 1)u \\ & = & (2x^2 + 3x + 1)(x^4 - u) + 3x^3 + (3 + 2u)x^2 + (1 + 3u)x + (1 + u) \end{array} \tag{24}$$

For $u = 1$ we obtain

$$f \, * \, g \; = \; 3x^3 + 4x + 2 \tag{25}$$

Lemma 2 *For $f, g \in R[x]$ univariate polynomials of degree less than n we have*

$$DFT_\omega(f * g) \; = \; DFT_\omega(f) DFT_\omega(g) \tag{26}$$

where the product of the vectors $DFT_\omega(f)$ and $DFT_\omega(g)$ is computed componentwise.

Proof $\triangleright$ Since $f \, * \, g$ and $f\,g$ are equivalent modulo $x^n - 1$, there exists a polynomial $q \in R[x]$ such that

$$f \, * \, g \; = \; f\,g + q\,(x^n - 1) \tag{27}$$

Hence for $i = 0 \cdots n-1$ we have

$$\begin{array}{rcl} (f \, * \, g)(\omega^i) & = & f(\omega^i)\, g(\omega^i) + q(\omega^i)((\omega^{i\,n} - 1) \\ & = & f(\omega^i)\, g(\omega^i) \end{array} \tag{28}$$

by application of Lemma 1. $\triangleleft$

Remark 3 Consider the multi-point evaluation map

$$E_\omega \, : \, \left\{ \begin{array}{rcl} R[x] & \longmapsto & R^n \\ f & \longmapsto & (f(1), f(\omega), f(\omega^2), \ldots, f(\omega^{n-1})) \end{array} \right. \tag{29}$$

Its kernel is generated by $x^n - 1$. Since E_ω is surjective, it follows that

$$R[x]/\langle x^n - 1 \rangle \; \simeq \; R^n \tag{30}$$

(which is not a surprise!) and DFT_ω realizes this isomorphism.

If R is a field, then this a special case of the Chinese Remaindering Theorem where $m_i = x - \omega^i$ for $i = 0 \cdots n - 1$.

1.4 The Fast Fourier Transform

The Fast Fourier Transform computes the DFT quickly. This important algorithm for computer science (not only computer algebra, but also digital signal processing for instance) was (re)-discovered in 1965 by Cooley and Tukey.

Let n be a positive **even** integer, $\omega \in R$ be a primitive n-th root of unity and $f = \Sigma_{0 \leq i < n} f_i x^i$. In order to evaluate f at $1, \omega, \omega^2, \ldots, \omega^{n-1}$. we divide f by $x^{n/2} - 1$ and $x^{n/2} + 1$ with remainder. So let q_0, q_1, r_0, r_1 be polynomials such that

$$f = q_0(x^{n/2} - 1) + r_0 \quad \text{with} \quad \left\{ \begin{array}{l} \deg(r_0) < n/2 \\ \deg(q_0) < n/2 \end{array} \right. \tag{31}$$

and

$$f = q_1(x^{n/2} + 1) + r_1 \quad \text{with} \quad \left\{ \begin{array}{l} \deg(r_1) < n/2 \\ \deg(q_1) < n/2 \end{array} \right. \tag{32}$$

The relations $\deg(q_0) < n/2$ and $\deg(q_1) < n/2$ hold because the polynomial f has degree less than n. Observe that the computation of (q_0, r_0) and (q_1, r_1) can be done very easily. Indeed, let $F_0, F_1 \in R[x]$ be such that

$$f = F_1 \, x^{n/2} + F_0 \quad \text{with} \quad \left\{ \begin{array}{l} \deg(F_1) < n/2 \\ \deg(F_0) < n/2 \end{array} \right. \tag{33}$$

We have

$$f = F_1(x^{n/2} - 1) + F_0 + F_1 \quad \text{and} \quad f = F_1(x^{n/2} + 1) + F_0 - F_1 \tag{34}$$

Hence we obtain

$$r_0 = F_0 + F_1 \quad \text{and} \quad r_1 = F_0 - F_1 \tag{35}$$

Let i be an integer such that $0 \leq i < n/2$. By using Relation (31) with $x = \omega^{2\,i}$ we obtain

$$f(\omega^{2\,i}) = q_0(\omega^{2\,i})(\omega^{n\,i} - 1) + r_0(\omega^{2\,i}) = r_0(\omega^{2\,i}) \tag{36}$$

since $\omega^{n\,i} = 1$. Then, by using Relation (32) with $x = \omega^{2\,i+1}$ we obtain

$$f(\omega^{2\,i+1}) = q_1(\omega^{2\,i+1})(\omega^{n\,i}\,\omega^{n/2} + 1) + r_1(\omega^{2\,i+1}) = r_1(\omega^{2\,i+1}) \tag{37}$$

since $\omega^{n/2} = -1$. Indeed, this last equation follows from

$$0 \;=\; \omega^n - 1 \;=\; (\omega^{n/2} - 1)(\omega^{n/2} + 1) \tag{38}$$

and the fact that $\omega^{n/2} - 1$ is not zero nor a zero divisor. Therefore we have proved the following

Proposition 4 *Evaluating $f \in R[x]$ (with degree less than n) at $1, \omega^1, \ldots, \omega^{n-1}$ is equivalent to*

- *evaluate r_0 at the even powers $\omega^{2\,i}$ for $0 \leq i < n/2$.*
- *evaluate r_1 at the odd powers $\omega^{2\,i+1}$ for $0 \leq i < n/2$.*

Since it is easy to show that ω^2 is a primitive $n/2$-th root of unity we can hope for a recursive algorithm. This algorithm would be easier if both r_0 and r_1 would be evaluated at the same points. So we define

$$r_1(\omega^{2\,i+1}) \;=\; {r_1}^*(\omega^{2\,i}) \tag{39}$$

Now we obtain the following algorithm.

Algorithm 1

Input: $n = 2^k$, $f = \Sigma_{0 \leq i < n} \; f_i \, x^i$, and the powers $1, \omega, \omega^2, \ldots, \omega^{n-1}$ of a primitive n-th root of unity $\omega \in R$.

Output: $DTF_\omega(f) = (f(1), f(\omega), f(\omega^2), \ldots, f(\omega^{n-1}))$.

if $n = 1$ **return** (f_0)
$r_0 := \Sigma_{0 \leq j < n/2} \; (f_j + f_{j+n/2}) x^j$
$r_1^* := \Sigma_{0 \leq j < n/2} \; \omega^j \; (f_j - f_{j+n/2}) x^j$
call the algorithm recursive to evaluate r_0 and r_1^*
at the k first powers of ω^2
return$(r_0(1), r_1^*(1), r_0(\omega^2), r_1^*(\omega^2), \ldots, r_0(\omega^{n-2}), r_1^*(\omega^{n-2}))$

Theorem 1 *Let n be a power of 2 and $\omega \in R$ a primitive n-th root of unity. Then Algorithm 1 computes $DTF_\omega(f)$ using*

- $n \log(n)$ *additions in* R,
- $(n/2) \log(n)$ *multiplications by powers of* ω.

leading in total to $3/2\, n \log(n)$ *ring operations.*

Proof $\triangleright$ By induction on $k = \log_2(n)$. Let $S(n)$ and $T(n)$ be the number of additions and multiplications in R that the algorithms requires for an input of size n. If $k = 0$ the algorithm returns (f_0) whose costs is null thus we have $S(0) = 0$ and $T(0) = 0$ which satisfies the formula since $\log(n) = \log(1) = 0$. Assume $k > 0$. Just by looking at the algorithm we that

$$S(n) \;=\; 2\, S(n/2) + n \quad \text{and} \quad T(n) \;=\; 2\, T(n/2) + n/2 \tag{40}$$

leading to the result by plugging in the induction hypothesis. $\triangleleft$

1.5 Fast Convolution and Multiplication

Algorithm 2

Input: $f, g \in R[x]$ with degree less than $n = 2^k$, a primitive n-th root of unity $\omega \in R$.

Output: $f * g \in R[x]$

compute $1, \omega, \omega^2, \ldots, \omega^{n-1}$
$\alpha := DTF_\omega(f)$
$\beta := DTF_\omega(g)$
$\gamma := \alpha\, \beta$
return $(DTF_\omega)^{-1}(\gamma) \;=\; 1/n\, DTF_{\omega^{-1}}(\gamma)$

Theorem 2 *Let* n *be a power of* 2 *and* $\omega \in R$ *a primitive* n*-th root of unity. Then convolution in* $R[x]/\langle x^n - 1\rangle$ *and multiplication in* $R[x]$ *of polynomials whose product has degree less than* n *can be performed using*

- $3\, n \log(n)$ *additions in* R,
- $3/2\, n \log(n) + n - 2$ *multiplications by a power of* ω,
- n *multiplications in* R,

- *n divisions by n (as an element of R),*

leading to $9/2\, n \log(n) + \mathcal{O}(n)$ operations in R.

Remark 4 To multiply two arbitrary polynomials of degree less than $n \in \mathbb{N}$ we only need a primitive 2^k-th root of unity where

$$2^{k-1} \;<\; 2\,n \;\leq 2^k \tag{41}$$

Then we have decreased the cost of about $\mathcal{O}(n^2)$ of the classical algorithm to $\mathcal{O}(n \log(n))$.

2 Computing primitive n-th roots of unity

2.1 Some results from group theory

Definition 4 *Let $(G, .)$ be a (multiplicative) group with neutral element e. A nonempty subset $H \subseteq G$ is a* subgroup *of G if the following three statements hold*

- *$e \in H$,*
- *for every $x, y \in H$ we have $x\, y \in H$,*
- *for every $x \in H$ the inverse x^{-1} of x belongs to H.*

Theorem 3 *For every subgroup H of the additive abelian group $(\mathbb{Z}, +)$ there exists an element $a \in \mathbb{Z}$ such that H is the set of the multiples of a, that is $H = a\mathbb{Z}$.*

Theorem 4 *Let G be a multiplicative group with neutral element e. Let $x \in G$ an element and $\mathsf{gr}(x)$ the subgroup of H consisting of all powers of x (including $e = x^0$ and x^{-1} the inverse of x). Let $\Theta(x)$ be the order of $\mathsf{gr}(x)$, that is the cardinality of $\mathsf{gr}(x)$. Then two cases arise*

- *either $\mathsf{gr}(x)$ is infinite and then the powers of x are pairwise different and thus H is isomorphic to $\mathbb{Z}$.*
- *or $\mathsf{gr}(x)$ is finite and we have the following properties*

  1. *$\Theta(x)$ is the smallest integer n such that $x^n = e$,*

2. $x^m = x^{m'}$ *iff* $m \equiv m' \mod \Theta(x)$,

3. H *is isomorphic to* $\mathbb{Z}/n\mathbb{Z}$ *where* $n = \Theta(x)$,

4. $H = \{e, x, x^2, \dots, x^{n-1}\}$ *where* $n = \Theta(x)$.

Theorem 5 (Lagrange) *For every subgroup H of the* **finite** *group G, the order (that is the cardinality) of H divides that of G.*

2.2 Primitive n-th roots of unity of finite fields

Theorem 6 *For $n, p > 1$, the finite field $\mathbb{Z}/p\mathbb{Z}$ has a primitive n-th root of unity if and only if n divides $p - 1$.*

PROOF ▷ If ω is a a primitive n-th root of unity in $\mathbb{Z}/p\mathbb{Z}$ then the set

$$\Omega = \{1, \omega, \dots, \omega^{n-1}\} \tag{42}$$

forms a cyclic subgroup H of the multiplicative group G_{p-1} of $\mathbb{Z}/p\mathbb{Z}$. By vertue of Lagrange's theorem (Theorem 5) the cardinality of H divides that of G_{p-1}. Since G_{p-1} has $p-1$ elements, n divides $p-1$.

Conversly, it is known from finite field theory that G_{p-1} is a cyclic group (even if p is a power of a prime rather than a prime). Let α be a generator of this group, that is

$$G_{p-1} = \{1, \alpha, \alpha^2, \dots, \alpha^{p-2}\} \tag{43}$$

Recall that $\alpha^{p-1} = 1$ from the little Fermat's theorem. Let $n > 1$ be an integer dividing $p-1$ and define

$$\omega = \alpha^{(p-1)/n}. \tag{44}$$

Then we have

$$\omega^n = 1. \tag{45}$$

For all $0 < k < n$ we have $k\frac{p-1}{n} < p-1$ so we have

$$\omega^k = \alpha^{k(p-1)/n} \neq 1. \tag{46}$$

This shows that ω is a primitive n-th root of unity. ◁

Example 6 Since $8 \mid (41-1)$ in $\mathbb{Z}/41\mathbb{Z}$ we have primitive 8-th root of unity. In fact the element $14 \in \mathbb{Z}/41\mathbb{Z}$ is such a root of unity.

Remark 5 Theorem 6 gives a necessary and sufficient condition for the existence of primitive n-th roots of unity in $\mathbb{Z}/p\mathbb{Z}$. But this does not give an algorithm to construct them.

We could use brute force. Given $\mathbb{Z}/p\mathbb{Z}$, for every n that we are interested in, for every $g \in G_{p-1}$ try if the following both statements hold:

- $g^n = 1$,
- for every prime divisor t of n we have $g^{n/t} \neq 1$.

However we can reduce the complexity of this search by computing a generator of G_{p-1} by means of Theorem 7 and then applying Relation (44).

Theorem 7 *An element α of the multiplicative subgroup G_{p-1} of the prime finite field $\mathbb{Z}/p\mathbb{Z}$ is a generator of G_{p-1} iff for every prime factor t of $p-1$ we have*

$$\alpha^{(p-1)/t} \neq 1 \tag{47}$$

Proof $\triangleright$ Follows from Lagrange's theorem by using techniques like in the proof of Lemma 1. $\triangleleft$

Remark 6 For the purpose of multiplying polynomials with coefficients in $\mathbb{Z}/p\mathbb{Z}$ we are interested in primes p and integers n of the form $n = 2^k$ such that $\mathbb{Z}/p\mathbb{Z}$ possesses a primitive n-th root of unity. By Theorem 6, the field $\mathbb{Z}/p\mathbb{Z}$ has such a root if there exists $q \in \mathbb{Z}$ such that

$$p = q\,2^k + 1 \tag{48}$$

Such primes are called *Fourier primes*.

But not all numbers of the form $q\,2^k + 1$ are prime (consider $q = k = 2$). So how frequent are the primes among the numbers of the form $q\,2^k + 1$ (for a given k). The answer is giving by the following theorem.

Theorem 8 *Let a and b be two relatively prime integers. Then the number of primes*

- *in the arithmetic progression $a\,k + b$ for $k = 1, 2, \ldots$*
- *and not greater than a given integer x*

is approximatively

$$\frac{x}{\phi(a)\log(x)} \tag{49}$$

where $\phi(a)$ is the Euler function at a (the number of integers less than and relatively prime to a).

Remark 7 We apply the previous formula with $a = 2^k$. All odd numbers less than 2^k are relatively prime to 2^k. (The even numbers are not relatively prime to 2^k.) Thus $\phi(2^k) = 2^{k-1}$. Therefore there are approximatively

$$\frac{x}{2^{k-1}\log(x)} \tag{50}$$

Fourier primes less than a given integer x.

Let $x = 2^{31}$, which represents the usual size required for single precision integers. For $k = 20$, there are approximatively 130 Fourier primes

- of the form $2^e\, k + 1$ with $e \geq 20$,
- and in the range $2^{20} \cdots 2^{31}$.

This shows that there are 130 prime finite fields $\mathbb{Z}/p\mathbb{Z}$ such that

- p fits in a machine integer (for a 32-bit processor)
- and allowing multiplication of polynomials in $\mathbb{Z}/p\mathbb{Z}[x]$ whose product is of degree less than 1048576.

So it looks like we have many possible primes for doing **fast** modular computations with polynomials. However primes p such that p^2 fits in a machine integer are preferable in order to avoid the use of two machine words for multiplying two elements of $\mathbb{Z}/p\mathbb{Z}$.

In the *BasicMath* library, one of the parents of `libalgebra`, *nice primes* are organized in tables. So there is a category for *tables of primes*!

```
macro SI == SingleInteger;

PrimesTableCategory(s: SI): Category ==  with {
        sizeBound: () -> SI;
        tableSize: () -> SI;
        rank: SI -> Partial(SI);
```

```
        maxPrime: () -> SI;
        minPrime: () -> SI;
        previousPrime: SI -> SI;
        nextPrime: SI -> SI;
        primes: () -> Generator(SI);
        FourierDegree: SI -> SI;
        getPrimeOfFourierDegree: SI -> SI;
        nextPrimeOfFourierDegree: (SI, SI) -> SI;
        unitsGenerator: SI -> SI;
        primitiveRootofUnity: (SI, SI) -> SI;
}
```

where

- `PrimesTableCategory(s)` specifies the operations that we expect from a table of primes p less than 2^s. For obvious reasons, this category is restricted to primes that fit in a machine word.
- `sizeBound()` returns `s`.
- `tableSize()` returns the number of entries in the table.
- `rank(p)` returns tha rank of p in the table if it lies in.
- `FourierDegree(p)` returns the biggest integer r in the table such that 2^r divides $p - 1$.
- `getPrimeOfFourierDegree(r)` returns the smallest prime p in the table such that `FourierDegree(p) >= r`, if any, otherwise 0 is returned.
- `nextPrimeOfFourierDegree(n,r)` returns the smallest prime p in the table such that `FourierDegree(p) = r` and $p > n$ hold, otherwise 0 is returned.
- `unitsGenerator(p)` assuming that p lies in the table returns a generator of the units group of the integers modulo p.
- `primitiveRootofUnity(p,n)` assuming that p lies in the table and assuming $0 < n \leq r$ where r is `FourierDegree(p)`, returns a primitive 2^n-root of unity of the units group of the integers modulo p.

```
macro SI == SingleInteger;

FourierPrimesTableCategory(r: SI, s: SI): Category ==  with {
        FourierDegree: () -> SI
        sizeBound: () -> SI
        tableSize: () -> SI
        rank: SI -> Partial(SI)
        maxPrime: () -> SI
        minPrime: () -> SI
        previousPrime: SI -> SI
        nextPrime: SI -> SI
        primes: () -> Generator(SI)
        unitsGenerator: SI -> SI
        primitiveRootofUnity: (SI, SI) -> SI
}
```

where`FourierPrimesTableCategory(r,s)` specifies the operations that we expect from the table of Fourier primes p less than 2^s and such that there exists an odd integer q satisfying $p - 1 = 2^r q$. Again we restrict to primes that fit in a machine word.

```
F6P15: FourierPrimesTableCategory(6,15) == add {
         local fd: SI == 6;
         local sb: SI == 15;
 -------------------------------------------------------------------
 -- Table (6,15)                                                  --
 -------------------------------------------------------------------

 -- Maximal size for a FFT: 2^6
 -- Number of 6-Fourier primes less than 2^15 is 58
MAXINDEX: SingleInteger == 58;
 -- These Fourier primes are:
local primeList: Array SingleInteger :=
   [193, 449, 577, 1217, 1601, 2113, 2753, 3137,
    4289, 4673, 4801, 5441, 5569, 5953, 6337, 6977,
    7489, 7873, 8513, 8641, 9281, 10177, 10433, 11329,
    11969, 12097, 13121, 13249, 13633, 14401, 14657, 15809,
```

```
    15937, 16193, 17729, 19009, 19777, 20161, 20929, 21313,
    21569, 22721, 23873, 24001, 25153, 25409, 25537, 25793,
    26177, 26561, 27073, 27329, 27457, 28097, 29633, 29761,
    30529, 32321];

 -- The corresponding units generators are:
local generatorList: Array SingleInteger :=
   [5, 3, 5, 3, 3, 5, 3, 3,
    3, 3, 7, 3, 13, 7, 10, 3,
    7, 5, 5, 17, 3, 7, 3, 7,
    3, 5, 7, 7, 5, 11, 3, 3,
    7, 5, 3, 23, 11, 13, 7, 5,
    3, 3, 3, 14, 10, 3, 10, 3,
    3, 3, 5, 3, 7, 3, 3, 17,
    13, 6];

        FourierDegree(): SI == fd;
        sizeBound(): SI == sb;
        tableSize(): SI == MAXINDEX;
..............................................
}
```

```
F7P15: FourierPrimesTableCategory(7,15) == add {
         local fd: SI == 7;
         local sb: SI == 15;
 --------------------------------------------------------------------
 -- Table (7,15)                                                   --
 --------------------------------------------------------------------

 -- Maximal size for a FFT: 2^7
 -- Number of 7-Fourier primes less than 2^15 is 29
MAXINDEX: SingleInteger == 29;
 -- These Fourier primes are:
local primeList: Array SingleInteger :=
   [641, 1153, 1409, 2689, 3457, 4481, 4993, 6529,
    7297, 9601, 9857, 10369, 11393, 12161, 13441, 13697,
```

```
    15233, 16001, 18049, 19073, 19841, 20353, 21121, 21377,
    26497, 28289, 29569, 30593, 31873];

 -- The corresponding units generators are:
local generatorList: Array SingleInteger :=
   [3, 5, 3, 19, 7, 3, 5, 7,
    5, 13, 5, 13, 3, 3, 11, 3,
    3, 3, 13, 3, 3, 5, 19, 3,
    5, 6, 17, 3, 11];

        FourierDegree(): SI == fd;
        sizeBound(): SI == sb;
        tableSize(): SI == MAXINDEX;
```

....

```
F8P15: FourierPrimesTableCategory(8,15) == add {
         local fd: SI == 8;
         local sb: SI == 15;
 --------------------------------------------------------------------
 -- Table (8,15)                                                  --
 --------------------------------------------------------------------

 -- Maximal size for a FFT: 2^8
 -- Number of 8-Fourier primes less than 2^15 is 12
MAXINDEX: SingleInteger == 12;
 -- These Fourier primes are:
local primeList: Array SingleInteger :=
   [257, 769, 3329, 7937, 9473, 14081, 14593, 22273,
    23297, 26881, 30977, 31489];

 -- The corresponding units generators are:
local generatorList: Array SingleInteger :=
   [3, 11, 3, 3, 3, 3, 5, 5,
    3, 11, 3, 7];

        FourierDegree(): SI == fd;
        sizeBound(): SI == sb;
```

```
        tableSize(): SI == MAXINDEX;
............................................................


9P23: FourierPrimesTableCategory(9,23) == add {
         local fd: SI == 9;
         local sb: SI == 23;
 --------------------------------------------------------------------
 -- Table (9,23)                                                  --
 --------------------------------------------------------------------

 -- Maximal size for a FFT: 2^9
 -- Number of 9-Fourier primes less than 2^23 is 1092
        MAXINDEX: SingleInteger == 1092;
 -- These Fourier primes are:
        local primeList: Array SingleInteger :=
           [7681, 10753, 11777, 17921, 23041, 26113, 32257,
............................................................

F10P24: FourierPrimesTableCategory(10,24) == add {
         local fd: SI == 10;
         local sb: SI == 24;
 --------------------------------------------------------------------
 -- Table (10,24)                                                  --
 --------------------------------------------------------------------

 -- Maximal size for a FFT: 2^10
 -- Number of 10-Fourier primes less than 2^24 is 1087
        MAXINDEX: SingleInteger == 1087;
 -- These Fourier primes are:
        local primeList: Array SingleInteger :=
           [13313, 15361, 19457, 25601,
............................................................

F11P25: FourierPrimesTableCategory(11,25) == add {
         local fd: SI == 11;
         local sb: SI == 25;
 --------------------------------------------------------------------
```

```
 -- Table (11,25)                                                    --
 ---------------------------------------------------------------------

 -- Maximal size for a FFT: 2^11
 -- Number of 11-Fourier primes less than 2^25 is 978
        MAXINDEX: SingleInteger == 978;
 -- These Fourier primes are:
        local primeList: Array SingleInteger :=
           [18433,
...........................................................

F12P26: FourierPrimesTableCategory(12,26) == add {
         local fd: SI == 12;
         local sb: SI == 26;
 ---------------------------------------------------------------------
 -- Table (12,26)                                                    --
 ---------------------------------------------------------------------

 -- Maximal size for a FFT: 2^12
 -- Number of 12-Fourier primes less than 2^26 is 972
        MAXINDEX: SingleInteger == 972;
 -- These Fourier primes are:
        local primeList: Array SingleInteger :=
           [12289,
...........................................................
```

3 Efficient implementation

The theoretical complexity of Algorithm 1 is good but does not lead to a practical implementation of a fast multiplication of polynomials. According to [GCL92] it is not better than the classical algorithm for $n \leq 300$. The reasons are

- the recursive calls,
- the creation of intermediate polynomials.

We are going to get rid off these two difficulties leading to a practical algorithm better than the classical one for $n \geq 32$ (in Aldor).

3.1 Towards an iterative algorithm for the DTF

We first change the intermediate polynomials. Let $n = 2^r$ be a power of 2, let $\omega \in R$ be a primitive n-th root of unity and $A(x) = \Sigma_{0 \leq i < n} \, a_i \, x^i$. We define

$$\begin{array}{rcl} A^{[0]}(x) & = & a_0 + a_2 x + a_4 x^2 + \cdots + a_{n-2} x^{n/2-1} \\ A^{[1]}(x) & = & a_1 + a_3 x + a_5 x^2 + \cdots + a_{n-1} x^{n/2-1} \end{array} \tag{51}$$

such that

$$A(x) \ = \ A^{[0]}(x^2) + x \, A^{[1]}(x^2) \tag{52}$$

Hence evaluating A at $1, \omega, \omega^2, \ldots, \omega^{n-1}$ is reduced to evaluate $A^{[0]}$ and $A^{[1]}$ at $1, \omega^2, \omega^4, \ldots, \omega^{2n-2}$. But this second sequence of values consists of two identical sequences of length $n/2$. Indeed for every integer k we have

$$(\omega^{k+n/2})^2 \ = \ (\omega^k)^2 \tag{53}$$

This shows

$$\omega^{k+n/2} \ = \ -\omega^k \tag{54}$$

Hence we only need to evaluate $A^{[0]}$ and $A^{[1]}$ at

$$1, \omega^2, \omega^4, \ldots, \omega^{n-2} \tag{55}$$

which allows us to evaluate A at

$$1, \omega, \omega^2, \ldots, \omega^{n/2-1} \tag{56}$$

Then to compute at A at

$$\omega^{n/2}, \omega^{n/2+1}, \ldots, \omega^{n-1} \tag{57}$$

we use the fact these numbers are respectively equal to

$$-1, -\omega, -\omega^2, \ldots, -\omega^{n/2-1}. \tag{58}$$

Algorithm 3

Input: $n = 2^r$, $A = [a_0, a_1, \ldots, a^{n/2-1}, \ldots, a^{n-1}]$ an array for the coefficient of the polynomial, $\Omega = [1, \omega, \ldots, \omega^{n/2-1}, \ldots, \omega^{n-1}]$ an array of the powers of a primitive n-th root of unity ω.

Output: An array $y = [y_0, \ldots, y_{n-1}]$ such that y_i is $A(\omega^i)$ by calling $DFT(n, A, \Omega)$.

$DFT(n, A, \Omega) ==$
  $step := \#(\Omega)/n$
  if $n = 1$ **return** $[f_0]$
  $A^{[0]} := [a_0, a_2, \ldots, a_{n-2}]$
  $A^{[1]} := [a_1, a_3, \ldots, a_{n-1}]$
  $y^{[0]} := DFT(n/2, A^{[0]}, \Omega)$
  $y^{[1]} := DFT(n/2, A^{[1]}, \Omega)$
  for $k := 0 \cdots n/2 - 1$ **repeat**
    $t := \Omega_{\mathbf{step\ k}}\ \ y_k^{[1]}$
    $y_k := y_k^{[0]} + t$
    $y_{k+n/2} := y_k^{[0]} - t$
  return y

The recursive calls of $DFT(n, A, \Omega)$ defines a ordering of the coefficients of A shown on Figure 1. Let us call this ordering the *DFT ordering* of A.

THIS PART REQUIRES PROCESSING COMPLETELY BY HAND THE CASE $n = 8$.

Algorithm 4

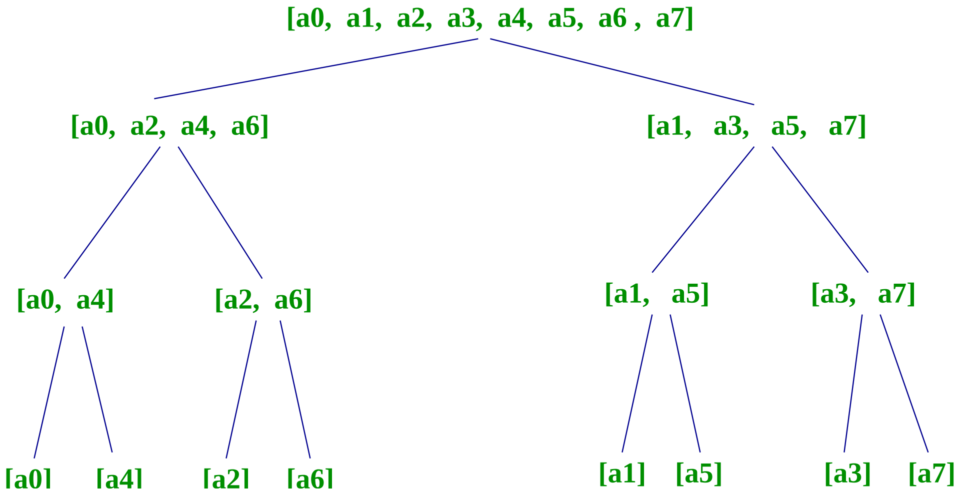

Figure 1: Recursive calls for $n = 8$.

Input: $n = 2^r$, A the array for the coefficient of the polynomial sorted by the *DFT ordering*. $\Omega = [1, \omega, \dots, \omega^{n/2-1}, \dots, \omega^{n-1}]$ an array of the powers of a primitive n-th root of unity ω.

Output: An array $y = [y_0, \dots, y_{n-1}]$ such that y_i is $A(\omega^i)$ by calling $DFT(n, A, \Omega)$.

$DFT(n, A, \Omega)$ ==
  for $s := 1 \cdots r$ **repeat**
  # for each flour of the tree, bottom-up.
    $m := 2^s$
    for $k := 0 \cdots n-1$ **by** m **repeat**
    # for each internal node from left to right
      for $k = 0 \cdots m/2 - 1$ **repeat**
      # combine its two children
        $t := \Omega_k \quad A[k + j + m/2]$
        $u := A[k + j]$
        $A[k + j] := u + t$
        $A[k + j + m/2] := u - t$
  return A

THE FORMULA FOR t IN THE ABOVE ALGORITHM NEEDS TO BE CHECKED.

Remark 8 To get the *DFT ordering* for A, that is $[a_0, a_4, a_2, a_6, a_1, a_5, a_3, a_7]$, observe that this ordering changes

$$000, 001, 010, 011, 100, 101, 011, 111 \tag{59}$$

to

$$000, 100, 010, 110, 001, 101, 110, 111 \tag{60}$$

Remark 9 The ring $\mathbb{Z}$ does not have any *interesting* roots of unity. In order to take advantage of the FFT-based multiplication one could use a modular approach as follows.

- Let f and g be two univariate polynomials in $\mathbb{Z}[x]$. We can assume that all their coefficients are positive. Otherwise, we can restrict to this case by setting
$$f \ = \ f^{+} - f^{-} \quad \text{and} \quad g \ = \ g^{+} - g^{-} \tag{61}$$
where f^{+}, f^{-}, g^{+}, g^{-} are polynomials with positive coefficients and writing
$$f\,g \ = \ f^{+}g^{+} + f^{-}g^{-} - f^{+}g^{-} - f^{-}g^{+} \tag{62}$$
and computing the products $f^{+}g^{+}$, $f^{-}g^{-}$, $f^{+}g^{-}$ and $f^{-}g^{+}$ one after another.

- Let $||f||$ and $||f||$ be the max-norms of f and g. Let d be the degree of $f\,g$. For $k = 0 \cdots d$ a bound for the coefficient c_k of x^k in $f\,g$ is given by
$$|c_k| \ \leq \ (k+1)||f||||f|| \tag{63}$$

- Let M be the biggest value of these bounds. Let $p_1, \ldots, p_r$ be primes such
  - 2^d divides each $p_1 - 1, \ldots, p_r - 1$
  - the product of $p_1, \ldots, p_r$ is bigger than $2\,M$.

- Then we can compute $f\,g$ in each of the $\mathbb{Z}/p_i\mathbb{Z}[x]$ using the FFT-based multiplication.

- Finally we can recombine these modular product by applying the CRA to each coefficient c_k (of x^k) in $f\,g$.

4 Division with remainder using Newton iteration

We combine in this section the results about the fast polynomial multiplication based on the FFT with Newton iteration in order to provide a fast polynomial division with remainder.

4.1 Classical division with remainder

We assume that R is a commutative ring with identity element. We consider the classical algorithm for univariate polynomial division with remainder stated over R instead of a field. However, Algorithm 5 is well defined because we assume that the leading coefficient of the polynomial b is a unit.

Algorithm 5

Input: univariate polynomials $a = \Sigma_0^n \, a_i x^i$ and $b = \Sigma_0^m \, b_i x^i$ in $R[x]$ with respective degrees n and m such that $n \geq m \geq 0$ and b_m is a unit.

Output: the quotient q and the remainder r of a w.r.t. b. Hence $a = bq + r$ and $\deg r < m$.

$r := a$
for $i = n - m, n - m - 1, \ldots, 0$ **repeat**
  if $\deg r = m + i$ **then**
    $q_i := \mathsf{leadingCoefficient}(r) \,/\, b_m$
    $r := r - q_i x^i b$
  else $q_i := 0$
$q := \Sigma_0^{n-m} \, q_i x^i$
return (q, r)

Proposition 5 *Let a and b two univariate polynomials in $R[x]$ with respective degrees n and m such that $n \geq m \geq 0$ and the leading coefficient of b is a unit. Then there exist unique polynomials q and r such that $a = bq + r$ and $\deg r < m$. The polynomials q and r are called the* quotient *and the* remainder *of a w.r.t. to b. Algorithm 5 compute them in $(2m + 1)(n - m + 1) \in \mathcal{O}(n\,m)$ operations in R.*

PROOF ▷ We leave to the reader the proof of the uniqueness of the quotient and the remainder of a w.r.t. to b. So we focus now on the complexity result Consider an iteration of the **for** loop where $\deg r = m + i$ holds at the begining of the loop. Observe that r and $q_i x^i b$ have the same leading coefficient. Since $\deg b = m$, computing the reductum of $q_i x^i b$ requires m operations. Then subtracting the reductum of $q_i x^i b$ to that of r requires again m operations. Hence each iteration of the **for** loop requires at most $2m + 1$ operations in R, since we need also to count 1 for the computation of q_i. The number of **for** loops is $n - m + 1$. Therefore Algorithm 5 requires $(2m + 1)(n - m + 1)$. ◁

Remark 10 One can derive from Algorithm 5 a bound for the coefficients of q and r, which is needed for performing a modular version (based for instance on the CRT). Let $||a||$, $||b||$ and $||r||$ be the max-norm of a, b and r. Let $\mid b_m \mid$ be the absolute value (over $\mathbb{Z}$) or the norm (over $\mathbb{C}$) of the leading coefficient of b. Then we have

$$||r|| \;\leq\; ||a|| \left(1 + \frac{||b||}{\mid b_m \mid}\right)^{n-m+1} \tag{64}$$

4.2 The quotient as a modular inverse

We shall see now that the complexity result of Proposition 5 can be improved. To do so, we will show that q can be computed from a and b by performing essentially one multiplication in $R[x]$. We start with the equation

$$a(x) \;=\; q(x)\, b(x) + r(x) \tag{65}$$

where a, b, q and r are in the statement of Proposition 5. Replacing x by $1/x$ and multiplying the equation by x^n leads to the new equation:

$$x^n\, a(1/x) \;=\; \left(x^{n-m} q(1/x)\right)\; \left(x^m\, b(1/x)\right) \;+\; x^{n-m+1} \left(x^{m-1}\, r(1/x)\right) \tag{66}$$

In Equation (66) each of the rational fractions $a(1/x)$, $b(1/x)$, $q(1/x)$ and $r(1/x)$ is multiplied by x^e such that e is an upper bound for the degree of its denominator. So Equation (66) is in fact an equation in $R[x]$. Definition 5 and Proposition 6 highlight this fact. Then Proposition 7 and Theorem 9 suggest Algorithm 6 which provides an efficient way to compute the inverse of a univariate polynomial in x modulo a power of x. Finally we obtain Algorithm 7 for fast division with remainder based on Equation (66).

Definition 5 *For a univariate polynomial* $p = \Sigma_0^d \, p_i x^i$ *in* $R[x]$ *with degree* d *and an integer* $k \geq d$, *the* reversal of order k *of* p *is the polynomial denoted by* $\mathrm{rev}_k(p)$ *and defined by*

$$\mathrm{rev}_k(p) \;=\; x^k \, p(1/x) \;=\; \Sigma_{k-d}^{k} \, p_{k-i} x^i. \tag{67}$$

When $k = d$ *the polynomial* $\mathrm{rev}_k(p)$ *is simply denoted by* $\mathrm{rev}(p)$. *Hence we have*

$$\mathrm{rev}(p) \;=\; p_d + p_{d-1}x + \cdots + p_1 x^{d-1} + p_0 x^d. \tag{68}$$

Proposition 6 *With* a, b, q *and* r *as abovce, we have*

$$\mathrm{rev}_n(a) \;\equiv\; \mathrm{rev}_{n-m}(q)\; \mathrm{rev}_m(b) \qquad \bmod x^{n-m+1} \tag{69}$$

Proof $\triangleright$ Indeed with Definition 5 Equation (66) reads

$$\mathrm{rev}_n(a) \;=\; \mathrm{rev}_{n-m}(q)\; \mathrm{rev}_m(b) + x^{n-m+1}\; \mathrm{rev}_{m-1}(r) \tag{70}$$

leading to the desised result. $\triangleleft$

Remark 11 If R is a field then we know that $\mathrm{rev}_{n-m}(q)$ is invertible modulo x^{n-m+1}. Indeed, $\mathrm{rev}_{n-m}(q)$ has constant coefficient 1 and thus the gcd of $\mathrm{rev}_{n-m}(q)$ and x^{n-m+1} is 1. The case where R is not a field leads also to a simple and surprising solution as we shall see in the next section.

4.3 Modular inverses using Newton iteration

Let R be a commutative ring with identity element. Given $f \in R[x]$ and $\ell \in \mathbb{N}$ such that $f(0) = 1$ compute the polynomials $g \in R[x]$ such that

$$f\, g \equiv 1 \mod x^\ell \;\text{ and }\; \deg g < \ell. \tag{71}$$

In Proposition 7 we observe that if there is a solution, then it is unique.

Proposition 7 *If Equation (71) has a solution* $g \in R[x]$ *with degree less than* ℓ *then it is unique.*

PROOF $\triangleright$ Indeed, let g_1 and g_2 be solutions of Equation (71). Then the product $f(g_1 - g_2)$ is a multiple of x^ℓ. Since $f(0) = 1$ then $g_1(0) - g_2(0)$ must be 0. Hence there is a constant $c \in R$ and polynomials h_1, h_2 with degree less than $\ell - 1$ such that

$$g_1(x) \;=\; h_1(x)\,x + c \;\text{ and }\; g_2(x) \;=\; h_2(x)\,x + c \tag{72}$$

It follows that $f(h_1 - h_2)$ is a multiple of $x^{\ell-1}$. By repeating the same argument we show that $h_1(0) = h_2(0)$. Then by induction on ℓ we obtain $g_1 = g_2$. $\triangleleft$

Remark 12 Since Equation (71) is an equation in $R[x]/\langle x^\ell \rangle$, a solution of this equation an be viewed as an approximation of a more general problem. Think of truncated Taylor expansions! So let us recall from numerical analysis the celebrated Newton iteration and let $\phi(g) = 0$ be an equation that we want to solve, where a $\phi : \mathbb{R} \longmapsto \mathbb{R}$ is a differentiable function. From a suitable initial approximation g_0, the sequence, called *Newton iteration step*,

$$g_{i+1} \;=\; g_i - \frac{\phi(g_i)}{\phi'(g_i)} \tag{73}$$

allows to compute subsequent approximations and converge toward a desired solution. In our case we have $\phi(g) = 1/g - f$ and the Newton iteration step is

$$g_{i+1} \;=\; g_i - \frac{1/g_i - f}{-1/{g_i}^2} \;=\; 2g_i - f\,{g_i}^2. \tag{74}$$

Theorem 9 *Let R be a commutative ring with identity element. Let f be a polynomial in $R[x]$ such that $f(0) = 1$. Let $g_0, g_1, g_2, \ldots$ be the sequence of polynomials defined for all $i \geq 0$ by*

$$\left\{ \begin{array}{rcl} g_0 & = & 1 \\ g_{i+1} & \equiv & 2g_i - f\,{g_i}^2 \quad \bmod x^{2^{i+1}} \end{array} \right. \tag{75}$$

Then for $i \geq 0$ we have

$$f\,g_i \;\equiv\; 1 \quad \bmod x^{2^i} \tag{76}$$

PROOF $\triangleright$ By induction on $i \geq 0$. For $i = 0$ we have $x^{2^i} = x$ and thus

$$f\, g_i \;\equiv\; f(0)\, g_0 \;\equiv\; 1 \times 1 \;\equiv\; 1 \quad \bmod x^{2^i} \tag{77}$$

For the induction step we have

$$\begin{array}{rcll} 1 - f\, g_{i+1} & \equiv & 1 - f(2g_i - f\, {g_i}^2) & \bmod x^{2^{i+1}} \\ & \equiv & 1 - 2\, f\, g_i + f^2\, {g_i}^2 & \bmod x^{2^{i+1}} \\ & \equiv & (1 - f\, g_i)^2 & \bmod x^{2^{i+1}} \\ & \equiv & 0 & \bmod x^{2^{i+1}} \end{array} \tag{78}$$

Indeed $f\, g_i \;\equiv\; 1 \quad \bmod x^{2^i}$ means that x^{2^i} divides $1 - f\, g_i$. Thus $x^{2^{i+1}} = x^{2^i + 2^i} = x^{2^i}\, x^{2^i}$ divides $(1 - f\, g_i)^2$. $\triangleleft$

Algorithm 6

Input: $f \in R[x]$ such that $f(0) = 1$ and $\ell \in \mathbb{N}$.

Output: $g \in R[x]$ such that $f\, g \equiv 1 \mod x^\ell$

$g_0 := 1$
$r := \lceil \log_2(\ell) \rceil$
for $i = 1 \cdots r$ **repeat**
  $g_i := (2g_{i-1} - f\, {g_{i-1}}^2) \quad \bmod x^{2^i}$
return g_r

Notation 1 We denote by $\mathbf{M}(n)$ an upper bound for the number of operations in R required to multiply two polynomials of $R[x]$ with degree less than n. We know that $M(n) = 2\, n^2$ is such a bound. Moreover if n is a power of 2, say 2^r, and if R possesses primitive 2^{r+1}-th roots of unity, then we can choose $M(n) = \mathcal{O}(n \log n)$.

Theorem 10 *Algorithm 6 computes the inverse of f modulo x^ℓ. Moreover, if ℓ be a power of 2, then Algorithm 6 requires $\mathcal{O}(\mathbf{M}(\ell))$ operations in R.*

Proof $\triangleright$ Theorem 9 tell us that Algorithm 6 computes the inverse of f modulo x^{2^r}. Since x^ℓ divides x^{2^r}, the result is also valid modulo x^ℓ. We do not give here a proof of the complexity result and we refer to Theorem 9.4 in [vzGG99]. In fact we prefer to give a very brutal explanation about it.

To understand the complexity result one needs to point out the following relation for $i = 1 \cdots r$.

$$g_i \ \equiv \ g_{i-1} \mod x^{2^{i-1}} \tag{79}$$

Indeed, by virtue of Theorem 9 we have

$$\begin{array}{rcll} g_i & \equiv & 2g_{i-1} - f\, {g_{i-1}}^2 & \mod x^{2^i} \\ & \equiv & 2g_{i-1} - f\, {g_{i-1}}^2 & \mod x^{2^{i-1}} \\ & \equiv & g_{i-1}(2 - f\, g_{i-1}) & \mod x^{2^{i-1}} \\ & \equiv & g_{i-1}(2-1) & \mod x^{2^{i-1}} \\ & \equiv & g_{i-1} & \mod x^{2^{i-1}} \end{array} \tag{80}$$

Therefore when computing g_i we only care about power of x in the range $x^{2^{i-1}} \cdots x^{2^i}$. This says that

- half of the computation of g_r is made during the last iteration of the **for** loop,
- a quater is made when computing g_{r-1} etc.

Now recall that

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots \ = \ 1 \tag{81}$$

So **roughly** the cost of the algorithm is in the order of magnitude of the cost of the last iteration. which consists of

- two multiplications of polynomials with degree less than 2^r,
- a multiplication of a polynomial (with degree less than 2^r) by a constant,
- truncations modulo x^{2^r}
- a subtraction of polynomials with degree less than 2^r.

leading to $\mathcal{O}(\mathbf{M}(2^r))$ operations in R. $\triangleleft$

Remark 13 Once again in Algorithm 6 for $i = 1 \cdots r$ we have

$$g_i \equiv g_{i-1} \mod x^{2^{i-1}} \tag{82}$$

So when implementing Algorithm 6 one should be extremely careful in not recomputing the *low terms* of g_i that come from g_{i-1}.

Remark 14 Algorithm 6 can be adapted to the case where $f(0)$ is a unit different from 1 by initializing g_0 to the inverse of $f(0)$ instead of 1. If $f(0)$ is not a unit, then no inverse of f modulo x^ℓ exists. Indeed $f\,g \equiv 1 \mod x^\ell$ implies $f(0)g(0) = 1$ which says that $f(0)$ is a unit.

4.4 Division with remainder using Newton iteration

Algorithm 7

> **Input:** $a, b \in R[x]$ with $b \neq 0$ monic.
>
> **Output:** q, r such that $a = b\,q + r$ and $\deg r < \deg b$.
>
> $n := \deg a$
> $m := \deg b$
> **if** $n < m$ **then**
> $q := 0$
> $r := a$
> **else**
> $f := \mathrm{rev}_m(b)$
> g inverse of f modulo x^{n-m+1}
> $q := \mathrm{rev}_n(a)\,g \mod x^{n-m+1}$
> $q := \mathrm{rev}_{n-m}(q)$
> $r := a - b\,q$
> **return** (q, r)

Theorem 11 *Let R be a commutative ring with identity element. Let a and b be univariate polynomials over R with respective degrees n and m such that $n \geq m \geq 0$ and $b \neq 0$ monic. Algorithm 7 computes the quotient and the remainder of a w.r.t. b in $4\,\mathbf{M}(n-m) + \mathbf{M}(m) + \mathcal{O}(m)$ operations in R*

We do not give here a proof and we refer to Theorem 9.5 in [vzGG99]. However, roughly speaking, Algorithm 7 consists essentially in

- c multiplications (where c is a constant) to compute g (by virtue of Theorem 9),
- one multiplication to compute q,
- one multiplication to compute r.

If several divisions by a given b needs to be performed then we may precompute the inverse of $\mathrm{rev}_m(b)$ modulo some powers $x, x^2, \ldots$ of x. Assuming that R possesses suitable primitive roots of unity, ee can also save their DTF.

5 P-adic Expansions and Approximations

The goal of this section is to formalize in an algebraic point of view the notions of expansions and approximations well known in numerical analysis.

Definition 6 *Let R be an Euclidean domain and let $p \in R$ be a prime element. Let $a \in E$ be an element. We say that a has* degree zero w.r.t. *p and we write* $\deg(a, p) = 0$ *if for every positive integer $k \in \mathbb{N}$ the quotient of a w.r.t. p^k is zero. Assume from now on that a does not have degree zero w.r.t. p. If for every positive integer ℓ there exists a positive integer $k > \ell$ such that the quotient of a w.r.t. p^k is not zero, then we say a has* infinite degree w.r.t. *p and we write* $\deg(a, p) = +\infty$*, otherwise we say that a has* finite degree w.r.t. *p and we write* $\deg(a, p) < +\infty$*. Assume furthermore that* $\deg(a, p) < +\infty$ *holds. The largest positive integer $k \in \mathbb{N}$ such that the quotient of a w.r.t. p^k is not zero is called the* degree of a w.r.t. *p and is denoted by* $\deg(a, p)$.

Definition 7 *The Euclidean size δ of an Euclidean domain R is said* regular *if for every $a, b \in R$ with $b \neq 0$ there exists a couple $(q, r) \in E \times E$ such that*

1. $a = bq + r$,
2. $r \neq 0 \quad \Rightarrow \quad \delta(r) < \delta(b)$,
3. $q \neq 0 \quad \Rightarrow \quad \delta(q) < \delta(a) \;\text{ and }\; \delta(b) \leq \delta(a)$.

Proposition 8 *Let R be an Euclidean domain with a regular Euclidean size δ. Let p be a prime element. Then for every non-zero $a \in E$ there exists an integer $d \in N$ and elements $a_0, a_1, \ldots, a_d \in E$ such that*

1. $a = a_d p^d + \cdots + a_1 p + a_0$,
2. $a_d \neq 0$,
3. *for every* $i = 0 \cdots d$ *we have* $\deg(a_i, p) = 0$.

The sequence $(a_0, a_1, \ldots, a_d)$ *is called a* p-adic *expansion of* a *w.r.t.* p.

PROOF $\triangleright$ Let (q_0, a_0) be a couple quotient-remainder given as in Definition 7 when dividing a w.r.t. p.

First we prove that a_0 must have degree 0 w.r.t. p. This is obvious if $a_0 = 0$. So let us assume that $a_0 \neq 0$ holds. This implies $\delta(p) > \delta(a_0)$. If a_0 would have positive degree k w.r.t. p, then, by the properties of a regular Euclidean size, we would have $\delta(a_0) \geq \delta(p^k)$. Since $\delta(p^k) \geq \delta(p)$ holds (by definition of an Euclidean domain) we would have a contradiction with $\delta(p) > \delta(a_0)$. Therefore in any case we have $\deg(a_0, p) = 0$.

Next, let us prove the proposition for those elements a which have quotient 0 w.r.t. p. Observe that if one quotient of a w.r.t. p is zero then we have $\delta(a) < \delta(p)$. Hence every quotient of a w.r.t. p is null. (Otherwise we would have $\delta(p) \leq \delta(a)$ by the properties of a regular Euclidean size.) Since $a = a_0$ and since a_0 has degree 0 w.r.t. p, the element a has degree 0 w.r.t. p and (a_0) is a p-adic expansion of a w.r.t. p. Therefore the proposition is proved in the case of the elements a which have quotient 0 w.r.t. p.

We can assume now that a does not have quotient 0 w.r.t. p and thus $q_0 \neq 0$ and $\delta(q_0) < \delta(a)$ hold. Then let (q_1, a_1) be a couple quotient-remainder given as in Definition 7 when dividing q_0 w.r.t. p. If $q_1 \neq 0$ then we have $\delta(q_1) < \delta(q_0)$ and let (q_2, a_2) be a couple quotient-remainder given as in Definition 7 when dividing q_1 w.r.t. p. Continuing in this manner we obtain a finite sequence $a_0, a_1, \ldots, a_d \in R$ and a finite sequence $q_0, q_1, \ldots, q_d \in R$ such that we have

$$\begin{array}{rclcrcl}
a & = & q_0 p + a_0 & \text{with} & q_0 & \neq & 0 \\
q_0 & = & q_1 p + a_1 & \text{with} & q_1 & \neq & 0 \\
q_1 & = & q_2 p + a_2 & \text{with} & q_2 & \neq & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
q_{d_2} & = & q_{d-1} p + a_{d-1} & \text{with} & q_{d-1} & \neq & 0 \\
q_{d-1} & = & q_d p + a_d & \text{with} & q_d & = & 0
\end{array}$$

This process stops since we have

$$\delta(a) > \delta(q_0) > \delta(q_1) > \delta(q_2) > \cdots > \delta(q_{d-1}) \tag{83}$$

and since the Euclidean size of an element has value in $\mathbb{N} \cup \{-\infty\}$. Clearly we have

$$a = a_d p^d + \cdots + a_1 p + a_0.$$

Moreover $a_d = q_{d-1} \neq 0$ holds. In addition, for every $i = 0 \cdots d$ we have $\deg(a_i, p) = 0$ by the first part of this proof. $\lhd$

Remark 15 Proposition 8 applies in the ring $\mathbb{Z}$ of integers and the ring $\mathbf{k}[x]$ of univariate polynomials over a field $\mathbf{k}$. It extends also for the ring $R[y]$ of univariate polynomials over an Euclidean domain R with a regular Euclidean size δ and a prime element $p \in R$. Indeed, one can apply Proposition 8 to each coefficient of a polynomial $\phi \in R[y]$.

As we shall see now, it extends to a ring $R[y]$ of univariate polynomials over a commutative ring with identity element and to the element $p = y - g$ where g is an element of R.

Proposition 9 *Let R be a commutative ring with identity element. Let $\phi \in R[y]$ be a polynomial of degree n and let $g \in R$ be an element. We define $p = y - g$. There exists a unique sequence $\phi_0, \phi_1, \ldots, \phi_n \in R$ such that we have*

$$\phi = \phi_0 + \phi_1 p + \cdots + \phi_n p^n. \tag{84}$$

The above expression is called the Taylor expansion *of ϕ at g,*

Proof $\rhd$ By induction on n. If $n = 0$, the result is clear. Otherwise let q and r be the quotient and the remainder in the division of ϕ by p. We have

$$\phi = qp + r \quad \text{and} \quad r \in R.$$

Since $r = \phi(g)$ the conclusion (existence and unicity) follows by the induction hypothesis. $\lhd$

Remark 16 We can characterize the Taylor expansion of ϕ at g by means of the notion of a formal derivative.

Definition 8 *Let R be a commutative ring with identity element. For*

$$\phi = \phi_0 + \phi_1 y + \phi_2 y^2 + \cdots + \phi_n y^n$$

we define the formal derivative *of ϕ by*

$$\phi' = \phi_1 + 2\phi_2 y + \cdots + n\phi_n y^{n-1}. \tag{85}$$

Remark 17 The terminology formal derivative comes from the fact that we do not make use of the notion of a limit to define this derivative. Indeed, if R is a finite ring, then this would not make clear sense. However we retrieve the usual properties of derivatives.

Proposition 10 *Let R be a commutative ring with identity element. Let $\phi, \psi \in R[y]$ and let $a, b \in R$. Then the following properties hold*

1. $(a\phi + b\psi)' = a\phi' + b\phi'$,
2. $(\phi\psi)' = \phi'\psi + \phi\psi'$,
3. $(\phi(\psi))' = \psi'\phi'(\psi)$.

PROOF $\triangleright$ See Lemma 9.19 in [vzGG99]. $\triangleleft$

Proposition 11 *Let R be a commutative ring with identity element. Let $\phi \in R[y]$ and let $g \in R$. There exists a polynomial $\psi \in R[y]$ such that we have*

$$\phi(y) = \phi(g) + \phi'(g)(y-g) + \psi(y)(y-g)^2. \tag{86}$$

PROOF $\triangleright$ Follows easily from Proposition 9. $\triangleleft$

Remark 18 Proposition 9 admits several generalizations. First one can use a monic polynomial p of degree $d > 1$. In this case the coefficients $\phi_0, \phi_1, \ldots, \phi_n$ are polynomials of degree less than d. Another generalization, given by Proposition 12, is for multivariate polynomials. To understand this generalization, it is important to make the following three observations.

- First, in Proposition 11, one can consider g as a new indeterminate, and then the result holds for a polynomial $\phi \in R[y, g]$.

- Secondly, the notion of a formal derivative of a univariate polynomial (over a commutative ring with identity element) leads naturally to the notion of a partial derivative of a multivariate polynomial w.r.t. a variable. We will not detail this here, since this is quite straightforward
- The quantity $\psi(y-g)^2$ in Relation (86) lies in the ideal $\langle y-g\rangle^2$.

Proposition 12 *Let R be a commutative ring with identity element. We consider the ring of multivariate polynomials $P = R[x_1, \ldots, x_r][y_1, \ldots, y_r]$. We define $\mathbf{x} = (x_1, \ldots, x_r)$ and $\mathbf{y} = (y_1, \ldots, y_r)$. For every polynomial $\phi \in R[\mathbf{x}]$ there exists a polynomial $\psi \in P$ lying in the ideal $\langle y_1, \ldots, y_r\rangle^2$ such that*

$$\phi(x_1+y_1, \ldots, x_r+y_r) = \phi(\mathbf{x}) + \Sigma_{j=1}^{j=r} \frac{\partial \phi}{\partial x_j}(\mathbf{x}) y_j + \psi. \tag{87}$$

Remark 19 Using the vocabulary of numerical analysis, Propositions 11 and 12 provides a Taylor expansion of ϕ of order 2. This leads to the following definition.

Definition 9 *Let R be a commutative ring with identity element and let $\mathcal{I}$ be an ideal of R. Let $a, b \in R$ be elements and k be a positive integer. We say that a is an $\mathcal{I}$-*adic approximation *of b at order k if we have*

$$a \equiv b \mod \mathcal{I}^k. \tag{88}$$

Remark 20 We conclude by two useful properties of ideal-adic approximations.

Proposition 13 *Let R be an Euclidean domain with a regular Euclidean size δ. Let $p \in R$ be a prime element. Let $a \in R$ be an element and let $(a_0, a_1, \ldots, a_d)$ be a p-adic expansion of a w.r.t. p. For every positive integer $k \leq d+1$ we define*

$$a^{(k)} = a_0 + a_1 p + \cdots a_{k-1} p^{k-1} \tag{89}$$

*Then $a^{(k)}$ is a p-*adic approximation *of a at order k, that is*

$$a^{(k)} \equiv a \mod p^k. \tag{90}$$

PROOF $\triangleright$ Indeed, we have

$$a - a^{(k)} = a_k p^k + a_{k+1} p^{k+1} + \cdots + a_d p^d = (a_k + a_{k+1} p + \cdots + a_d p^{d-k}) p^k$$

$\triangleleft$

Proposition 14 *Let R be a commutative ring with identity element and let $\mathcal{I}$ be an ideal of R. Let $a, b \in R$ be elements, let k be a positive integer and let $\phi \in R[y]$ be a polynomial. Then we have*

$$a \equiv b \mod \mathcal{I}^k \quad \Rightarrow \quad \phi(a) \equiv \phi(b) \mod \mathcal{I}^k \tag{91}$$

That is, if a is an $\mathcal{I}$-adic approximation of b at order k, then $\phi(a)$ is an $\mathcal{I}$-adic approximation of $\phi(b)$ at order k.

PROOF $\triangleright$ This is simply because of the ring-homomorphism between R and $R/\mathcal{I}$. $\triangleleft$

6 Symbolic Newton Iteration

6.1 Linear Newton Iteration

Theorem 12 *Let R be an Euclidean domain with a regular Euclidean size δ. Let $p \in R$ be a prime element. Let $\phi \in R[y]$ and let $a, a_0 \in R$ such that*

1. $\phi(a) = 0$,
2. $a \equiv a_0 \mod p$,
3. $\phi'(a_0) \not\equiv 0 \mod p$.

If a is unknown, but ϕ and a_0 are known then we can compute a p-adic expansion of a. Therefore we compute a **exactly**.

PROOF $\triangleright$ Let $(a_0, a_1, \ldots, a_d)$ be a p-adic expansion of a w.r.t. p. Let $k < d+1$ be a positive integer. By Proposition 13, the element

$$a^{(k)} = a_0 + a_1 p + \cdots a_{k-1} p^{k-1} \tag{92}$$

is a p-adic approximation of a at order k. By Proposition 11 there exists a polynomial $\psi \in R[y]$ such that

$$\phi(a^{(k)} + a_k p^k) = \phi(a^{(k)}) + \phi'(a^{(k)}) a_k p^k + \psi(a^{(k)} + a_k p^k)(a_k p^k)^2. \tag{93}$$

Since we have

$$a \equiv a^{(k)} + a_k p^k \mod p^{k+1}$$

we deduce from Proposition 14

$$\phi(a) \equiv \phi(a^{(k)} + a_k p^k) \mod p^{k+1} \tag{94}$$

Since $\phi(a) = 0$ this shows that $\phi(a^{(k)} + a_k p^k)$ is in the ideal generated by p^{k+1}. Similarly, $\phi(a^{(k)})$ is in the ideal generated by p^k. Therefore we can divide $\phi(a^{(k)} + a_k p^k)$ and $\phi(a^{(k)})$ by p^k, leading to

$$\frac{\phi(a^{(k)} + a_k p^k)}{p^k} = \frac{\phi(a^{(k)})}{p^k} + \phi'(a^{(k)})a_k + \psi(a^{(k)} + a_k p^k)(a_k)^2 p^k. \tag{95}$$

Now observe that

$$\frac{\phi(a^{(k)} + a_k p^k)}{p^k} \equiv 0 \equiv \psi(a^{(k)} + a_k p^k)(a_k)^2 p^k \mod p. \tag{96}$$

Let us denote by Φ_p the canonical homomorphism from R to $R/\langle p \rangle$. Then we obtain

$$\Phi_p(\frac{\phi(a^{(k)})}{p^k}) = -\Phi_p(\phi'(a^{(k)}))\Phi_p(a_k) \tag{97}$$

Now, since $a \equiv a_0 \mod p$ holds we have

$$\Phi_p(\phi'(a^{(k)})) \equiv \Phi_p(\phi'(a_0)) \mod p \tag{98}$$

Finally, since $\phi'(a_0) \not\equiv 0 \mod p$ holds we can solve Equatiion 97 for $\Phi_p(a_k)$. $\triangleleft$

Theorem 13 *Let R be a commutative ring with identity element and let $\mathcal{I}$ be a finitely generated ideal of R. Let r be a positive integer. Let $f_1, \ldots, f_n \in R[x_1, \ldots, x_r]$ be n multivariate polynomials in the r variables $x_1, \ldots, x_r$. Let $a_1, \ldots, a_r \in R$ be elements. Let U be the Jacobian matrix of $f_1, \ldots, f_n$ evaluated at $(a_1, \ldots, a_r)$. That is, U is the $n \times r$ matrix defined by*

$$U = (u_{ij}) \quad \text{where} \quad u_{ij} = \frac{\partial f_i}{\partial x_j}(a_1, \ldots, a_r) \tag{99}$$

We assume that the following properties hold

1. *for every $i = 1 \cdots n$ we have $f_i(a_1, \ldots, a_r) \equiv 0 \mod \mathcal{I}$.*

2. *the Jacobian matrix U is left-invertible.*

Then, for every positive integer ℓ we can compute $a_1^{(\ell)}, \ldots, a_r^{(\ell)} \in R$ such that

1. *for every $i = 1, \ldots, n$ we have $f_i(a_1^{(\ell)}, \ldots, a_r^{(\ell)}) \equiv 0 \mod \mathcal{I}^\ell$,*

2. *for every $j = 1, \ldots, r$ we have $a_j^{(\ell)} \equiv a_j \mod \mathcal{I}$.*

Proof $\triangleright$ We proceed by induction on $\ell \geq 1$. For $\ell = 1$ the claim follows from the hypothesis of the theorem. So let $\ell \geq 1$ be such that the claim is true. Hence there exist $a_1^{(\ell)}, \ldots, a_r^{(\ell)} \in R$ such that

$$f_i(a_1^{(\ell)}, \ldots, a_r^{(\ell)}) \equiv 0 \mod \mathcal{I}^\ell, \quad i = 1, \ldots, n \tag{100}$$

and

$$a_j^{(\ell)} \equiv a_j \mod \mathcal{I}, \quad i = 1, \ldots, r \tag{101}$$

Since $\mathcal{I}$ is finitely generated, then so is $\mathcal{I}^\ell$ and let $g_1, \ldots, g_s \in R$ such that

$$\mathcal{I}^\ell = \langle g_1, \ldots, g_s \rangle \tag{102}$$

Therefore, for every $i = 1, \ldots, n$, there exist $q_{i1}, \ldots q_{is} \in R$ such that

$$f_i(a_1^{(\ell)}, \ldots, a_r^{(\ell)}) \;=\; \Sigma_{k=1}^{k=s} q_{ik} g_k \tag{103}$$

For each $j = 1, \ldots, r$ we want to compute $B_j \in R$ such that

$$a_j^{(\ell+1)} = a_j^{(\ell)} + B_j \tag{104}$$

is the desired *next approximation*. We impose $B_j \in \mathcal{I}^\ell$ so let $b_{j1}, \ldots, b_{js} \in R$ be such that

$$B_j \;=\; \Sigma_{k=1}^{k=s} b_{jk} g_k \tag{105}$$

Using Proposition 12 we obtain

$$\begin{array}{rcl} f_i(a_1^{(\ell+1)}, \ldots, a_r^{(\ell+1)}) & = & f_i(a_1^{(\ell)} + B_1, \ldots, a_r^{(\ell)} + B_r) \\ & \equiv & f_i(a_1^{(\ell)}, \ldots, a_r^{(\ell)}) + \Sigma_{j=1}^{j=r} u_{ij}^{(\ell)} B_j \mod \mathcal{I}^{\ell+1} \\ & \equiv & \Sigma_{k=1}^{k=s} q_{ik} g_k + \Sigma_{j=1}^{j=r} u_{ij}^{(\ell)} B_j \mod \mathcal{I}^{\ell+1} \\ & \equiv & \Sigma_{k=1}^{k=s} q_{ik} g_k + \Sigma_{j=1}^{j=r} u_{ij}^{(\ell)} \left(\Sigma_{k=1}^{k=s} b_{jk} g_k\right) \mod \mathcal{I}^{\ell+1} \\ & \equiv & \Sigma_{k=1}^{k=s} \left(q_{ik} + \Sigma_{j=1}^{j=r} u_{ij}^{(\ell)} b_{jk}\right) g_k \mod \mathcal{I}^{\ell+1} \end{array} \tag{106}$$

where $\left(u_{ij}^{(\ell)}\right)$ is the Jacobian matrix of $(f_1, \ldots, f_n)$ at $(a_1^{(\ell)}, \ldots, a_r^{(\ell)})$.

Hence, solving for $a_1^{(\ell+1)}, \ldots, a_r^{(\ell+1)} \in R$ such that

$$f_i(a_1^{(\ell+1)}, \ldots, a_r^{(\ell+1)}) \equiv 0 \mod \mathcal{I}^{\ell+1}$$

leads to solving the system of linear equations:

$$q_{ik} + \Sigma_{j=1}^{j=r} u_{ij}^{(\ell)} b_{jk} \equiv 0 \mod \mathcal{I} \tag{107}$$

for $k = 1, \ldots, s$ and $i = 1, \ldots, n$.

Now using $a_j^{(\ell)} \equiv a_j \mod \mathcal{I}$ for $j = 1, \ldots, r$ we obtain

$$u_{ij}^{(\ell)} \equiv u_{ij} \mod \mathcal{I} \tag{108}$$

Therefore the system linear equations given by Relation (107) has solutions. $\triangleleft$

6.2 Quadratic Newton Iteration

Let R be a commutative ring R with units. Let $p \in R$ and $\phi \in R[y]$. We are concerned with solving the equation

$$\phi(g) \;=\; 0. \tag{109}$$

The following Proposition 15 states that Relation (111) computes better and better approximations of solutions of Relation (109) that is, approximations modulo higher and higher powers of p. W.r.t. Theorem 12 this proposition offers a significant improvement: the accuracy doubles at each step, leading to the so-called *Quadratic Symbolic Newton Iteration.*

Proposition 15 *Let $\phi \in R[y]$ and let $g, h \in R$ be such that*

$$\phi(g) \equiv 0 \mod p \tag{110}$$

Assume that $\phi'(g)$ is invertible modulo p and let denote by $\phi'(g)^{-1}$ its inverse. Consider an element $h \in R$ such that

$$h \equiv g - \phi(g)\, \phi'(g)^{-1} \mod p^2 \tag{111}$$

Then the following assertions hold

1. $\phi(h) \equiv 0 \mod p^2$

2. $h \equiv g \mod p$

3. $\phi'(h)$ *is invertible modulo* p

PROOF $\triangleright$ Since $\phi'(g)$ is invertible modulo p then $\phi'(g)$ is invertible modulo p^2. (This is easy to check: Exercise!) Hence Relation 111 makes sense. In fact, Algorithm 6 computes $\phi'(g)^{-1} \mod p^2$ given $\phi'(g)^{-1} \mod p$.

Since p divides p^2, the following congruence holds

$$h \equiv g - \phi(g)\,\phi'(g)^{-1} \mod p \tag{112}$$

Since $\phi(g) \equiv 0 \mod p$, this leads to

$$h \equiv g \mod p \tag{113}$$

and we have proved the second claim. Let us prove the first one. Using the Taylor expansion of ϕ at h and the fact that p^2 divides $(h-g)^2$ we obtain

$$\begin{array}{rcl} \phi(h) & \equiv & \phi(g) + \phi'(g)(h-g) \mod p^2 \\ & \equiv & \phi(g) - \phi'(g)\phi(g)\,\phi'(g)^{-1} \mod p^2 \\ & \equiv & 0 \mod p^2 \end{array} \tag{114}$$

proving the first claim. Finally, observe that $h \equiv g \mod p$ implies $\phi'(h) \equiv \phi'(g) \mod p$. Indeed ϕ' is a polynomial in $R[x]$. Therefore, $\phi'(h)$ is invertible modulo p, since $\phi'(g)$ is invertible modulo p, by hypothesis.

$\triangleleft$

Remark 21 Intuitively, Proposition 15 says that if g is a good approximation of a zero of ϕ then h is a better approximation, at least *twice as accurate*.

Remark 22 Proposition 15 leads to an algorithm provided that we can

- check whether $\phi'(g)$ is invertible modulo p and
- compute its inverse modulo p in case.

Then for computing the inverse of $\phi'(h)$ modulo p the idea is to apply the algorithm to itself! That is, to the particular case of

$$\phi(g) = 1/g - f \tag{115}$$

This leads to Algorithm 8. Finally recall that for a prime element p in an Euclidean domain R, the computation of $\phi'(g)^{-1}$ modulo p can be done by the Extended Euclidean Algorithm.

Algorithm 8

Input: $\ell \in \mathbb{N}$, $\phi \in R[y]$, $p \in R$ and $g_0, s_0 \in R$ such that $\phi(g_0) \equiv 0 \mod p$ and $s_0\, \phi'(g_0) \equiv 1 \mod p$.

Output: $g \in R$ such that $\phi(g) \equiv 0 \mod p^\ell$ and $g \equiv g_0 \mod p$.

$r := \lceil \log_2(\ell) \rceil$
for $i = 1 \cdots r-1$ **repeat**
  $g_i \equiv g_{i-1} - \phi(g_{i-1}) s_{i-1} \mod p^{2^i}$
  $s_i \equiv (2s_{i-1} - \phi'(g_i)\, {s_{i-1}}^2) \mod p^{2^i}$
$g := g_{r-1} - \phi(g_{r-1}) s_{r-1} \mod p^\ell$
return g

Theorem 14 *Algorithm 8 is correct.*

PROOF $\triangleright$ Let $g_r \equiv g_{r-1} - \phi(g_{r-1}) s_{r-1} \mod p^{2^r}$. Then we have

$$g \equiv g_r \mod p^\ell \tag{116}$$

In other words g_r is a *solution of higher precision.* Then proving the theorem turns into proving the invariants for $i = 0 \cdots r$

1. $g_i \equiv g_0 \mod p$,
2. $\phi(g_i) \equiv 0 \mod p^{2^i}$,
3. if $i < r$ then $s_i\, \phi'(g_i) \equiv 1 \mod p^{2^i}$.

The case $i = 0$ is clear and we assume that $i > 0$ holds. Then by induction hypothesis $p^{2^{i-1}}$ divides $\phi(g_{i-1})$ and $s_{i-1} - \phi'(g_{i-1})^{-1}$. Then p^{2^i} divides their product, that is

$$\phi(g_{i-1}) s_{i-1} \equiv \phi(g_{i-1}) \phi'(g_{i-1})^{-1} \mod p^{2^i}. \tag{117}$$

Now we calculate

$$\begin{aligned} g_i &\equiv g_{i-1} - \phi(g_{i-1}) s_{i-1} \mod p^{2^i} \\ &\equiv g_{i-1} - \phi(g_{i-1}) \phi'(g_{i-1})^{-1} \mod p^{2^i} \end{aligned} \tag{118}$$

Now applying Proposition 15 with $g = g_{i-1}$, $h = g_i$ and $m = p^{2^{i-1}}$ we obtain the first two invariants. In addition, this leads to

$$g_i \equiv g_{i-1} \mod p^{2^{i-1}} \tag{119}$$

for $i < r$. This implies

$$\phi'(g_i)^{-1} \equiv \phi'(g_{i-1})^{-1} \equiv s_{i-1} \mod p^{2^{i-1}} \tag{120}$$

Now, s_i is obtained by one Newton step for inversion (see Algorithm 6). Therefore the third invariant follows. $\triangleleft$

7 Hensel Lifting

7.1 Linear Hensel Lifting

Theorem 15 *Let R be a commutative ring with identity element. Let f, g_0, h_0 be univariate polynomials in $R[x]$ and let $m \in R$. We assume that the following relation holds*

$$f \equiv g_0 h_0 \mod m \tag{121}$$

We assume also that g_0 and h_0 are relatively prime modulo m, that is there exists $s, t \in R$ such that

$$sg_0 + th_0 \equiv 1 \mod m \tag{122}$$

Then, for every integer ℓ there exist $g^{(\ell)}, h^{(\ell)} \in R[x]$ such that we have

1. $f \equiv g^{(\ell)}h^{(\ell)} \mod m^\ell$,
2. $g_0 \equiv g^{(\ell)} \mod m$.

Proof $\triangleright$ We apply Theorem 13 with

$$\mathcal{I} = \langle m \rangle,\ n = 1,\ r = 2,\ f_1(x_1, x_2) = x_1 x_2 - f. \tag{123}$$

Thus we have

$$U = (u_{11}\ u_{12}) = (g_0\ h_0) \tag{124}$$

Observe that this matrix is left-invertible modulo m if and only if g_0, h_0 are relatively prime modulo m. Indeed, the matrix U is left-invertible modulo m iff there exists a matrix

$$U^{-1} = \begin{pmatrix} s \\ t \end{pmatrix}$$

such that $U U^{-1} \equiv (1) \mod m$, that is

$$sg_0 + th_0 \equiv 1 \mod m \tag{125}$$

We prove now the claim of the theorem. Assume we have computed $g^{(\ell)}, h^{(\ell)} \in R$ such that $f \equiv g^{(\ell)} h^{(\ell)} \mod m^\ell$ and $g \equiv g^{(\ell)} \mod m$ hold. We want to compute $g^{(\ell+1)}, h^{(\ell+1)} \in R$. Let v be such that

$$f_1(g^{(\ell)}, h^{(\ell)}) = g^{(\ell)} h^{(\ell)} - f = v m^\ell \tag{126}$$

We look for $g_\ell, h_\ell \in R$ such that

$$g^{(\ell+1)} = g^{(\ell)} + g_\ell m^\ell \quad \text{and} \quad h^{(\ell+1)} = h^{(\ell)} + h_\ell m^\ell \tag{127}$$

Following the proof of Theorem 13 we are led to solve the equation

$$v + g_\ell h_0 + h_\ell g_0 \equiv 0 \mod m \tag{128}$$

A solution of this equation is

$$\begin{pmatrix} g_\ell \\ h_\ell \end{pmatrix} = \begin{pmatrix} -sv \\ -tv \end{pmatrix} \tag{129}$$

This proves the claim of the theorem. $\lhd$

Remark 23 The proof of Theorem 15 provides a solution (g_ℓ, h_ℓ) to Equation (128). It is desirable to add constraints such that Equation (128) has a unique soulution. In particular, one would like to guarantee that for every integer ℓ the polynomials $g^{(\ell)}$ and g_0 have the same degree. This problem is solved in the following sub-section where we study *Linear Diophantine Equations.*

Definition 10 *Let R be a commutative ring with identity element. A* linear Diophantine equation *over R is an equation of the form*

$$f_1 s_1 + \cdots + f_n s_n = a \tag{130}$$

where $f_1, \ldots, f_n, a$ are given in R and where $s_1, \ldots, s_n$ are unknowns in R.

Theorem 16 *Let R be an Euclidean domain and let $f, g, h, a \in R$ such that $h = \gcd(f, g)$. We consider the linear Diophantine equation*

$$sf + tg = a. \tag{131}$$

Then the following hold

(*i*) *Equation (131) has a solution if and only if* h *divides* a.

(*ii*) *If* $h \neq 0$ *and if* $(s_0, t_0) \in R^2$ *is a solution of Equation (131) then every other solution is of the form*

$$(s_0 + r\frac{g}{h}, t_0 - r\frac{f}{h}) \quad \text{where} \quad r \in R. \tag{132}$$

(*iii*) *If* $R = F[x]$ *where* F *is a field,* $h \neq 0$, *and if Equation (131) is solvable, where*

$$\deg f + \deg g - \deg h > \deg a \tag{133}$$

then there is a unique solution $(s_0, t_0) \in R^2$ *of Equation (131) such that*

$$\deg s < \deg g - \deg h \quad \text{and} \quad \deg t < \deg f - \deg h. \tag{134}$$

PROOF $\triangleright$ First we prove (*i*). If $(s_0, t_0) \in R^2$ is a solution of Equation (131), then h which divides $s_0 f + t_0 g$, divides also a. Conversly, we assume that $h = \gcd(f, g)$ divides a. The claim is trivial if $h = 0$. (Indeed, this implies $f = g = 0$ and also $a = 0$.) Otherwise, let $(s_0, t_0) \in R^2$ be computed by the Extended Euclidean Algorithm applied to (f, g), such that we have $s_0 f + t_0 g = h$. Then $(s, t) = (s_0 a/h, t_0 a/h)$ is a solution of Equation (131).

Now we prove (*ii*). Assume $h \neq 0$ and let $(s_0, t_0) \in R^2$ is a solution of Equation (131). Since $h \neq 0$, then f/h and g/h are coprime. Let (s, t) be in R^2. Then we have

$$\begin{array}{rcl} sf + tg = a & \Longleftrightarrow & (s - s_0)f + (t - t_0)g = 0 \\ & \Longleftrightarrow & (s - s_0)(f/h) = -(t - t_0)(g/h) \\ & \Longleftrightarrow & (s - s_0)(f/h) = (t_0 - t)(g/h) \\ & \Longleftrightarrow & (g/h) \mid (s - s_0) \ \text{ and } \ (f/h) \mid (t_0 - t) \\ & \Longleftrightarrow & (\exists r \in R) \ (s - s_0) = r(g/h) \ \text{ and } \ (t_0 - t) = r(f/h) \\ & \Longleftrightarrow & (\exists r \in R) \ s = s_0 + r(g/h) \ \text{ and } \ t = t_0 - r(f/h). \end{array}$$

Finally, we prove (*iii*). Let $(s_1, t_1) \in R^2$ be a solution of Equation (131). Let q and t_0 be the quotient and the remainder of t_1 w.r.t. f/h. Hence we have

$$t_1 = f/hq + t_0 \quad \text{and} \quad \deg t_0 < \deg f - \deg h. \tag{135}$$

We define

$$s_0 = s_1 + qg/h. \tag{136}$$

Observe that

$$(s_0, t_0) = (s_1, t_1) + q(g/h, -f/h)$$

solves Equation (131) too. By Relation (135) and by hypothesis we have

$$\deg(t_0 g) < \deg f + \deg g - \deg h \ \text{ and } \ \deg a < \deg f + \deg g - \deg h$$

leading to

$$\deg s_0 + \deg f = \deg s_0 f = \deg(a - t_0 g) < \deg f + \deg g - \deg h.$$

Therefore we have

$$\deg t_0 < \deg f - \deg h \ \text{ and } \ \deg s_0 < \deg g - \deg h.$$

This proves the existence. Let us prove the unicity. Let (s, t) be a solution of Equation (131). We know that there exists $r \in R$ such that $s = s_0 + rg/h$. Since $\deg s_0 < \deg g - \deg h$ holds, if $r \neq 0$, we have

$$\deg s \geq \deg(g/h) = \deg g - \deg h.$$

This implies the uniquemess. $\triangleleft$

Theorem 17 *Let R be a commutative ring with identity element. Let f, g_0, h_0 be univariate* **monic** *polynomials in $R[x]$. Let $m \in R$ be such that R/m is a field. We assume that the following relation holds*

$$f \ \equiv \ g_0 \, h_0 \quad \mod m \tag{137}$$

and that g_0 and h_0 are relatively prime modulo m. Then, for every integer ℓ there exist **unique monic** *polynomials $g^{(\ell)}, h^{(\ell)} \in R[x]$ such that we have*

1. $f \ \equiv \ g^{(\ell)} h^{(\ell)} \mod m^{\ell}$,
2. $g_0 \ \equiv \ g^{(\ell)} \mod m$,
3. $h_0 \ \equiv \ h^{(\ell)} \mod m$.

PROOF $\triangleright$ By induction on $\ell \geq 1$. The clain is clear for $\ell = 1$. So let us assume it is true for $\ell \geq 1$ and consider monic polynomials $g^{(\ell+1)}, h^{(\ell+1)} \in R[x]$ satisfying

$$f \ \equiv \ g^{(\ell+1)} h^{(\ell+1)} \quad \mod m^{\ell+1} \tag{138}$$

and

$$g_0 \equiv g^{(\ell+1)} \mod m \text{ and } h_0 \equiv h^{(\ell+1)} \mod m. \tag{139}$$

Such polynomials $g^{(\ell+1)}, h^{(\ell+1)}$ exist by Theorem 15. (The fact that they can be chosen monic is left to the reader as an exercise.) Observe that we have

$$f \equiv g^{(\ell+1)} h^{(\ell+1)} \mod m^\ell \tag{140}$$

So the induction hypothesis leads to

$$g^{(\ell)} \equiv g^{(\ell+1)} \mod m^\ell \text{ and } h^{(\ell)} \equiv h^{(\ell+1)} \mod m^\ell \tag{141}$$

Hence there exist polynomials $q_g, q_h \in (R/m)[x]$ such that

$$g^{(\ell+1)} = g^{(\ell)} + m^\ell q_g \text{ and } h^{(\ell+1)} = h^{(\ell)} + m^\ell q_h. \tag{142}$$

Since f, g_0, h_0 are given to be monic, it is easy to prove that for $k \geq 1$ we have

$$\deg q_g < \deg g_0 \text{ and } \deg q_h < \deg h_0 \tag{143}$$

In addition, observe that combining Equation (138) and Equation (142) we obtain

$$f \equiv g^{(\ell)} h^{(\ell)} + \left(g^{(\ell)} q_h + h^{(\ell)} q_g \right) m^\ell \mod m^{\ell+1} \tag{144}$$

The induction hypothesis shows that $f - g^{(\ell)} h^{(\ell)}$ is a multiple of m^ℓ. Then we obtain

$$g_0 q_h + h_0 q_g \equiv \frac{f - g^{(\ell)} h^{(\ell)}}{m^\ell} \mod m. \tag{145}$$

By Theorem 16 and Equation (143), the Equation (145) has a unique solution. $\lhd$

7.2 Quadratic Hensel Lifting

Let R be a commutative ring with identity element. Let f, g, h be univariate polynomials in $R[x]$ and let $m \in R$. We assume that the following relation holds

$$f \equiv g\, h \mod m \tag{146}$$

We aim to lift this factorization modulo m into a factorization modulo m^2. Hence we want to compute $\widehat{g}$ and $\widehat{h}$ such that

$$f \equiv \widehat{g}\, \widehat{h} \mod m^2 \tag{147}$$

We will refer to this problem as the *factorization lifting problem*. To do so we will assume that we are given two polynomials s and t such that

$$sg + th \equiv 1 \mod m \tag{148}$$

Hence we will say that g and h are coprime modulo m. When R/m is a field then we can compute s and t by applying the Extended Euclidean Algorithm in $R[x]/m$.

Proposition 16 *A solution to the factorization lifting problem of Equation 147 is given by*

$$\widehat{g} = g + t\,e \;\text{ and }\; \widehat{h} = h + s\,e \;\text{ where }\; e = f - g\,h \tag{149}$$

PROOF ▷ We have the following relations.

$$\begin{array}{rcll} f - \widehat{g}\widehat{h} & = & f - (g + t\,e)\,(h + s\,e) & \\ & = & f - g\,h - g\,s\,e - h\,t\,e - s\,t\,e^2 & \\ & = & e - (s\,g + t\,h)\,e - s\,t\,e^2 & \\ & = & (1 - s\,g + t\,h)\,e - s\,t\,e^2 & \\ & \equiv & (1 - s\,g + t\,h)\,e - s\,t\,e^2 & \mod m^2 \end{array} \tag{150}$$

By hypothesis

$$e \equiv 0 \mod m \quad \text{and} \quad (1 - s\,g + t\,h) \equiv 0 \mod m \tag{151}$$

Hence both $(1 - s\,g + t\,h)\,e$ and e^2 are mutiple of m^2. It follows that

$$f - \widehat{g}\widehat{h} \equiv 0 \mod m^2 \tag{152}$$

◁

Remark 24 Observe that Proposition 16 works also if f, g, h are constants of R. In other words, f, g, h do not need to be polynomials! The factorization lifting problem and its solution by Proposition 16 are valid in any commutative ring with identity element. But for this solution to be an algorithm one needs to be able to compute the Bézout coefficients s and t.

Example 7 Consider $f = x^4 - 1$, $g = x - 2$, $h = x^3 + 2\,x^2 - x - 2$ and $m = 5$. We have the following computation in AXIOM

```
N := NonNegativeInteger


   (1)  NonNegativeInteger
                                                        Type: Domain

Z := Integer


   (2)  Integer
                                                          Type: Domain

U := UnivariatePolynomial(x,Z)


   (3)  UnivariatePolynomial(x,Integer)
                                                          Type: Domain

m: Z := 5


   (4)  5
                                                         Type: Integer

f: U := x^4 - 1


         4
   (5)  x  - 1
                              Type: UnivariatePolynomial(x,Integer)

g: U := x - 2


   (6)  x - 2
                              Type: UnivariatePolynomial(x,Integer)

h: U := x^3 + 2 *x^2 -x -2
```

```
         3     2
   (7)  x  + 2x  - x - 2
                      Type: UnivariatePolynomial(x,Integer)

e := f - g*h


          2
   (8)  5x  - 5
                      Type: UnivariatePolynomial(x,Integer)

K5 := PrimeField(5)


   (9)  PrimeField
                                                   Type: Domain

U5 := UnivariatePolynomial(x,K5)


   (10)  UnivariatePolynomial(x,PrimeField 5)
                                                   Type: Domain

f5: U5 := f :: U5


          4
   (11)  x  + 4
                   Type: UnivariatePolynomial(x,PrimeField 5)

g5: U5 := g :: U5


   (12)  x + 3
                   Type: UnivariatePolynomial(x,PrimeField 5)
```

```
h5: U5 := h ::U5


         3     2
   (13)  x  + 2x  + 4x + 3
                    Type: UnivariatePolynomial(x,PrimeField 5)

res := extendedEuclidean(g5,h5)

                   2
   (14)  [coef1= 2x  + 3x + 4,coef2= 3,generator= 1]

s5: U5 := res.coef1


          2
   (15)  2x  + 3x + 4
                    Type: UnivariatePolynomial(x,PrimeField 5)

t5: U5 := res.coef2


   (16)  3
                    Type: UnivariatePolynomial(x,PrimeField 5)

g_^ := g + t5 :: U * e


           2
   (17)  15x  + x - 17
                         Type: UnivariatePolynomial(x,Integer)

h_^ := h + s5 :: U * e


           4      3      2
   (18)  10x  + 16x  + 12x  - 16x - 22
                         Type: UnivariatePolynomial(x,Integer)
```

```
factor(f -  g_^  * h_^ )


                                     4       3     2
   (19)  - 25(x - 1)(x + 1)(6x  + 10x  + 7x  - 10x - 15)
                Type: Factored UnivariatePolynomial(x,Integer)
```

Remark 25 We see with Example 7 that the degrees of $\widehat{g}$ and $\widehat{h}$ are higher that those of g and h. In particular

$$\deg \widehat{g} + \deg \widehat{h} \ > \ \deg f \tag{153}$$

which may not look satisfactory! To overcome this problem we need to use division with remainder in $R[x]$. We will take advantage of the following.

Proposition 17 *Let a and b be in $R[x]$ such that b is monic. Let q and r be the quotient and the remainder of a w.r.t. b. Let m be in R. If $a \equiv 0 \mod m$ then*

$$q \equiv 0 \mod m \quad \text{and} \quad r \equiv 0 \mod m \tag{154}$$

also hold.

PROOF $\triangleright$ There exists $a^* \in R[x]$ such that $a = m\, a^*$. Let q^* and r^* be the quotient and the remainder of a^* w.r.t. b. Then we have

$$a^* \ = \ q^*\, b + r^* \quad \text{and} \quad \deg r^* \ < \ \deg b \tag{155}$$

Hence we have

$$a \ = \ m\, q^*\, b + m\, r^* \quad \text{and} \quad \deg\,(m\, r^*) \ < \ \deg b \tag{156}$$

Since the quotient and the remainder of a w.r.t. b are unique we have

$$q \ = \ m\, q^* \quad \text{and} \quad r \ = \ m\, r^* \tag{157}$$

$\triangleleft$

Definition 11 *Let f, g, h, s, t be polynomials in $R[x]$ and let $m \in R$. We say that (f, g, h, s, t) is a* liftable quintet *modulo m if the following conditions hold*

- $f \equiv g\,h \mod m$,
- $s\,g + t\,h \equiv 1 \mod m$,
- *h is monic,*
- $\deg f = \deg g + \deg h$,
- $\deg s < \deg h$,
- $\deg t < \deg g$.

Algorithm 9

Input: $f, g, h, s, t \in R[x]$ and $m \in R$ such that (f, g, h, s, t) is a liftable quintet modulo m.

Output: $g^*, h^*, s^*, t^* \in R[x]$ such that (f, g^*, h^*, s^*, t^*) is a liftable quintet modulo m^2 such that $u^* \equiv u \mod m$ for $u = g, h, s, t$ and $\deg g^* = \deg g$ and $\deg h^* = \deg h$.

$e := f - g\,h \mod m^2$
$(q,\, r) := \mathsf{divisionWithRemainder}(s\,e,\, h)$
$(q,\, r) := (q \mod m^2,\, r \mod m^2)$
$g^* := (g + t\,e + q\,g) \mod m^2$
$h^* := (h + r) \mod m^2$
$b := (s\,g^* + t\,h^* - 1) \mod m^2$
$(c,\, d) := \mathsf{divisionWithRemainder}(s\,b,\, h^*)$
$(c,\, d) := (c \mod m^2,\, d \mod m^2)$
$s^* := s - d \mod m^2$
$t^* := t - t\,b - c\,g^* \mod m^2$
return (g^*, h^*, s^*, t^*)

Theorem 18 *Algorithm 9 works correctly. Let $n = \deg f$. If $R = \mathbb{Z}$ and all input polynomials have max-norm less than m^2 then it requires $\mathcal{O}(\mathbf{M}(n)\,\mathbf{M}(\log m))$ word operations.*

PROOF $\triangleright$ We shall prove the following claims about g^* and h^*:

1. $f - g^* h^* \equiv 0 \mod m^2$,
2. $g^* \equiv g \mod m$,
3. $h^* \equiv h \mod m$,
4. h^* monic and of the same degree as h,
5. g^* of the same degree as g.

We calculate modulo m^2:

$$\begin{array}{rcl} f - g^* h^* & \equiv & f - (g + te + qg)(h + se - qh) \\ & = & f - gh - gse + gqh - teh - tese + teqh - qgh - qgse + qgqh \\ & = & f - gh - gse - teh - tese + teqh - qgse + qgqh \\ & = & e - (sg + th)e - ste^2 + qe(th + sg) + ghq^2 \\ & = & (1 - sg - th)(f - gh) - ste^2 + qe(th + sg) + ghq^2 \end{array} \tag{158}$$

By assumption we have

- $1 - sg - th \equiv 0 \mod m$
- $f - gh \equiv 0 \mod m$

Moreover

$$e \equiv 0 \mod m \quad \text{and} \quad se \equiv qh + r \mod m^2 \tag{159}$$

by Proposition 17 imply

$$q \equiv 0 \mod m \tag{160}$$

This leads to

$$f - g^* h^* \equiv 0 \mod m^2. \tag{161}$$

Now the relation defining g^*

$$g^* = (q + t\,e + q\,g) \mod m^2 \tag{162}$$

together with $e \equiv 0 \mod m$ and $q \equiv 0 \mod m$ imply

$$g^* \equiv g \mod m \tag{163}$$

Similarly we have

$$h^* \equiv h \mod m \tag{164}$$

Now observe that by definition of r we have

$$\deg r < \deg h \tag{165}$$

Then since

$$h^* = (h + r) \mod m^2 \tag{166}$$

we obtain

$$h \text{ monic} \quad \text{and} \quad \deg h = \deg h^* \tag{167}$$

These last relations together with

$$f \equiv g^* h^* \mod m^2 \tag{168}$$

imply

$$\begin{aligned} \deg g^* &= \deg f - \deg h^* \\ &= \deg f - \deg h \\ &= \deg g \end{aligned} \tag{169}$$

This concludes this proof. We refer to Theorem 15.11 in [vzGG99] for the other claims of the theorem. $\lhd$

Example 8 Consider $f = x^4 - 1$, $g = x - 2$, $h = x^3 + 2\,x^2 - x - 2$ and $m = 5$. Using Proposition 16 we computed

$$\widehat{g} = 15x^2 + x - 17 \quad \text{and} \quad \widehat{h} = 10x^4 + 16x^3 + 12x^2 - 16x - 22 \tag{170}$$

Now using the improved formula of Algorithm 9 we have the following AXIOM session

```
N := NonNegativeInteger


   (1)  NonNegativeInteger
                                                    Type: Domain

Z := Integer
```

```
   (2)  Integer
                                                         Type: Domain

U := UnivariatePolynomial(x,Z)


   (3)  UnivariatePolynomial(x,Integer)
                                                         Type: Domain

m: Z := 5


   (4)  5
                                                        Type: Integer

f: U := x^4 - 1


         4
   (5)  x  - 1
                            Type: UnivariatePolynomial(x,Integer)

g: U := x - 2


   (6)  x - 2
                            Type: UnivariatePolynomial(x,Integer)

h: U := x^3 + 2 *x^2 -x -2


         3     2
   (7)  x  + 2x  - x - 2
                            Type: UnivariatePolynomial(x,Integer)

e := f - g*h
```

```
          2
   (8)  5x  - 5
                   Type: UnivariatePolynomial(x,Integer)

K5 := PrimeField(5)


   (9)  PrimeField 5
                                            Type: Domain

U5 := UnivariatePolynomial(x,K5)


   (10)  UnivariatePolynomial(x,PrimeField 5)
                                            Type: Domain

f5: U5 := f :: U5


          4
   (11)  x  + 4
              Type: UnivariatePolynomial(x,PrimeField 5)

g5: U5 := g :: U5


   (12)  x + 3
              Type: UnivariatePolynomial(x,PrimeField 5)

h5: U5 := h ::U5


          3     2
   (13)  x  + 2x  + 4x + 3
              Type: UnivariatePolynomial(x,PrimeField 5)

res := extendedEuclidean(g5,h5)
```

```
                  2
   (14)  [coef1= 2x  + 3x + 4,coef2= 3,generator= 1]
Type: Record(coef1: UnivariatePolynomial(x,PrimeField 5),
coef2: UnivariatePolynomial(x,PrimeField 5),
generator: UnivariatePolynomial(x,PrimeField 5))
```

```
s5: U5 := res.coef1
```

```
          2
   (15)  2x  + 3x + 4
              Type: UnivariatePolynomial(x,PrimeField 5)
```

```
s: U := s5 :: U
```

```
          2
   (16)  2x  + 3x + 4
                   Type: UnivariatePolynomial(x,Integer)
```

```
t5: U5 := res.coef2
```

```
   (17)  3
              Type: UnivariatePolynomial(x,PrimeField 5)
```

```
t: U := t5 :: U
```

```
   (18)  3
                   Type: UnivariatePolynomial(x,Integer)
```

```
R25 := IntegerMod(25)
```

```
   (19)  IntegerMod 25
                                          Type: Domain
```

```
U25 := UnivariatePolynomial(x,R25)


   (20)  UnivariatePolynomial(x,IntegerMod 25)
                                                Type: Domain

e25 := f::U25 - g::U25 * h::U25


           2
   (21)  5x  + 20
             Type: UnivariatePolynomial(x,IntegerMod 25)

e : U := e25 :: U


           2
   (22)  5x  + 20
                   Type: UnivariatePolynomial(x,Integer)

res := monicDivide(s*e, h)


                                                2
   (23)  [quotient= 10x - 5,remainder= 80x  + 75x + 70]
Type: Record(quotient: UnivariatePolynomial(x,Integer),
             remainder: UnivariatePolynomial(x,Integer))

q25: U25 := res.quotient :: U25


   (24)  10x + 20
             Type: UnivariatePolynomial(x,IntegerMod 25)

r25: U25 := res.remainder :: U25
```

```
           2
   (25)  5x  + 20
              Type: UnivariatePolynomial(x,IntegerMod 25)
```

```
g_* := (g::U25 + t::U25 * e25 + q25 * (g::U25)) :: U
```

```
   (26)  x + 18
                    Type: UnivariatePolynomial(x,Integer)
```

```
h_* := (h::U25 + r25) :: U
```

```
          3     2
   (27)  x  + 7x  + 24x + 18
                    Type: UnivariatePolynomial(x,Integer)
```

```
(f - g_* * h_*) :: U25
```

```
   (28)  0
              Type: UnivariatePolynomial(x,IntegerMod 25)
```

Theorem 19 *Let R be a commutative ring with identity element, an element $m \in R$ which is not a zero divisor, an integer $\ell \in \mathbb{N}$ and nonzero univariate polynomials $g, h, g^*, h^*, s, t \in R[x]$ such that*

1. $sg + th \equiv 1 \mod m$
2. *the leading coefficients of g and h are not zero divisors modulo m*
3. *the polynomials g and g^* have the same leading coefficients, the same degree and coincide modulo m*
4. *the polynomials h and h^* have the same leading coefficients, the same degree and coincide modulo m*
5. $gh \equiv g^*h^* \mod m^\ell$

Then we have

$$g \equiv g^* \mod m^\ell \quad \text{and} \quad h \equiv h^* \mod m^\ell \tag{171}$$

PROOF ▷ We assume to the contrary that

$$g \not\equiv g^* \mod m^\ell \quad \text{or} \quad h \not\equiv h^* \mod m^\ell \tag{172}$$

Let $1 \leq i < \ell$ be maximal such that m^i both divides $g^* - g$ and $h^* - h$. Thus there exist polynomials $u, v \in R[x]$ such that

$$g^* - g \;=\; u\,m^i \quad \text{and} \quad h^* - h \;=\; v\,m^i \tag{173}$$

and also m does not divide u or v. We may assume that m does not divide u. Now we have

$$\begin{aligned} 0 &\equiv g^*h^* - gh \mod m^\ell \\ &= g^*(h^* - h) + h(g^* - g) \\ &= g^*v\,m^i + hu\,m^i \end{aligned} \tag{174}$$

This means that m^ℓ divides $g^*v\,m^i + hu\,m^i$. In other words there exist $w \in R[x]$ such that

$$(g^*v + hu)m^i \;=\; m^\ell\,w \tag{175}$$

Since m is not a zero divisor in R we have

$$(g^*v + hu) \;=\; m^{\ell-i}\,w \tag{176}$$

which implies

$$g^*v + hu \equiv 0 \mod m \tag{177}$$

Recall also that we have

$$th \equiv 1 - sg \mod m \tag{178}$$

and

$$g^* \equiv g \mod m \quad \text{and} \quad h^* \equiv h \mod m \tag{179}$$

Therefore we obtain the following calculation modulo m

$$\begin{aligned} 0 &\equiv t(g^*v + hu) \\ &\equiv tg^*v + thu \\ &\equiv tg^*v + (1 - sg)u \\ &\equiv (tv - su)g + u \end{aligned} \tag{180}$$

This shows that g divides u modulo m. However from $g^* - g \; = \; u\, m^i$ together with the facts that g^* and g have same leading coefficient and degree shows that

$$\deg u \; < \; \deg g \tag{181}$$

Since the leading coefficient of g is not a zero divisor modulo m we must have

$$u \equiv 0 \mod m \tag{182}$$

A contradiction. Therefore the theorem is proved. $\lhd$

References

[BL82] Bruno Buchberger and Rüdiger Georg Konrad Loos. *Algebraic Simplification*, pages 11–43. Springer Verlag, Wien–New York, 1982.

[DST93] J. Davenport, Y. Siret, and E. Tournier. *Calcul Formel*. Masson, 1993. 2^e édition révisée.

[GCL92] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluver Acadenic Publishers, 1992.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.