Advanced Algorithms	Lecture 6 : March 29, 1995
Lecturer: Johan Håstad	*Scribe: Johan Håstad*

6.1 Factoring polynomials mod 2

First we will study factorization of polynomial mod small primes p and for concreteness we study the case of $p = 2$. Assume that we are given a polynomial $g(x)$ of degree n and we want to decide if is irreducible or factor it into irreducible factors.

First note that we can assume that $g(x)$ is square free, i.e. that it there is no square of any irreducible polynomial that divides $g(x)$. The reason is that computing the greatest common divisor of two polynomial is easy and if $p(x)^2$ divides g then $p(x)$ divides $GCD(g(x), g'(x))$ where $g'(x)$ is the derivative of $g(x)$. The derivative is defined in the formal sense in that the derivative of x^d is dx^{d-1}. Note that over a finite field $g'(x)$ might be identically zero even if $g(x)$ is not a constant. This happens mod 2 when $g(x) = h(x^2)$ for some polynomial h. In this case $GCD(g(x), g'(x)) = g(x)$ so we do not get a nontrivial factorization, however we also have $g(x) = h(x)^2$ so there is no problem.

The key to our algorithm is the mapping

$$f(x) \mapsto f(x)^2 = f(x^2) \mod g(x)$$

of polynomial of degree at most $n - 1$. This is a linear map. We will be interested in the fix-points of this map, i.e. polynomials $f(x)$ such that

$$f(x) = f(x)^2 \mod g(x) \tag{6.1}$$

Just one word of warning, the equality symbol will be used in these notes, even though most equalities are either mod 2 (or some other prime p) or the polynomial $g(x)$ we want to factor, or mod both (which is the most common). Sometimes we will mention this fact, but it will also be dropped to ease readability.

If g is irreducible, then working mod $g(x)$ is actually working in the finite field $GF[2^n]$ and in any field a quadratic equation has at most 2 solutions and it is easy to see that the given equation has exactly two solutions namely the constants 0 and 1.

If g is the product of a number of irreducible polynomials (let us assume two for concreteness), the situation changes. If $g(x) = g_1(x)g_2(x)$ then by the Chinese remainder theorem, each polynomial f mod g can instead be represented as the pair (f_1, f_2) where $f(x) = f_i(x) \mod g_i(x)$. Then it is clear that f solves equation (6.1) iff f_1 solves it mod g_1 and f_2 solves it mod g_2. Since we know the set of solutions mod irreducible polynomial, this implies that we get 4 solutions mod $g(x)$. These are (when written as pairs, i.e. the first component is the remainder mod g_1 and the second mod g_2) $(0,0), (1,1), (0,1)$ and $(1,0)$. The first two solutions correspond to the standard roots $f(x)$ being the constants 0 or 1 while the other two are much more interesting. If $f_{(0,1)}(x)$ is the polynomial corresponding to the third solution then clearly $GCD(g(x), f_{(0,1)}(x)) = g_1(x)$ and our problem is solved. In general it is easy to see that the number of fix-points is given by 2^l where l is the number of irreducible factors in g. Let us see how this works in a couple of examples.

Example 1 *Let $g(x) = x^5 + x^2 + 1$. Then under our mapping*

$$\begin{aligned} 1 &\mapsto 1 \\ x &\mapsto x^2 \\ x^2 &\mapsto x^4 \\ x^3 &\mapsto x^6 = x^3 + x \\ x^4 &\mapsto x^8 = x^2 \cdot x^6 = x^5 + x^3 = x^3 + x^2 + 1 \end{aligned}$$

Thus in the basis 1, x, x^2, x^3, x^4 our mapping is given by the matrix.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Since we are looking for fix-points to the mapping given by M we want to find the null-space of the mapping given by the matrix $M + I$ where I is the identity matrix i.e. we want to study the matrix.

$$M + I = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

It is easy to check that the null-space of this matrix is given by the only vector $(1, 0, 0, 0, 0)$ and thus $x^5 + x^2 + 1$ is irreducible.

Example 2 *Let $g(x) = x^5 + x + 1$. This time*

$$\begin{aligned} 1 &\mapsto 1 \\ x &\mapsto x^2 \\ x^2 &\mapsto x^4 \\ x^3 &\mapsto x^6 = x^2 + x \\ x^4 &\mapsto x^8 = x^2 \cdot x^6 = x^4 + x^3 \end{aligned}$$

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

which gives

$$M + I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The null-space of this matrix is spanned by $(1, 0, 0, 0, 0)$ and $(0, 1, 0, 1, 1)$. The first is the trivial solution, while the second corresponds to the polynomial $x^4 + x^3 + x$. Computing $GCD(x^5 + x + 1, x^4 + x^3 + x)$ is done by the standard Euclidean algorithm giving

$$\begin{aligned} x^5 + x + 1 &= (x+1)(x^4 + x^3 + x) + x^3 + x^2 + 1 \\ x^4 + x^3 + x &= x(x^3 + x^2 + 1) \end{aligned}$$

Finally a division gives $x^5 + x + 1 = (x^3 + x^2 + 1)(x^2 + x + 1)$ which is the complete factorization.

6.2 Other primes p and the analysis of the algorithm

Let us see what happens for other primes p. Many things work completely analogously and in particular we should study the mapping

$$f(x) \mapsto f(x)^p = f(x^p) \bmod g(x)$$

which is a linear map. The set of fix-points to this map is a linear subspace of the polynomials of degree at most $n-1$ of dimension l where l is the number of irreducible factors in g. If g is not irreducible a nontrivial factorization can be found by taking any fix-point $h(x)$ which is not a constant and computing $GCD(h(x)+i, g(x))$ for $i = 0, 1 \ldots p-2$. We need the many GCD computations since we are no longer sure that $h(x)$ is zero mod one of the factors $g_i(x)$.

Let us analyze the running time of this algorithm. For simplicity let us count the number of operations that are done on integers mod p rather than bitoperations. Let us first assume that $p \leq n$. Then using $x^{p(i+1)} = x^p(x^{pi})$ it is easy to see that each column in the matrix M can be computed from the previous in $O(pn)$ operations. To determine the null-space can be done by Gaussian elimination in $O(n^3)$[1] operations and finally each GCD can be done in $O(n^2)$ operations giving a total of $O(pn^2+n^3) = O(n^3)$ operations on integers mod p.

If p is large then the cost $O(pn^2)$ becomes the dominant cost. However in the first step (constructing the matrix M) we can be more efficient. We can precompute $x^p \bmod g(x)$ by $O(\log p)$ squarings and multiplications by x^2 and this can be done in $O(n^2 \log p)$ operations. Each succesive column can then be computed $O(n^2)$ operations. Thus the first part of the algorithm that the determines the number of irreducible factors in $g(x)$ (and in particular if g is irreducible) can be made to run in $O(n^2 \log p + n^3)$ operations. However, the second step is inefficient for large p.

To get an efficient algorithm for large p we need to make an additional observation. The inefficiency comes from the fact that if h is a fix-point, then we know that $h(x)$ mod $g_1(x)$ is one of the numbers $0, 1, \ldots p-1$ but we do not know which. We need to narrow the choice. The trick is to consider $h(x)^{(p-1)/2}$. Since for any $a \neq 0 \bmod p$ we have $a^{(p-1)/2} = \pm 1$ (half of the a's giving each possibility) we are in much better shape. The second step is thus transformed to

- For a random fixpoint h compute $GCD(h^{(p-1)/2} - 1, g(x))$.

The polynomial $h^{(p-1)/2} \bmod g(x)$ can be computing in $O(n^2 \log p)$ operations by the standard method of repeated squaring. It is easy to show that with probability close to one half the above GCD is nontrivial. Thus we get a method that runs in expected time $O(n^2 \log p + n^3)$. Note that the answer we get is always correct since the part determining the number of irreducible factors is deterministic.

---

[1]There are more efficient methods as we will see later in the course, but for practical purposes this is what we use.

[2]Compare to how we computed $a^{p-1} \bmod p$ when we were doing primality tests.

6.3 Special case of squareroots

Let us consider the special case of computing square roots modulo large numbers p. This is a special case of what we have done since if we want to take the squareroot of a then this is the same as factoring $x^2 - a$. We assume that $a \neq 0$, since the case when $a = 0$ is not interesting and also in this case the polynomial x^2 is not squarefree which we assumed. Let us first check that our test for irreducibility works out. Since our $g(x)$ is of degree 2 we just need to find out to what values 1 and x goes under our mapping $f(x) \mapsto f(x)^p$. Clearly 1 goes to 1 and x goes to $x^p = xx^{2((p-1)/2)} = xa^{(p-1)/2}$. Thus the matrix whose nullspace we should find is given by:

$$\begin{pmatrix} 0 & 0 \\ 0 & a^{(p-1)/2} - 1 \end{pmatrix}$$

and this has null-space dimension 2 iff $a^{(p-1)/2} = 1$ which is just the old criteria for a being a square mod p. So far so good!

Now let us try to find the factors. According to our general algorithm we should take a random polynomial h and then compute $GCD(h(x)^{(p-1)/2} - 1, x^2 - a)$. Since h should be a polynomial from the null-space which is not a constant the first values that comes to mind is $h(x) = x$. Let us try this. We need to consider two cases, when $p = 1 \bmod 4$ and $p = 3 \bmod 4$. Let us start with the second. In this case $x^{(p-1)/2} = xx^{2((p-3)/4)} = xa^{(p-3)/4}$. Thus $x^{(p-1)/2} - 1 = xa^{(p-3)/4} - 1$ and it is easy to check that

$$x^2 - a = (xa^{(p-3)/4} - 1)(xa^{(p+1)/4} + a)$$

which implies that we get a solution $x = a^{(3-p)/4} =$ (since $a^{(p-1)/2} = 1$) $a^{(p+1)/4}$. That the solutions is correct can easily be verified directly since

$$(a^{(p+1)/4})^2 = aa^{(p-1)/2} = a.$$

Now let us turn to the case when $p = 1 \bmod 4$. Then

$$x^{(p-1)/2} = x^{2((p-1)/4)} = a^{(p-1)/4}.$$

This is just a constant and this implies that $GCD(x^{(p-1)/2} - 1, x^2 - a)$ is never a nontrivial factor. Thus in this case we have to try another $h(x)$. It turns out that in the case $p = 1 \bmod 4$, there is no known algorithm for extracting square roots that run in deterministic polynomial time. We have to resort to the general procedure of picking a random h.

Thus we have a big difference for extracting square roots depending on $p \bmod 4$. If one thinks about it for a while it turns out that the reason for the difference is -1 is a quadratic non-residue mod p iff $p = 3 \bmod 4$, and -1 is exactly the factor we need to multiply one square root by to get the other.