Berlekamp's algorithm

From Wikipedia, the free encyclopedia

In mathematics, particularly computational algebra, **Berlekamp's algorithm** is a well-known method for factoring polynomials over finite fields (also known as *Galois fields*). The algorithm consists mainly of matrix reduction and polynomial GCD computations. It was invented by Elwyn Berlekamp in 1967. It was the dominant algorithm for solving the problem until the Cantor–Zassenhaus algorithm of 1981. It is currently implemented in many well-known computer algebra systems, including PARI/GP.

Contents

Overview

Berlekamp's algorithm takes as input a square-free polynomial $f(x)$ (i.e. one with no repeated factors) of degree n with coefficients in a finite field $\mathbb{F}_q$ and gives as output a polynomial $g(x)$ with coefficients in the same field such that $g(x)$ divides $f(x)$. The algorithm may then be applied recursively to these and subsequent divisors, until we find the decomposition of $f(x)$ into powers of irreducible polynomials (recalling that the ring of polynomials over a finite field is a unique factorization domain).

All possible factors of $f(x)$ are contained within the factor ring

$$R = \frac{\mathbb{F}_q[x]}{\langle f(x) \rangle}.$$

The algorithm focuses on polynomials $g(x) \in R$ which satisfy the congruence:

$$g(x)^q \equiv g(x) \pmod{f(x)}.$$

These polynomials form a subalgebra of R (which can be considered as an n-dimensional vector space over $\mathbb{F}_q$), called the *Berlekamp subalgebra*. The Berlekamp subalgebra is of interest because the polynomials $g(x)$ it contains satisfy

$$f(x) = \prod_{s \in \mathbb{F}_q} \gcd(f(x), g(x) - s).$$

In general, not every GCD in the above product will be a non-trivial factor of $f(x)$, but some are, providing the factors we seek.

Berlekamp's algorithm finds polynomials $g(x)$ suitable for use with the above result by computing a basis for the Berlekamp subalgebra. This is achieved via the observation that Berlekamp subalgebra is in fact the kernel of a certain $(n+1) \times (n+1)$ matrix over $\mathbb{F}_q$, which is derived from the so-called Berlekamp matrix of the polynomial, denoted $\mathcal{Q}$. If $\mathcal{Q} = [q_{i,j}]$ then $q_{i,j}$ is the coefficient of the j-th power term in the reduction of x^{iq} modulo $f(x)$, i.e.:

$$x^{iq} \equiv q_{i,n}x^n + q_{i,n-1}x^{n-1} + \ldots + q_{i,0} \pmod{f(x)}.$$

With a certain polynomial $g(x) \in R$, say:

$$g(x) = g_n x^n + g_{n-1}x^{n-1} + \ldots + g_0,$$

we may associate the row vector:

$$g = (g_0, g_1, \ldots, g_n).$$

It is relatively straightforward to see that the row vector $g\mathcal{Q}$ corresponds, in the same way, to the reduction of $g(x)^q$ modulo $f(x)$. Consequently a polynomial $g(x) \in R$ is in the Berlekamp subalgebra if and only if $g(\mathcal{Q} - I) = 0$ (where I is the $(n+1) \times (n+1)$ identity matrix), i.e. if and only if it is in the null space of $\mathcal{Q} - I$.

By computing the matrix $\mathcal{Q} - I$ and reducing it to reduced row echelon form and then easily reading off a basis for the null space, we may find a basis for the Berlekamp subalgebra and hence construct polynomials $g(x)$ in it. We then need to successively compute GCDs of the form above until we find a non-trivial factor. Since the ring of polynomials over a field is a Euclidean domain, we may compute these GCDs using the Euclidean algorithm.

Applications

One important application of Berlekamp's algorithm is in computing discrete logarithms over finite fields $\mathbb{F}_{p^n}$, where p is prime and $n \geq 2$. Computing discrete logarithms is an important problem in public key cryptography. For a finite field, the fastest known method is the index calculus method, which involves the factorisation of field elements. If we represent the field $\mathbb{F}_{p^n}$ in the usual way - that is, as polynomials over the base field $\mathbb{F}_p$, reduced modulo an irreducible polynomial of degree n - then this is simply polynomial factorisation, as provided by Berlekamp's algorithm.

Implementation in Computer Algebra Systems

Berlekamp's algorithm may be accessed in the PARI/GP package using the factormod (http://pari.math.u-bordeaux.fr/dochtml/html.stable/Arithmetic_functions.html#factormod) command.

See also

- Polynomial factorisation
- Factorization of polynomials over a finite field and irreducibility tests
- Cantor-Zassenhaus algorithm

References

- Berlekamp, Elwyn R. (1967). "Factoring Polynomials Over Finite Fields". *Bell System Technical Journal* **46**: 1853–1859. MR 0219231 (//www.ams.org/mathscinet-getitem?mr=0219231). BSTJ (http://www.alcatel-lucent.com/bstj/vol46-1967/articles/bstj46-8-1853.pdf) Later republished in: Berlekamp, Elwyn R. (1968). *Algebraic Coding Theory*. McGraw Hill. ISBN 0-89412-063-8.
- Knuth, Donald E (1997). "4.6.2 Factorization of Polynomials". *Seminumerical Algorithms*. The Art of Computer Programming **2** (Third ed.). Reading, Massachusetts: Addison-Wesley. pp. 439–461, 678–691. ISBN 0-201-89684-2.

Retrieved from "http://en.wikipedia.org/w/index.php?title=Berlekamp%27s_algorithm&oldid=588510269"

Categories: Computer algebra | Finite fields

- This page was last modified on 31 December 2013 at 11:39.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.