Berlekamp-Massey algorithm

Canonical name	BerlekampMasseyAlgorithm
Date of creation	2013-03-22 14:28:55
Last modified on	2013-03-22 14:28:55
Owner	mathcam (2727)
Last modified by	mathcam (2727)
Numerical id	7
Author	mathcam (2727)
Entry type	Definition
Classification	msc 15A03
Classification	msc 11B37
Related topic	RecurrenceRelation
Related topic	MapleImplementationOfBerlekampMasseyAlgorithm
Defines	linear recurrent sequence
Defines	minimal polynomial of a sequence
Defines	annihilator

The Berlekamp-Massey algorithm is used for finding the minimal polynomial of a linearly recurrent sequence. The algorithm itself is presented at the end of this article.

Definition 1. *Suppose the infinite sequence a with elements from a field K has the property that there exist constants $c_1, \ldots, c_k$ in K such that, for all $t > k$,*

$$a_t = a_{t-1}c_1 + a_{t-2}c_2 + ... + a_{t-k}c_k.$$

Then a is called a ***linearly recurrent sequence****.*

Definition 2. *Given a linearly recurrent sequence a, suppose $c_0 \ldots c_k \in K$ with $c_0 \neq 0$ satisfy, for all $t > k$,*

$$c_0 a_t = a_{t-1}c_1 + a_{t-2}c_2 + ... + a_{t-k}c_k.$$

Then the polynomial

$$c_0 x^k - c_1 x^{k-1} - c_2 x^{k-2} - ... - c_k$$

is called an ***annihilator*** *for a.*

Proposition 1. *The annihilators of a form an ideal of $K[x]$.*

Definition 3. *Since $K[x]$ is a principal ideal domain, the ideal of a's annihilators have a unique monic generator of minimal degree. This annihilator is called the* ***minimal polynomial*** *of a.*

To find the minimal polynomial, we need to be given an upper bound m on its degree; having done so, the minimal polynomial is uniquely determined by the first $2m$ elements of a (since we need to get m equations to solve for the unknowns $c_1, \ldots c_m$).

There is another way to determine the minimal polynomial, originally presented by Dornstetter, which uses the Euclidean Algorithm. It can be shown that the characteristic polynomial of a sequence is the unique monic polynomial $C(x)$ of least degree for which the infinite product

$$C(x)(a_1 + a_2 x + a_3 x^2 + ...)$$

has finitely many nonzero terms. (In fact, the nonzero terms will have coefficients up to x^{k-1} where k is the degree of C).

We can rewrite this as

$$C(x) \cdot (a_1 + a_2 x + ... + a_{2m} x^{2m-1}) - Q(x) \cdot x^{2m} = R(x)$$

where $R(x)$ is a remainder polynomial of degree ¡ m, and $Q(x)$ is a quotient polynomial. Denote by $A(x)$ the sum $\Sigma_{i=1}^{2m} a_i x^{i-1}$.

This is where the Euclidean Algorithm comes in; if we take the GCD of $A(x)$ and x^{2m}, keeping track of remainders, we get two sequences $P_i(x), Q_i(x)$ such that

$$P_i(x) \cdot A(x) - Q_i(x) \cdot x^{2m}$$

forms a series of polynomials whose degree is decreasing; as soon as this degree is less than m, we have the needed polynomials with $C = P_i, Q = Q_i$.

There is more info about the Extended Euclidean Algorithm in “Modern Computer Algebra” by von zur Gathen and Gerhard.

(Berlekamp’s algorithm proper to come)

The original algorithm is from *Algebraic Coding Theory* by Elwyn R. Berlekamp, McGraw-Hill, 1968. Its application to linearly recurrent sequences was noted by J.L.Massey, in “Shift-register synthesis and BCH decoding”, 1969.