EE793: Home Paper Assignment Problem: Distribution of Linear Complexities of Maps at local points

Submission before End Sem Exam. Maximum team size 2.

February 24, 2022

1 Linear complexity of sequences generated by maps at given values

You are given a map F mapping a finite field $\mathbb{F}$ to itself and a point y in the range of the map giving an equation $y = F(x)$. Since y is in the range of the map F this equation has a solution x. The assignment problem is aimed to study the sequence called as the recurrence sequence and denoted and defined as

$$S(F, y) = \{y, F(y), F^{(2)}(y), \ldots\}$$

Because of the finiteness of the field the sequence repeats previous values after a certain number of terms. The term $F^{(k+1)}(y) = F(F^{(k)}(y))$ and $F^{(2)}(y) = F(F(y))$ is the double composition evaluated at y. Since the sequence $S(F, y)$ repeats, there exist two numbers $r \geq 0$ and N called as the *pre-period* and *period* respectively such that

$$F^{(j+r+N)}(y) = F^{j+r}(y), \text{ for } j = 0, 1, 2, \ldots \tag{1}$$

The sequence is called *periodic* of period N if $r = 0$ and N is the smallest such number in the above equation. The periodicity equation

$$F^{(N)}(y) = y$$

is associated with a polynomial $X^N - 1$. The recurrence relation (1) shows that

$$(X^N - 1)(F^{(j)}(y)) = 0 \text{ for } j = 0, 1, 2, \ldots$$

Hence for a periodic sequence there possibly exists a smallest degree polynomial

$$m(X) = X^m - \sum_{i=0}^{(m-1)} \alpha_i X^i$$

such that the sequence satisfies the *linear recurrence relation* $m(X)(F^{(j)}(y)) = 0$ which is the same as

$$F^{(m+j)}(y) = \sum_{i=0}^{(m-1)} \alpha_i F^{(j+i)}(y) \tag{2}$$

Such a minimal polynomial is unique and its degree is called the *Linear Complexity* (LC) of the sequence $S(F, y)$. In this assignment the objective is to create the data of densities of linear complexities of maps F over finite fields.

1.1 selection of parameters and algorithm

Following tasks are to be performed and the data tabulated.

1. Choose maps F. (In the next section a list of methods are given for choosing the map). F must map a finite field $\mathbb{F}_q$ of size q which is at aleast 12-bits up to a maximum of 16-bits. Results are likely to be better with larger number of bits. You can either choose $\mathbb{F}_p$ with prime p of $n = 12$ to 16 bit size or construct an irreducible polynomial of that much bit size to construct the field $\mathbb{F}_{2^n}$.

2. Choose $k_0 = 2n$, $k_1 = 2n^2$, $k_2 = 2n^3$.

3. You may be required to choose at least three maps F with the same field for getting a variation in the computed data.

4. Get yourself familiar with the Berlekamp-Massey (BM) function in SAGEmath which computes the LC of a sequence.

5. Algorithm:

   (a) Choose y in the field $\mathbb{F}_q$.

   (b) Compute the three sequences $S(F, y)$ upto the terms k_0, k_1 and k_2.

   (c) Find LC of each of these three sequences using the BM function.

   (d) Repeate step c) over 1000 randomly chosen y in the field.

   (e) Repeate step c) over multiple maps F that are chosen.

   (f) Draw the graphs (y, LC) of the data of Linear Complexities of these three sequences for each y and F. You may also draw the histograms of LC plots over the 1000 date points.

1.2 How to choose maps F

Guidelines for choosing varieties of maps F involving 12 to 15 bit fields or Boolean variables.

1. Exponential maps in prime fields. Let $\mathbb{F} = \mathbb{F}_p$ for a prime p. Choose a primitive element ζ in $\mathbb{F}_p^*$. Define $F(x) = \zeta^x \mod p$ for x in $[1, p-1]$. For an arbitrary x_0 let $y = \zeta^{x_0} \mod p$. Find LC of the sequence $S(F, y)$. For different y choose different x_0. By changing, ζ and x_0 you can define different maps F and y. Also by changing p you can get further variations in F and y.

2. Choose any CNF formula $C_1 C_2 \ldots C_n$ in n variables and n clauses. Each clause may contain small number of variables out of n. Define the equations $F(x_1, x_2, \ldots, x_n) = y$ by the satisfiability condition

$$C_i(x_1, x_2, \ldots, x_n) = 1, \text{ for } i = 1, 2, \ldots n$$

   Find the LC of sequences $S(F, y)$ for different F defined by CNF formulas. You can also make variation of y as an arbitrary vector $(y_1, y_2, \ldots, y_n)$ in $\mathbb{F}_2^n$.

3. Use a small scale version of a block cipher $C = E(K, P)$ such that K has 12 to 16 bits. Fix P_0 and define $F(X) = E(X, P_0)$ for a fixed X_0 compute $Y = E(X_0, P_0)$. Now compute the LC of the sequence $S(F, Y)$. You can make variations in F by choosing different P_0 and variations in Y by choosing different X_0.

4. Use a small scale version of a stream cipher

$$x(k+1) = F(x(k))$$
$$w(k) = h(x(k))$$

   Number of states are chosen $2n$ for $n = 12$ to 16. Initial condition $x(0) = (K, IV)$ where K and IV are both n-bits. Choose $k_0 > n$. Define the map F as mapping K to

$$y = (w(k_0), w(k_0 + 1), \ldots, w(k_0 + n - 1))$$

   for a fixed IV and K_0. Compute LC of the sequence $S(F, y)$. Make variations in the map by changing IV and K_0.

5. Define $F : \mathbb{F} \to \mathbb{F}$ by choosing a permutation polynomial in any finite field $\mathbb{F}$. Examples of permutation polynomials are available on internet. Choose finite fields with size $|\mathbb{F}|$ with $n = 12$ to 16 bit length (hence $\mathbb{F}$ muct have 2^n number of elements. Choose a point x_0 in $\mathbb{F}$ and compute $y = F(x_0)$. Then find LC of $S(F, y)$. Make variations in y using different x_0 and in F by choosing different polynomials.

1.3 Computation of the minimal polynomial using BM algorithm

The step c) of the algorithm to compute LC of vector sequences using the BM algorithm is now explained in detail. The problem to be solved in step c) of the algorithm is to find the LC and the minimal polynomial of a vector sequence

$$\hat{y} = \{y_0, y_1, y_2, \ldots, y_{(M-1)}\}$$

using the BM algorithm, where M is any one of $2n, 2n^2, 2n^3$. Here moreover each y_i is a vector

$$y_i = (y_{i1}, y_{i2}, y_{i2}, \ldots, y_{in})^T$$

The BM function takes only the scalar sequence of field numbers

$$\hat{s} = \{s_0, s_1, s_2, \ldots, s_{(M-1)}\}$$

as input and computes the minimal polynomial $m_{\hat{s}}(X)$ of the sequence. The degree m of the minimal polynomial is the LC of the sequence. Hence we need a more general procedure to extend this computation to vector sequences. This extended procedure is carried out as follows:

1. Fix the index i of an i-th component of the vectors in the vector sequence $\hat{y}$. (Say $i = 0$).

2. For i chosen there is a sequence of scalars

   $$\hat{y}(i) = \{y_{0i}, y_{1i}, y_{2i}, \ldots, y_{(M-1)i}\}$$

   Give this sequence $\hat{y}(i)$ as an input to BM algorithm and compute the minimal polynomial $m_i(X)$.

3. Check whether the sequence $\hat{y}$ satisfies the linear recurrence (2) defined by the polynomial $m_i(X)$. If (2) is satisfied, then $m_i(X)$ is the desired minimal polynomial of the vector sequence $\hat{y}$ and its degree is the LC.

4. If the relation (2) is not satisfied for $m_i(X)$, choose another index $j \neq i$ and find a minimal polynomial $m_j(X)$ for the sequence of j-the components of $\hat{y}$. Compute $m(X) = \text{lcm}\,(m_i, m_j)$. Check whether $m(X)$ satisfies (2) for $\hat{y}$.

5. Repeat this selection of an index not considered in previous calculation, computing the minimal polynomial and taking the lcm of previous and the current polynomial until you find a polynomial $m(X)$ which satisfies the relation (2).

The last polynomial will be the minimal polynomial of the vector sequence $\hat{y}$ and its degree is the LC of the vector sequence.

End of Assignment statement