EE793: Home Paper Assignment Problems

Maximum team size 5.
Maximum marks 20.

February 24, 2023

1 Introduction

Home assignment problems are described below. Please follow the process as given below for carrying out the work and submitting the assignment.

1. This assignment document is uploaded on EE793 team's "Assignment" channel. The deadline is strict.
2. Form a team of maximum 5 members and submit the assignment on one of the problems. Write roll numbers and names of all members on the submission document. One of the member will act as the co-ordinator for the team who will submit. Marks for the assignment will be common for all team members.
3. One single PDF file of the submission document should be submitted. Marks will be allotted to each member of the team after evaluation of the submission. Please do not submit zip files. The code should be within the text of the file, not separately attached.
4. Data generated in your submission may be utilised in possible publications. If such publications are made, the team's member's contribution shall be acknowledged.

2 Home assignment problems

2.1 Problem 1: Implicant based Boolean system solver

1. Write a pseudocode for the implicant solver algorithm of a Boolean system of equations and inequalities for computing a complete OG set of implicants. The pseudocode should have pseudocode for implicitly used subroutines which may be called as functions to make the overall pseudocode as structured as possible.
2. Implement the pseudocode in SAGEmath or Macauley. Implementation must be carefully commented to show the pseudocode steps.

3. Implement and explain with pseudocode, a code for generation of random 3-SAT cases.

4. Create and document a record of case studies of solving random all-3-SAT problems with 32 to 128 variables and equations using the implementation. The record should have the case data showing numbers of variables and equations, processor profile (speed, number of cores used, memory used), time taken to solve, number of threads created, number of implicants and predicted number of solutions. Implicant set should be reduced by retaining only largest implicants.

2.2 Problem 2: Linear complexity distribution of SAT problems

1. A brief writeup on Linear Complexity (LC) is given in the next section and will also be discussed in the class.

2. A SAT problem gives rise to a Boolean map $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ with number of variables n and number of clauses m. Satisfiability of the problem is equivalent to the equation

   $$F(x_1, \ldots, x_n) = y = (1, 1, \ldots, 1)^T$$

   which is of the form $C_i(x_1, \ldots, x_n) = 1$ where C_i is the i-th clause in the SAT problem. In this assignment you will choose random 3-SAT problems and determine the distribution of LC of F at y.

3. For the case $m = n$, construct the recursive sequence

   $$S(F, y) = \{y, F(y), F^{(2)}(y), \ldots, F^{(M-1)}(y)\}$$

   for $M = O(n^2)$. Compute LC of $S(F, y)$ over $\mathbb{F}_2$ using the BM function in SAGE.

4. Choose a sample of 1000 such random 3-SAT instances with $n \leq m$ and compute the LCs. Plot the histogram of LC over this sample space.

5. Explain with pseudocode the code for generation of random 3-SAT problems.

6. Write pseudocode to handle the problem of computation of LC in different cases $n = m$, $n < m$. Implement the code in SAGEmath or Macauley.

7. Create and document a record of distribution of LC of random 3-SAT problems. The minimum number of variables shall be between 32 to 64. The record shall contain case data and histogram of the LC over a sample of about 1000 cases, time taken for each case to compute LC and processor profile. The size (length) M of the recursive sequence generated by the map shall be of order $O(n^2)$ where n is the number of variables.

2.2.1 Procedure for computing LC

The procedure is primarily valid only for the square maps $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$. In an intermediate step you will need to extend the procedure for $n < m$.

1. Choose map F. You can choose n as specified.

2. Choose sequence length M.

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

4. Algorithm:

   (a) Choose y in $\mathbb{F}_2^n$.

   (b) Compute the recursive sequences $S(F, y)$ upto the length M.

   (c) Find LC of the component scalar sequences using the BM function.

   (d) Compute the LC of the vector sequence $S(F, y)$.

   (e) For the 3-SAT problem there is only one $y = (1, 1, \ldots, 1)^T$ in $\mathbb{F}_2^n$. Repeat step c) over randomly chosen sample of 1000 maps F by extending this procedure to cases $n < m$.

   (f) Draw the histogram of LC of the sample of random maps. On x-axis you have LC and y-axis the number of maps.

2.3 Problem 3: LC distribution of the exponential map in prime fields

Objective of this problem is to gather the data on distribution of LC of recursively generated sequences $S(F, y, p)$ by the exponential map in the group of units of prime fields $\mathbb{F}_p$. Hence this problem is primarily a number theoretic question and has surprisingly not been much deeply studied. Hence this data is likely to provide interesting insights into properties of the exponential map over $\mathbb{F}_p$. Let p be a prime number and ζ a primitive element in $\mathbb{F}_p$. The map $F(x) : \mathbb{F}_p \to \mathbb{F}_p$ is defined as

$$y = F(x) = \zeta^x \mod p$$

Problem tasks are as follows.

1. Choose a prime p of bit length n between 16 to 32 bits.

2. Fix a primitive element ζ of $\mathbb{F}_p$.

3. For a sample of 1000 random x in $\mathbb{F}_p$ compute $y = F(x)$. Generate the sample of recursive sequences

$$S(\zeta, x) = \{y, F(y), F^{(2)}(y), \ldots, F^{(M-1)}(y)\}$$

   in $\mathbb{F}_p$ for $M = O(n^2)$.

4. Compute the LC of $S(\zeta, x)$ using the BM algorithm over $\mathbb{F}_p$.

5. Draw the histogram of LC.

6. Plot such histograms for 5 different primes p.

2.4 Problem 4: Distribution of LC of quadractic residue function

In this problem the function given is $F(x) = x^2 \mod N$ where $n = pq$ for distinct odd primes p, q. This function is called the quadratic residue function and for $y = F(x)$, x is said to be a square root y modulo N. It is known that the problem of computing square root mod N is equivalent to factoring N. Hence inversion of the map locally is computationlly equivalent to factoring. Assignment tasks are as follows.

1. Choose a pair of distinct primes of length $[n/2]$ bits such that N has length n bits.

2. Choose a random x_0 in $[0, N-1]$ and compute $y_0 = F(x_0)$. For this y_0, compute the sequence

$$S(F, y_0) = \{y_0, F(y_0), F^{(2)}(y_0), \ldots, F^{(M-1)}(y_0)\}$$

   for $M = O(n^2)$.

3. Represent $S(F, y_0)$ as a sequence of n-bit vectors in $\mathbb{F}_2^n$.

4. Compute the LC of $S(F, y_0)$ for a sample of 1000 random x_0 using the BM algorithm described in the next section.

5. Draw the histogram of LC for the sample.

6. Repeat the above procedure and collect histogram data for five different pairs of primes p, q such that bit length of N is between 16 to 48 bits. Write clearly the parameters p, q for each histogram. Write your observations about the statistics when p, q are far separated such as when p is close to $N^(1/3)$ or when p, q are twin primes.

3 Linear complexity of sequences generated by maps at given values

You are given a map F mapping $\mathbb{F}_2^n$ 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.

3.1 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 $m(X) = \operatorname{lcm}(m_i, m_j)$. 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