ECE F344 - Information Theory and Coding Assignment 2

Group 22

Kushaal Tummala	2018AAPS0422H
Sai Vamshi Kamaraju	2018AAPS0420H
Padharthi Sai Sridhar	2018AAPS0472H
Sushant Kunchala	2018AAPS0411H
V Abhinav Sai Venkat	2018AAPS0451H

Question:

Write a computer program that takes in a polynomial with coefficients in GF(q) and returns whether it is irreducible?

THEORY:

A polynomial f(x) in F[x] is said to be reducible if f(x) = a(x)b(x), where a(x), b(x) are elements of F[x] and deg a(x) and deg b(x) are both smaller than deg f(x). If f(x) is not reducible, it is called irreducible. An irreducible monic polynomial of degree at least one is called a prime polynomial. It is helpful to compare a reducible polynomial with a positive integer that can be factorized into a product of prime numbers. Any monic polynomial in f(x) can be factorized uniquely into a product of irreducible monic polynomials(prime polynomials). Prime polynomials of every degree exist over every Galois Field.

So the idea is to divide the polynomial f(x) given with all the possible polynomials in GF(q) with a degree less than that of f(x). if none of them divide f(x) (remainder = 0), f(x) is said to be irreducible. If it's reducible at least two factors will exist.

Another idea is to substitute all the elements of GF(q) in f(x) and if the values obtained are non-zero for all the elements, then too the polynomial is said to be irreducible. We implemented the first idea in the code given below.

CODE:

The assignment was coded in MATLAB. A function “npermutek” is required to run the code. The function is also attached in the zip folder.

1. Read the input from the user. The user needs to provide a prime number for the field (p in GF(p)) and the polynomial to be tested for irreducibility. The polynomial coefficients are to be entered in square brackets separated by spaces in the ascending order of degree (Example: [1 1 0 1] translates to $1 + x + x^3$). The default input for p is 2 and the default polynomial is [1 1 1 1].

```
clc;
clear all;
close all;
%==========================================================================
%Input dialog
prompt = {'Enter p(in GF(p)):', 'Enter the coefficients of the polynomial inside square brackets separated by spaces (increasing order of degree):'};
dlgtitle = "Input";
definput = {'2', '[1,1,1,1]'};
dims = [1 35];
inp = inputdlg(prompt,dlgtitle,dims,definput);
%==========================================================================
```

2. The input is processed. A vector v containing all possible elements in the Galois field is created (0 to p-1 for GF(p)). The input polynomial is displayed in the traditional format using x as a variable. “npermutek” is used to generate all possible polynomials with degree less than the input polynomial and having coefficients from the GF(p) and is stored in “a_”.

```
%Input processing
p = str2double(inp{1}); %GF(p)
v = 0:p-1; %Possible elements in GF(p)
% Example: b = [1 0 1 0 0 0 0 1] Input polynomial 1 + x^2 + x^7 + x^3
b = str2num(inp{2});
disp('Input polynomial:')
gfpretty(b)
m_ = size(b);
m = m_(2); %Size of the input polynomial array
a_ = npermutek(v,m-1); %All possible polynomials in GF(p) with a degree less than the input polynomial
n_ = size(a_);
n = n_(1);
%==========================================================================
```

3. A boolean ("is_irreducible") is created to keep track of the reducibility of the polynomial. The input polynomial is divided with all polynomials in "a_". The loop is exited when a factor is found (the remainder is 0) and the factor is printed.

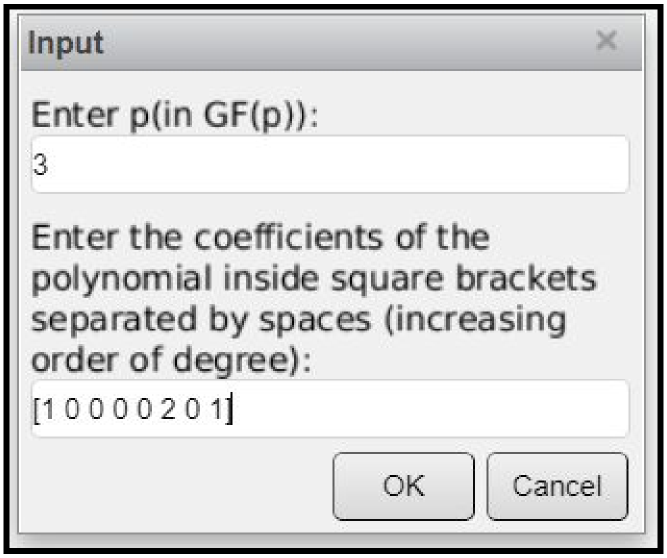
```
%Check for irreducibility
is_irreducible = true;
r = -1;
for i = 1:n
    is_zero = false;
    for j = 0:p-1
        if isequal(a_(i,:), [j,zeros(1,m-2)]) %Check if all zeros or constant. If so, skip checking with that polynomial.
            is_zero = true;
        end
    end

    if is_zero == true
        continue
    end

    [q,r] = gfdeconv(b,a_(i,:),p); %Divide all polynomials and check the remainder

    if r == 0 %If remainder is 0, the
        is_irreducible = false;
        disp("Factor:")
        gfpretty(a_(i,:))
        break
    end
end
%==========================================================================
```

4. Print the corresponding message using the Boolean.

```
%Print the result
if is_irreducible == true
    disp("The input polynomial is irreducible over GF(p)");
else
    disp("The input polynomial is reducible over GF(p)");
end
%==========================================================================
```

OUTPUTS:

Case 1:




```
Input polynomial:

                                          5    7
                                    1 + 2 X  + X
The input polynomial is irreducible over GF(p)
```

Case 2:

Input

Enter p(in GF(p)):

3

Enter the coefficients of the polynomial inside square brackets separated by spaces (increasing order of degree):

[1 0 0 0 0 1 0 1]

OK Cancel

```
Input polynomial:

                                  5    7
                             1 + X  + X
Factor:

                                  2    3
                             1 + X  + X
The input polynomial is reducible over GF(p)
```