Project 2

Harald Nordgren, Gustaf Malm

November 24, 2014

1 Polynomials, fields and cycle sets

Exercise 1

Paper exercise 1

1 We started with

$$p(x) = x^4 + x^2 + 1$$

and to divided by every polynomial up to degree

$$\sqrt{deg(p)} = \sqrt{4} = 2$$

We found that no polynomial divides our $p(x)$, so it is *irreducible.* However, since already $\alpha^6 = 1$, the cycle has fewer elements than $|F_{2^4}| - 1 = 15$, and the polynomial is *not primitive.*

2 Over F_3 our polynomial

$$p(x) = x^3 + x + 1$$

is divisible by $x + 2$. Thus it is *reducible.*

3 By using

$$\alpha^4 = \alpha + 1 \tag{1}$$

we calculate $\alpha^5, \alpha^6, ...$ by successive multiplication with α and substitution by (1) when applicable. We found that $n = 15$ is the smallest exponent that fulfills $\alpha^n = 1$, and by the reasoning from **Assignment 1** this shows that $p(x)$ is *primitive* and *irreducible.*

Computer exercise 1

We test if the function is primitive by using `Primitive` in Maple. If this is true then the polynomial is also irreduible, otherwise we test for reducibility with `Berlekamp`. Berlekamp returns the factorization if one exists, otherwise it just gives the polynomial.

1

```
Primitive (x^23+x^5+1) mod 2;
```

Returns `true`, polynomial is *primitive* and thus *irreducible.*

2

```
Primitive (x^23+x^6+1) mod 2;
```

Returns `false`, so we move on to `Berlekamp`.

```
Berlekamp(x^23+x^6+1,x) mod 2;
```

which gives us the folliwing factorization...

```
  3           4    3
{x  + x + 1, x  + x  + 1,

     16    15    13    12    8    6    4    3    2
    x   + x   + x   + x   + x  + x  + x  + x  + x  + x + 1}
```

so the polynomial is *reducible.*

3

```
Primitive (x^18+x^3+1) mod 2;
```

returns `false`, but

```
Berlekamp(x^18+x^3+1,x) mod 2;
```

tells us the polynomial is *irreducible.*

4

```
Berlekamp(x^8+x^6+1,x) mod 7;
```

gives us

```
  4      3      2             4      3      2
{x  + 3 x  + 5 x  + 5 x + 6, x  + 4 x  + 5 x  + 2 x + 6}
```

so the polynomial is *reducible.*

Exercise 2

Paper exercise 2

1 By calculating the powers of α using

$$\pi(\alpha) = \alpha^4 + \alpha + 1$$

we reach $\alpha^{15} = 1$, and no smaller n fulfills $\alpha^n = 1$. The order is 15 and π is a primitive.

2 Since $gcd(15, 2) = 1$, the order of α^2 is also 15.

3 Since $(\alpha^3)^5 = \alpha^{15} = 1$, this subgroup contains

$$\{1, \alpha^3, \alpha^6, \alpha^9, \alpha^{12}\}$$

and order of α^3 is 5.

4 From the subgroup generated by α^5:

$$\{1, \alpha^5, \alpha^{10}\}$$

it is clear that $ord(\alpha^5) = 3$.

Computer exercise 2

1 We used Galois field(GF) in maple to calculate the orders. First we define the field as

```
G := GF(2,18,a^{18}+a^3+1);
```

then we define a varible `x` to be `a^n`. Then we used

```
G[Order]
```

to calculate the order of `a` like

```
x := G[ConvertIn](a);
G[order](x) = 189
```

2

```
x := G[ConvertIn](a^2);
G[order](x) = 189
```

3

```
x := G[ConvertIn](a^3);
G[order](x) = 63
```

4

```
x := G[ConvertIn](a^3+a);
G[order](x) = 262143
```

which is $2^{18} - 1$, i.e. full order, and the polynomial is primitive.

Exercise 3

Paper exercise 3

1

$$p(x) = x^4 + x^2 + 1$$

gives us $\alpha^6 = 1$, and no smaller integer fullfills $\alpha^n = 1$. The cycle set consists of $\{\alpha^n | n \in Z_6\}$.

2 Since $\alpha^{14} = 1$, our cycle contains 14 elements, namely

$$\{\alpha^n | n \in Z_{14}\}$$

Computer exercise 3

1 We create a field for $x^{23} + x^5 + 1$:

```
G:=GF(2,23,x^23+x^5+1);
```

and test its order

```
a := G[ConvertIn](x);
G[order](a);
```

which is $8388607 = 2^{23} - 1$, i.e.e full order, so the cycle contains every elements except 0.

2 We tried creating a field for

$$x^{23} + x^6 + 1$$

However since it is not an irreducible polynomial the function we used for the first polynomial does not support this. We were unable to find any maple function that supports reducible polynomials and therefore we have to conclude that we can not find any cycle sets for this function.

It might be possible to write a program that tests it way to find the field in the same way as we did in the paper exercise however this would require considerable effort and would not appear be in the scope of this task since it is supposed to be completed in Maple.

Exercise 4

Paper exercise 4

We took random polynomial of degree 4 and tested whether it was primitive or not by writing out it's multiplicative cycle. We found the primitive polynomial:

$$p(x) = x^4 + x + 1$$

Computer exercise 4

We found a little program written in maple that tests it way to primitive polynomials of a certain degree. It can be seen below.

```
F:= NULL: n:= 0:
  while n < 10 do
      f:= 1+expand(x*randpoly(x,degree=3,coeffs=rand(0..4)))+x^4;
      if Primitive(f) mod 5 and not member(f,[F]) then
          F:= F,f; n:= n+1 fi
  end do:
  F;
```

The program provides us with 10 different primitive polynomials and we chose to use:

$$3x^4 + 3x^2 + 4x + 1$$

2 De Bruijn sequence

Exercise 5

Paper exercise 5

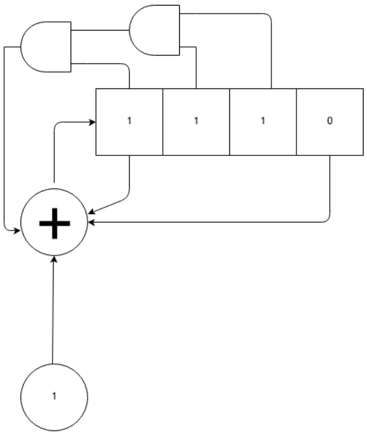

Figure 1: Shift register implementation of $x^4 + x + 1$, including the zero element

Computer exercise 5

```
import java.io.IOException;
import java.io.PrintWriter;

public class Main {
    final static int MOD_BASE = 5;

    static boolean[] WheresThatZeroElement = new boolean[4445];
    public static void main(String[] args) throws IOException {
        final int[] COEFF = {4,3,0,3};
        final int K = 1;
        final int[] START_VALUES = {0,0,0,0};
        final int[] zeroNumbers = {1,1,1,1};
        final int[] preZeroNumbers = {1,1,1,2};
        final int postZeroNumber = 4;

        final int MOD_BASE2 = 2;
        final int[] COEFF2 = {1,0,0,1};
        final int K2 = 1;
        final int[] START_VALUES2 = {0,0,0,0};
        final int[] zeroNumbers2 = {1,1,1,1};
        final int[] preZeroNumbers2 = {1,1,1,0};
        final int postZeroNumber2 = 0;
        PrintWriter writer = new PrintWriter("seq");
        DeBruijnSeq seq2 = new DeBruijnSeq(MOD_BASE2,COEFF2,
            K2,START_VALUES2, zeroNumbers2, preZeroNumbers2, postZeroNumber2);

        DeBruijnSeq seq = new DeBruijnSeq(MOD_BASE,COEFF,
            K,START_VALUES, zeroNumbers, preZeroNumbers, postZeroNumber);

        writer.print("000");
        int count = 0;
        for(int i=0; i < Math.pow(MOD_BASE2,4); i++){
            for(int k = 0; k < Math.pow(MOD_BASE,4); k++){
                int[] seq1Value = seq.getNextValue();
                int[] seq2Value = seq2.getNextValue();

                System.out.print(count + ":\t");
                for(int m = 0; m < 4; m++){
                    System.out.print(seq1Value[m] + MOD_BASE * seq2Value[m]);
                }

                writer.print(seq1Value[0]+seq2Value[0]*MOD_BASE);
                System.out.println();
                count++;
            }
        }
        writer.close();
    }
}

import java.util.LinkedList;


public class DeBruijnSeq {
    private  int MOD_BASE;
    private  int[] COEFF;
    private  int K;
    LinkedList<Integer> list;
    private int[] zeroNumbers;
    private int[] preZeroNumbers;
    private int postZeroNumber;

    public DeBruijnSeq(int MOD_BASE,int[] COEFF,
        int K, int[] START_VALUES, int[] zeroNumbers,
        int[] preZeroNumbers, int postZeroNumber){
        this.COEFF = COEFF;
        this.MOD_BASE = MOD_BASE;
        this.K = K;
        this.zeroNumbers = zeroNumbers;
        this.preZeroNumbers = preZeroNumbers;
        this.postZeroNumber = postZeroNumber;
        list = new LinkedList<Integer>();
        for (int i = 0; i < COEFF.length; i++) {
            list.add(START_VALUES[i]);
        }
    }

    public int[] getNextValue(){
        boolean zeroNbr = true;
        boolean preZeroNbr = true;
        int sum = K;
        for (int i = 0; i < COEFF.length; i++) {
            sum += COEFF[i]*list.get(i);
            if(list.get(i) != preZeroNumbers[i]){
                preZeroNbr = false;
            }
            if(list.get(i) != zeroNumbers[i]){
                zeroNbr = false;
            }
        }
        if(zeroNbr){
            list.addFirst(postZeroNumber);
        }else if(preZeroNbr){
            list.addFirst(zeroNumbers[0]);
        }else{
            list.addFirst(sum % MOD_BASE);
        }
        list.removeLast();
        int[] nbrs =  new int[4];
        for (int k = 0; k < 4; k++) {
            nbrs[k] = list.get(k);
        }
        return nbrs;

    }
}
```