

# Groups and Rings - SF2729

## Homework 2 (Rings)

Jim Holmström - 890503-7571

April 16, 2012

**Exercise 1.** Let  $\sigma_m : \mathbb{Z} \rightarrow \mathbb{Z}_m$  be the natural homomorphism given by  $\sigma_m(a) = a \pmod{m}$ .

a. Show that  $\overline{\sigma}_m : \mathbb{Z}[x] \rightarrow \mathbb{Z}_m[x]$  given by

$$\overline{\sigma}_m(a_0 + a_1x + \dots + a_nx^n) = \sigma_m(a_0) + \sigma_m(a_1)x + \dots + \sigma_m(a_n)x^n \quad (1)$$

is an homomorphism of  $\mathbb{Z}[x]$  onto  $\mathbb{Z}_m[x]$ .

b. Show that  $\text{degree}(f(x) \in \mathbb{Z}[x]) = \text{degree}(\overline{\sigma}_m(f(x))) = n \wedge \overline{\sigma}_m(f(x))$  has no nontrivial factors in  $\mathbb{Z}_m[x] \Rightarrow f(x)$  is irreducible in  $\mathbb{Q}[x]$ .

c. Show that  $x^3 + 17x + 36$  is irreducible in  $\mathbb{Q}[x]$

*Solution.* a.  $\overline{\sigma}_m(f(x)+g(x)) = \overline{\sigma}_m \sum (f_i + g_i)x^i = \sum \overline{\sigma}_m(f_i + g_i)x^i = \sum (\overline{\sigma}_m(f_i) + \overline{\sigma}_m(g_i))x^i = \overline{\sigma}_m(f(x)) + \overline{\sigma}_m(g(x))$  and  $\overline{\sigma}_m(f(x)g(x)) = \overline{\sigma}_m(\sum (\sum f_i g_{n-i}) x^n) = \sum \overline{\sigma}_m(\sum f_i g_{n-i}) x^n = \sum (\sum \overline{\sigma}_m(f_i g_{n-i})) x^n = \sum (\sum \overline{\sigma}_m(f_i) \overline{\sigma}_m(g_{n-i})) x^n = \overline{\sigma}_m(f(x)) \overline{\sigma}_m(g(x))$  Which shows that  $\overline{\sigma}_m$  is an homomorphism.

$a(x) \in \mathbb{Z}_m[x]$  and  $b(x) \in \mathbb{Z}[x]$  having the same coeffs but seen as in  $\mathbb{Z}$  instead of  $\mathbb{Z}_m$  with this we see that  $\overline{\sigma}_m(a(x)) = b(x)$ , so it is onto.  $\square$

b.  $f = gh$  for  $g, h \in \mathbb{Z}[x]$  where  $\text{degree}(f) > \text{degree}(g) \wedge \text{degree}(f) > \text{degree}(h)$

Applying  $\overline{\sigma}_m$  on  $f$ :  $\overline{\sigma}_m(f) = \overline{\sigma}_m(g)\overline{\sigma}_m(h)$  is a factorization of  $\overline{\sigma}_m$  into polynoms with a degree less then  $n$  of  $\overline{\sigma}_m(f)$  which is a contradiction

$\Rightarrow f(x)$  is irreducible in  $\mathbb{Z}[x]$

$\Rightarrow$  (by Theorem 23.11)  $f(x)$  is irreducible in  $\mathbb{Q}[x]$   $\square$

c. Magically choosing  $m = 5$

$$\overline{\sigma}_5(x^3 + 17x + 36) = x^3 + 2x + 1$$

By hand it's simple to show that:

$$(x^3 + 2x + 1)(\{-2, -1, 0, 1, 2\}) \neq 0 \quad (2)$$

and by Theorem 23.10 irreducible over  $\mathbb{Z}_5$  and by the findings in (b) we also have that  $x^3 + 17x + 36$  is irreducible over  $\mathbb{Q}$   $\square$

**Exercise 2.** Let  $f(X) = X^4 - X^2 + 1$ . Prove that  $f(X)$  is irreducible in  $\mathbb{Z}[X]$  and show that  $f(X)$  is reducible in  $\mathbb{Z}_m[X]$  for  $m = \{2, 3, 5\}$  by determining the factorization into a product of irreducible polynomials.

*Solution.* Starting with the smaller rings:

$m=2$ :

$$(x^2 + x + 1)(x + 1)^2 = x^4 - x^2 + 1 \quad m=3:$$

$$(x^2 + 1)^2 = x^4 - x^2 + 1 \quad m=5:$$

$(x^2 + 3x + 1)(x^2 + 2x + 4) = x^4 - x^2 + 1$  Which are all found by a computer program (see last in document) and hand verified to so that the calculations is correct and the terms indeed irreducible.

Now prove that  $f(X)$  is irreducible in  $\mathbb{Z}[X]$ . Firstly noting that I don't have infinite RAM nor infinite time so are abandoning the computer program for this part.

For  $f$  to have a zero in  $\mathbb{Z}$  it must divide 1, so the only 2 possibilities are the units and we have that  $f(1) = f(-1) = 1 \neq 0$ . Which results in no factors of degree 1.

Now look for factors of degree 2:

Assume factors and exists and expand the polynoms with general coeffs in  $\mathbb{Z}$ :

$$(a_2x^2 + a_1x + a_0)(b_2x^2 + b_1x + b_0) = x^4 - x^2 + 1 \quad (3)$$

Calculate the left side (by Cauchy-product) and pattern-match the coeffs:

$$a_0b_0 = 1 \quad (4)$$

$$a_0b_1 + a_1b_0 = 0 \quad (5)$$

$$a_0b_2 + a_1b_1 + a_2b_0 = -1 \quad (6)$$

$$a_2b_1 + a_1b_2 = 0 \quad (7)$$

$$a_2b_2 = 1 \quad (8)$$

One can see that this system is not solvable in  $\mathbb{Z}$  since from the first 2 equations we have  $(a_0 = b_0 = 1 \vee a_0 = b_0 = -1) \wedge (a_2 = b_2 = 1 \vee a_2 = b_2 = -1)$

and testing all these possible combinations of  $a_0, b_0, a_2, b_2$  in the three last equations will all be insolvable in  $\mathbb{Z}$  and thus leaving us with that  $f$  can't be factored in  $\mathbb{Z}$  by definition irreducible in  $\mathbb{Z}[X]$   $\square$

The computer-program in use:

```
import operator
import copy
import itertools as itt
import string
import math

#=====Ring definition=====
class Zn:
    def __init__(self,n,i):
        """
        Initz Z_n with the element i
        """
        self.n=n
        self.i=i%n #fugly but works with negative numbers which is nice i
                    #(but platform dependent perhaps)

    def __str__(self):
        """
        You are on your own on tracking n, mostly one has the same n
        """
        return str(self.i)

    def __eq__(self,other):
        """
        NOOOOT!! Must be of the same Zn to be the same
        """
        if(isinstance(other,int)):
            return self.i==other
        return self.i==other.i
    def __ne__(self,other):
        return not operator.__eq__(self,other)

    def __add__(self,other):
        assert self.n==other.n
        return Zn(self.n,(self.i+other.i)%self.n)
    def __mul__(self,other):
        assert self.n==other.n #not defined else
        return Zn(self.n,(self.i*other.i)%self.n)
    def __pow__(self,m):
        """
        return g**n
```

```

        """
        return Zn(self.n,(self.i**m)%self.n)

def __hash__(self):
    return self.i

class Polynom:
    def __init__(self,c):
        """
        Starts with the constant c_0
        """
        self.c=c

    def __str__(self):
        output=""
        for i,c_i in enumerate(reversed(self.c)):
            if((len(self)-i-1)>1): #fugly with double reverse TODO fix
                output+=str(c_i)
                output+="X^"+str(len(self)-i-1)+"+"
            elif((len(self)-i-1)==1):
                output+=str(c_i)+"X+"
            else:
                output+=str(c_i)
        return output

    def __eq__(self,other):
        N=len(self)
        M=len(other)
        #if all elements is equal and the part hanging outside is all zero
        #then the polynoms are equal
        pseudoeq=all(map(lambda (a,b):a==b,zip(self.c,other.c)))
        if(N==M):#fugly code #TODO fixit
            return pseudoeq
        elif(M<N):
            return pseudoeq and reduce(operator.add,self.c[M:N])==0
        else: #(N<M)
            return pseudoeq and reduce(operator.add,other.c[N:M])==0

    def __neq__(self,other):
        return not operator.__eq__(self,other)

    def __add__(self,other):
        c_res=map(lambda (i,j):i+j,itt.izip_longest(self.c,other.c,fillvalue=0))
        return Polynom(c_res)

```

```

def __len__(self):
    return len(self.c)

def __mul__(self, other):
    #cauchy
    c_n=[0]*(len(self)+len(other)-1)

    for k in range(len(c_n)):
        filtered=filter(lambda (i,j):i+j==k,itt.product(range(len(self)),
            range(len(other))))
        c_n[k] = reduce(operator.add,map(lambda (i,j):
            self.c[i]*other.c[j],filtered))

    return Polynom(c_n)

degree=3
for m in [2,3,5]:
    print "m=",m,"degree",degree,": "
    Zm=map(lambda i:Zn(m,i),range(m))

    PZm=map(lambda c:Polynom(c),itt.product(Zm,repeat=(degree+1)))

    F=Polynom(map(lambda i:Zn(m,i),[1,0,-1,0,1]))

    factors=filter(lambda (f,g):f*g==F,itt.product(PZm,repeat=2))

    for f,g in factors:
        print str(f)+"*"+str(g)+"="+str(f*g)

#also tested the choosen factors from above from factors (to be irreducible)

```