PROJECT

Paul Vrbik : 301056796 : MATH 819 : Spring 2007

May 28, 2007

The modular GCD algorithm and Hensel lifting algorithm

Question 1

Chinese Remainder Theorem for $\mathbb{Z}_q[t]$

For coprime polynomials $[m_1(t), m_2(t), ..., m_N(t)]$ where $M = \prod m_k$. The set of congruences

$$c(t) \equiv c_k(t) \mod m_k(t)$$

has unique solution satisfying $\deg f < \deg M$ given by

$$c(t) = \sum_{k=1}^{N} \big(c_k(t) \cdot N_k(t) \cdot M_k(t)\big) \mod M(t) \tag{1}$$

where

$$M_k(t) = \frac{M(t)}{m_k(t)}$$

and $N_k(t)$ satisfies

$$N_k(t) \cdot M_k(t) + n_k(t) \cdot m_k(t) = 1.$$

That is, $N_k(t)$ is the inverses of $M_k(t) \mod m_k(t)$.

Proof

We first show that $c(t)$ satisfies $c(t) \equiv c_k(t) \mod m_k(t)$ for any $k \in [1, N]$. Denote $s_i = c_i(t) \cdot N_i(t) \cdot M_i(t)$ and rewrite the summand in (1) as

$$c(t) = \sum_{i=1}^{N} s_i \mod M.$$

Observing that $M_i(t) \equiv 0 \mod m_i \Rightarrow s_i \equiv 0 \mod m_i$ for $i \neq k$ and that $N_i(t)$ always exists since $\mathbb{Z}_q[t]$ is a Euclidean domain, we simplify

$$\begin{aligned} c(t) &\equiv \sum_{i=1}^{N} s_i \mod M. \\ &\equiv s_k \mod M \\ &\equiv c_i(t) \cdot \left(\frac{M(t)}{m_i(t)}\right)^{-1} \cdot \left(\frac{M(t)}{m_i(t)}\right) \mod m_i(t) \\ &\equiv c_i(t) \mod m_i(t). \end{aligned}$$

To show that this solution is unique suppose we have an alternate solution c' such that $c' \not\equiv c \mod M$ satisfying $c'(t) \equiv c_k \mod m_k(t)$ for any k.

$$\begin{aligned} &\Rightarrow c(t) - c'(t) \equiv 0 \mod m_k(t) \\ &\Rightarrow m_i(t) \mid c(t) - c'(t) \end{aligned}$$

Since we have unique factorization and relatively prime m_i's, their product most also divide $c(t) - c'(t)$.

$$\begin{aligned} &\Rightarrow M(t) \mid c(t) - c'(t) \\ &\Rightarrow c(t) - c'(t) \equiv 0 \mod M \\ &\Rightarrow c(t) \equiv c'(t) \mod M \end{aligned}$$

This contradicts the assumption that $c' \not\equiv c \mod M$ implying the uniqueness of $c(t)$. □

Implementation

```
CRT:=(As,fs,x,Q)->h_CRT(As,fs,x,Q,mul(fs[i], i=1..nops(fs))):

h_CRT:=proc(As,fs,x,Q,M) local v1,v2,Abar,‘&pmod‘:

     ‘&pmod‘:=(a,b)-> simplify(Rem(a,b,x) mod Q):

    if (nops(As)=1) then
        return As[1] &pmod fs[1]:
    end if:

    Abar:=CRT(As[1..-2], fs[1..-2],x,Q):

    Gcdex(fs[-1],simplify(M/fs[-1]),x,’f1Inv’) mod Q:
    v1:=fs[-1]*f1Inv:

    Gcdex(simplify(M/fs[-1]),fs[-1],x,’f2Inv’) mod Q:
    v2:=M/fs[-1]*f2Inv:

    v1:=simplify(v1):
    v2:=simplify(v2):

    return (Abar*v1 + As[-1]*v2) &pmod M:

end proc:
```

Testing

```
> restart;
> read "CRT.mpl";
```

$$cs := [t^2, t^2 + t + 1]$$

$$ms := [t^3 + t + 1, t^3 + t^2 + 1]$$

$$q := 2$$

```
> c:=CRT(cs,ms,t,q);
```

$$c := t^5 + t^4 + t^3 + t^2 + t$$

```
> Rem(c,ms[1],t) mod q;
```

$$t^2$$

```
> Rem(c,ms[2],t) mod q;
```

$$t^2 + t + 1$$

Question 2

The Modular GCD Algorithm for $\mathbb{Z}_p[t][x]$

```
fModGcd:=proc(inA,inB,q)
local Phi,a, b, contA, contB, contG, gam, G, M, p, lcA, ap, bp, gp, u, g:

    read "procs.mpl": read "CRT.mpl":

    `mod`:=mods;

    a:=inA:
    b:=inB:

    contA:=Content(a,x,'ppa') mod q;
    contB:=Content(b,x,'ppb') mod q;
    contG:=myGcd(contA,contB,q);

#   Inexplicably ppa and ppb are assigned 1 I was unable to resolve this bug.

    a:=Primpart(a,x) mod q;
    b:=Primpart(b,x) mod q;

    gam:=myGcd(lcoeff(a,x), lcoeff(b,x),q);
    G:=0:
    M:=1:

    p:=Nextprime(t^2,t) mod q;

    lcA:=lcoeff(a,x):

    while (true) do

        while (Rem(lcA,p,t) mod q = 0) do
            print("BAD PRIME",p):
            p:=NextMonicPrime(p,t,q):
        end do:

        Phi:=x->Rem(x,p,t) mod q:

        ap:=Phi(a):
        bp:=Phi(b):
        gp:=fgcd(ap,bp,p,q):

        if degree(gp,x)=0 then
            return contG:
        end if:

#       correct gp
        gp:=Phi(gam*gp):
        gp:=Phi(gp,p,q):

        if G=0 then
            G:=gp:
            M:=p:
```

```
        elif degree(gp,x) > degree(G,x) then
#           do nothing (unlucky prime)
        elif degree(gp,x) < degree(G,x) then
            G:=gp:
            M:=p:
        else
            u:=CRT([G,gp],[M,p],t,q):

            if Expand(u-G) mod q=0 then
                g:=Primpart(contG*u,x) mod q:

                if Rem(a,g,t) mod q=0 and Rem(b,g,t) mod q =0 then
                    return g;
                end if:
            end if:
            G:=u:
            M:=M*p:
        end if:

        p:=NextMonicPrime(p,t,q):

    end do:

end proc:
```

procs.mpl

```
myGcd:=proc(a,b,p) local ca,cb,cg:
    ca:=Content(a,t,'ppa') mod p:
    cb:=Content(b,t,'ppb') mod p:
    cg:=Gcd(ca,cb) mod p:
    return Expand(cg*(Gcd(ppa,ppb) mod p)) mod p:
end proc:

fgcd:=proc(inF1,inF2,p,q) local g, f1, f2,r:
    f1:=inF1 mod q:
    f2:=inF2 mod q:
    r:=RootOf(p):
    g:=subs(r=t, (myGcd(subs(t=r,f1),subs(t=r,f2),q))):
    return Expand(g) mod q:
end proc:

NextMonicPrime:=proc(p,t,q) local np:
    np:=Nextprime(p,t) mod q:
    if lcoeff(np,t)=1 then
        return np:
    else
        return Nextprime(t^(degree(np,t)+1),t) mod q:
    end if
end proc:
```

Testing

```
> restart;
> read "fModGcd.mpl";
```

$$q := 3$$

$$g := (t^3 - t) x^5 - t^{11} x^3 + t^7 x + t^9 + 1$$

$$abar := t x^5 - t^6 x^2 + 1$$

$$bbar := t x^4 + x^2 + t^7$$

$$a := (t x^5 - t^6 x^2 + 1) ((t^3 - t) x^5 - t^{11} x^3 + t^7 x + t^9 + 1)$$

$$b := (t x^4 + x^2 + t^7) ((t^3 - t) x^5 - t^{11} x^3 + t^7 x + t^9 + 1)$$

```
> fModGcd(a,b,q);
```

$$x^5 t^3 - t x^5 - t^{11} x^3 + t^7 x + t^9 + 1$$

```
> -myGcd(a,b,q);
```

$$x^5 t^3 - t x^5 - t^{11} x^3 + t^7 x + t^9 + 1$$

Question 3

Hensel Lifting for $\mathbb{Z}_q[t][x]$

```
UHLA:=proc(inA,q,p,inU0,inW0,B,inGam)
local a, u0, w0, Phi, alpha, gam, rof, ss, tt, u, w, err, modulus,
c, sigmabar, taubar, rr, qq, sigma, tau, t1:

    read "common.mpl";

    `mod`:=mods:
    a:=inA; u0:=inU0; w0:=inW0;

    Phi:=x-> Rem(x,p,t) mod q:

    alpha:=lcoeff(a,x):

    if nargs<7 then
        gam:=alpha:
    else
        gam:=inGam:
    end if:

    a:=gam*a:

    u0 := Phi(gam*Monic(u0,p,q),p,q):
    w0 := Phi(gam*Monic(w0,p,q),p,q):

    rof:=RootOf(p):
    Gcdex(subs(t=rof,u0), subs(t=rof,w0), x, 'ss', 'tt') mod q:
    ss := subs(rof=t, ss):
    tt := subs(rof=t, tt):

    u:=replace_lc(u0,gam):

    w:=replace_lc(w0,alpha):

    err:=Expand(a-u*w) mod q:
    modulus:=p:
    err:=Quo(err,modulus,x,'rr') mod q;

    while (err <> 0 and degree(modulus,t)<B*degree(gam,t)+1) do

        sigmabar:=Phi(ss*err);
        taubar:=Phi(tt*err);

        rr:=subs(rof=t, Rem( subs(t=rof,sigmabar), subs(t=rof,w0), x, 'qq') mod q);
        qq:=subs(rof=t, qq);

        sigma:=rr;
        tau:=Phi(Expand(taubar+qq*u0) mod q);

        err:=Expand(err-(u*sigma+tau*w)-sigma*tau*modulus) mod q;
        err:=Quo(err,p,t,'rr') mod q;
```

```
        u:=Expand(u+tau*modulus) mod q;
        w:=Expand(w+sigma*modulus) mod q;

        modulus:=modulus*p;

    end do;

    if err=0 then
        sigma:=Content(u,x) mod q;
        u:=Quo(u,sigma,x) mod q;
        t1:=Quo(gam,sigma,x) mod q:
        w:=Quo(w,t1,x) mod q;
        return (u,w);
    else
        return "FAIL";
    end if:

end proc:
```

common.mpl

```
locPhi := proc(a, p, q)
    Rem(a, p, t) mod q;
end:

Monic := proc(a, p, q)
local lcA, lcAinv, rof, g;
    lcA := lcoeff(a, x);
    g:=Gcdex(lcA, p, t, 'lcAinv') mod q;
    lcAinv := subs(rof=t, lcAinv);
    return collect(locPhi(a*lcAinv,p,q),x);
end:

replace_lc := proc(f,a) local n,lc:
    n:=degree(f,x):
    lc:=lcoeff(f,x):
    return (f-lc*x^n+a*x^n):
end proc:

fgcd:=proc(inF1,inF2,p,q) local g, f1, f2,r:
    f1:=inF1 mod q:
    f2:=inF2 mod q:
    r:=RootOf(p):
    g:=subs(r=t, (myGcd(subs(t=r,f1),subs(t=r,f2),q))):
    return Expand(g) mod q:
end proc:

myGcd:=proc(a,b,p) local ca,cb,cg:
    ca:=Content(a,t,'ppa') mod p:
    cb:=Content(b,t,'ppb') mod p:
    cg:=Gcd(ca,cb) mod p:
    return Expand(cg*(Gcd(ppa,ppb) mod p)) mod p:
end proc:
```

Testing

```
> read "UHLA.mpl";
                                    q := 3

                                        3
                                p := t  + 2 t + 2

                            Phi2 := x -> Rem(x, p, t) mod q

                                  3       5    11  3    7      9
                        u := (t  - t) x  - t   x  + t  x + t  + 1

                                        5    6  2
                                  w := t x  - t  x  + 1

                         3       5    11  3    7      9          5    6  2
                 a := ((t  - t) x  - t   x  + t  x + t  + 1) (t x  - t  x  + 1)

                           5      3                     3        3         2
                    u0 := x  + 2 x  + x + (2 x + 1 + 2 x ) t + (x  + 2 x) t

                                      2     5    2         2  2
                             w0 := 1 + 2 x  + (x  + x ) t + 2 x  t

> fgcd(u0,w0,p,q);
                                        1

> Phi2(a-u0*w0);
                                        0

> #B=17 since it is the largest degree in t of the coefficients of a;
> (ut,wt):=UHLA(a,q,p,u0,w0,17);
                              3       5    11  3    7      9         5    6  2
                   ut, wt := (t  - t) x  - t   x  + t  x + t  + 1, t x  - t  x  + 1

> Expand(a-ut*wt) mod q;
                                        0
```

Question 4

To be more efficient we determine an update formula for

$$\frac{\varepsilon_k}{p^k}$$

where ε_k is the error on the kth iteration.

$$\left(\frac{\varepsilon_{k+1}}{p^{k+1}}\right) = \frac{\varepsilon_k - (u^{(k)} \cdot w_k + u_k \cdot w^{(k)}) \cdot p^k - u_k \cdot w_k \cdot p^{2k}}{p^{k+1}} \tag{2}$$

$$= \left(\frac{\varepsilon_k}{p^k}\right) \cdot \frac{1}{p} - \left(u^{(k)} \cdot w_k + u_k \cdot w^{(k)}\right) \cdot \frac{1}{p} - \left(u_k \cdot w_k \cdot p^k\right) \cdot \frac{1}{p} \tag{3}$$

$$= \left(\frac{\varepsilon_k}{p^k} - \left(u^{(k)} \cdot w_k + u_k \cdot w^{(k)}\right) - \left(u_k \cdot w_k \cdot p^k\right)\right) \cdot \frac{1}{p} \tag{4}$$

Assuming the classical algorithms for polynomial multiplication and division in $\mathbb{Z}_q[t]$, if $\deg_x(a) = n$ and $\deg_t(a) = m$ then calculating the sum in (4) costs

$$\sum_{k=1}^{m} O(n^2 k) \in O(n^2 m^2)$$

and division by p is $O(n^2 m)$. This implies the total cost of the error calculation is reduced to $O(n^2 m^2)$ as desired. This change is reflected in the implementation given in three.

Question 5

MODGcd Algorithm An *unlucky prime* is one that satisfies (in $\mathbb{Z}_q[t]$)

$$\gcd\left(\phi_p(\bar{a}), \phi_p(\bar{b})\right) \neq 1$$

or

$$\mathrm{Res}_x(\bar{a}, \bar{b}) \equiv 0 \mod p$$

For question two where

$$\mathrm{Res}_x(\bar{a}, \bar{b}) = t^{41} + t^{39} + t^{31} + 2t^{18} + t^5 + t^2$$

these primes are given below :

- $p = t$
- $p = 2 + t + t^{31} + t^{18} + t^2 + 2t^7 + 2t^9 + t^{10} + 2t^{11} + t^{12} + 2t^{13} + t^{15} + t^{16} + 2t^{17} + t^{20} + t^{21} + t^{22} + t^{23} + 2t^{24} + 2t^{25} + 2t^{27} + t^{29} + t^{30} + 2t^{32} + t^{33} + t^{34}$
- $p = t^3 + 2t^2 + 2t + 2$
- $p = t^2 + 1$
- $p = t^2 + 2t + 3$

Are all unlucky since $\mathrm{Res}_x(\bar{a}, \bar{b})$ is divisible by p.

Hensel Lifting An *unlucky prime* is one that satisfies (in $\mathbb{Z}_q[t]$)

$$\mathrm{Res}_x(u, w) \equiv 0 \mod p$$

For $u = (t^3 - t)x^5 - t^{11}x^3 + t^7x + t^9 + 1$ and $w = (tx^5 - t^6x^2 + 1)$ where

$$\begin{aligned}\mathrm{Res}_x(u, w) = {} & t^9 + 2t^7 + t^{11} + t^5 + 2t^{13} + t^{14} + t^{16} + t^{15} + 2t^{62} + t^{52} + t^{53} \\ & + 2t^{61} + 2t^{59} + t^{55} + 2t^{49} + t^{45} + 2t^{46} + t^{20} + t^{31} + t^{27} + t^{22} + 2t^{21} \\ & + t^{35} + t^{23} + t^{51} + 2t^{42} + t^{39} + t^{32} + t^{71} + 2t^{34} + t^{43} + 2t^{40} + t^{50}\end{aligned}$$

these primes are given below:

- $p = t$
- $p = t^{14} + t^{13} = 2t^{11} + t^9 + t^6 + t^5 + 2t^4 + t^2 + t + 2$
- $p = t^{11} + t^{10} + 2t^9 + 2t^7 + 2t^5 + t^4 + t^2 + 2t + 2$
- $p = t^5 + t^4 + 2$
- $p = t^{31} + t^{30} + t^{29} + 2t^{27} + t^{26} + 2t^{25} + t^{23} + 2t^{21} + 2t^{20} + 2t^{18} + 2t^{16} + t^{15} + 2t^{14} + 2t^{12} + 2t^{10} + t^8 + 2t^6 + 2t^4 + t^3 + t^2 + 1$
- $p = t^2 + 2t + 2$
- $p = t^3 + 2t + 1$

Are all unlucky since $\mathrm{Res}_x(u, w)$ is divisible by p.

For $u = (t^3 - t)x^5 - t^{11}x^3 + t^7x + t^9 + 1$ and $w = (tx^4 - x^2 + t^7)$ where

$$\begin{aligned}\mathrm{Res}_x(u, w) = {} & 2t^{11} + t^2 + 2t^{10} + t^4 + 2t^{33} + 2t^{22} + t^{15} + t^{31} + t^{23} + 2t^{14} \\ & + 2t^{44} + 2t^{35} + t^{25} + t^{32} + t^{21} + t^6 + t^{36} + 2t^{45} + t^{47} + t^5 \\ & + t^{40} + t^{67} + 2t^{49} + 2t^{29} + t^{27} + 2t^{58} + 2t^{37} + 2t^{28} + 2t^{48} \\ & + 2t^{39} + 2t^{30} + t^{46} + 2t^{57} + t^{55} + 2t^{53} + t^{42} + t^{19}\end{aligned}$$

these primes are given below:

- $p = t$
- $p = t^4 + t^3 + t^2 + 2t + 2$
- $p = t^8 + 2t^5 + 2t^3 + t^2 + 2$
- $p = t^3 + 2t + 1$
- $p = 1 + t + 2t^3 + t^7 + 2t^{11} + t^2 + 2t^8 + 2t^{13} + 2t^4 + 2t^{24} + t^{15} + 2t^{31} + 2t^{23} + 2t^{14} + 2t^{13} + 2t^{17} + t^{32} + 2t^{16} + t^{21} + 2t^6 + 2t^{18} + 2t^5 + 2t^{29} + 2t^{27} + 2t^{28}$
- $p = t^{18} + t^{16} + 2t^{13} + t^{12} + t^{11} + t^{10} + t^9 + t^8 + 2t^7 + t^4 + 2t^3 + t + 1$

Are all unlucky since $\mathrm{Res}_x(u, w)$ is divisible by p.