Extended Euclidean Algorithm

Lecture eea

The EEA solves A¢ P + B¢ Q = G

Given integers P and Q.

Determine A, B, G such that G=gcd(A,B)

A¢ P + B¢ Q = G

Where uniqueness is asserted by deciding on |A|<|Q|, |B|<|P|, and G is the (positive) GCD..

1*39-1*26= 13 because gcd(39,26)=13.

13*25-9*36=1 because gcd(25,36)=1.

The EEA finds inverses mod Q from A¢ P + B¢ Q = G

Assume Q is a prime integer, and P ≠ Q.
Determine A, B such that A¢ P + B¢ Q = 1.
(gcd(P,Q)=1)
Where uniqueness requires 0<A<Q-1, or |A|<(Q-1)/2
If we do all our arithmetic modulo Q, then Q ≡ 0
And so A*P=1 mod Q.
Thus A is the inverse of P mod Q.
Example:
(-5)*5 +2*13=1, so -5 is the inverse of 5 mod 13.
-5 ≡8 mod 13…

The EEA solves A¢ P + B¢ Q = G, polynomials

Given polynomials (coefficient field??) P and Q.

Determine A, B, G such that G=gcd(A,B)

A¢ P + B¢ Q = G

Where uniqueness is asserted by deciding on a main variable x, with respect to which $\deg(A)<\deg(Q)$, $\deg(B)<\deg(P)$, and G is in some normal form. For example, over rationals, we would insist that G be unit normal. E.g. if it were an integer, G would be 1.

The EEA solves A¢ P + B¢ Q = G

For Knuth's polynomials we would like …

$P=x^8+x^6-3*x^4-3*x^3+8*x^2+2*x-5$,

$Q=3*x^6+5*x^4-4*x^2-9*x+21$,

→

$A=(13989*x^5+18450*x^4+40562*x^3+67125*x^2+5149*x-9737)/130354$

$B=-(4663*x^7+6150*x^6+10412*x^5+18275*x^4-9888*x^3-21579*x^2-3820*x-3889)/130354$

$G=1$

The EEA algorithm (in Macsyma)

```
extended_gcd(u,v,x):=
 block([u1,u2,u3,v1,v2,v3,t1,t2,t3],
       u: rat(u,x), v: rat(v,x),
      [u1,u2,u3]:[rat(1),rat(0),u],
       [v1,v2,v3]:[rat(0),rat(1),v],
     while v3#0 do
         (q: quotient(u3,v3,x),
          [t1,t2,t3]:[u1,u2,u3]-q*[v1,v2,v3],
          [u1,u2,u3]:[v1,v2,v3],[v1,v2,v3]:[t1,t2,t3]),
   [u1,u2,u3])
```

The EEA algorithm (in Macsyma)

Actually, we lied, and the GCD instead of being 1 comes out as -1288744821/543589225.

We have to make a correction here..

The EEA algorithm, reducing the result

```
eea(u,v,x):= /* smallest gcd */
 block([u1,u2,u3,v1,v2,v3,t1,t2,t3, realgcd:gcd(u,v),correction:1],
       u: rat(u,x), v: rat(v,x), [u1,u2,u3]:[rat(1),rat(0),u],
       [v1,v2,v3]:[rat(0),rat(1),v],
      while v3#0 do
         ( print([v1,v2,v3]),
         q: quotient(u3,v3,x),  /* here is where we might like to patch*/
          [t1,t2,t3]: [u1,u2,u3]-q*[v1,v2,v3],
          [u1,u2,u3]:[v1,v2,v3],[v1,v2,v3]:[t1,t2,t3]),
     correction:realgcd/u3, /* the patch we used */
   [u1*correction,u2*correction,realgcd]),
```

The EEA algorithm, reducing the result

Note that in particular, the terms A, B are not directly derived from G=gcd(P,Q), but part of the process.