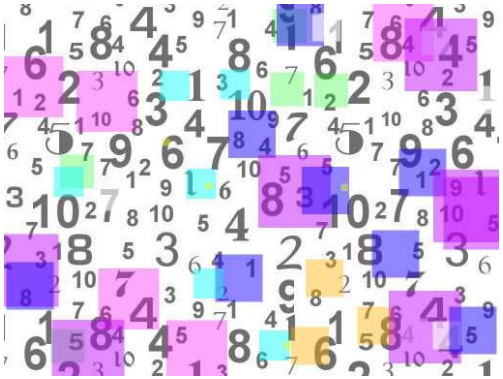
Modular Arithmetic

(Number Theory)



Andrew Juren
CS490
March 29, 2006

Moving on…

- What is modular arithmetic?
  - "**Modular arithmetic"** (sometimes called modulo arithmetic) is a system of *arithmetic* for *integers*, where numbers "wrap around" after they reach a certain value – the modulus. Modular arithmetic was introduced by Carl Friedrich Gauss in 1801.
  - Sometimes called "Clock Arithmetic", four hours after 9 o'clock is 1 o'clock

More formally…

Consider: $A \equiv B \pmod{n}$
["A is congruent to B, modulo n"]

True if A and B have the same remainder when they are each divided by n.

A-B is divisible by n (with no remainder) $\leftrightarrow A - B = kn$

A and B are in the same **equivalence class**

Equivalence Classes

For example, let n=5. Then there are 5 **equivalence classes**, which we will denote as [0], [1], [2], [3] and [4] .

The set [0] contains all numbers that are divisible by 5 – using our notation from yesterday: 5|Z

The set [1], for example, contains these numbers:
{1, -4, 51, 76, 15728934726, -5539485734, …}

What can we do with "Equivalence Classes"?

Add them! [a] + [b] = [a+b]

Let A ∈ [a], B ∈ [b], from the previous slide, we know that

A-a = jn and B-b = kn, so

(A-a) + (B-b) = jn+kn

(A+B) - (a+b) = (j+k)n

A+B ≡ a+b

Since A and B are arbitrary numbers from [a] and [b], the addition of equivalence classes is well defined:

[a] + [b] = [a+b], and subtraction is similarly well defined:

[a] – [b] = [a-b]

What can we do with "Equivalence Classes"?

Multiply them! [a][b] = [ab]

Let A ∈ [a], B ∈ [b], then $A-a = jn$ and $B-b = kn$, for some j, k

$$(A-a) \times (B-b) = jkn^2$$

$$AB + ab - Ab - aB = jkn^2$$

$$AB + (ab - ab) + ab + Ab - aB = jkn^2$$

$$AB - ab - aB + ab - Ab + ab = jkn^2$$

$$AB - ab - a(B-b) - b(A-a) = jkn^2$$

$$AB - ab = jkn^2 + a(kn) + b(jn)$$

$$AB - ab = (jkn + ak + bj)n$$

$$AB \equiv ab \pmod{n}$$

What can we do with "Equivalence Classes"?

Divide them!

Some important **caveat**s:

- With regular integers, we have limits on division – one integer divided by another does not always give an integer result. Additionally, we can not divide by zero.
- Similarily, in modular arithmetic, dividing by [0] is forbidden, but many of the other restrictins disappear, as long as we have a "good" modulus n.

In real numbers, $x \div y = x \times 1 / y$

Similarily, we do the same in modular arithmetic and must find the *modular inverse*

Modular Inverse

The *modular inverse* of an equivalence class b is the equivalence class b^{-1} that obeys $b \times b^{-1} \equiv 1$

$b\ b^{-1} - 1 = kn$

$b\ b^{-1} - nk = 1$

From the Euclidean algorithm, we know that there exists a solution if $gcd(b, n) = 1$ (and the extended algorithm will give us an infinite number of solutions, all of which differ by a multiple of n)

This only happens if b is relatively prime to n (i.e. they do not share any prime factors)

Note: when n is prime, then every integer except those in [0] is relatively prime and has a corresponding inverse class.

The % Operator

The equivalence class [a] contains all numbers that have the same remainder as a when divided by n. Note that a does not have any bounds. Typically, an a in [0, n-1] is chosen to represent that equivalence class.

In Java (and C/C++) the % operator does this, except that *it does not work as expected for negative numbers!*

Given $b \geq 0$, $b \% n$ will give you the remainder in [0, n-1]

But when $b < 0$, $b \% n$ can be negative. Example.

Modular Linear Equations

Now that we can divide, we can solve equations of the form: $Ax \equiv B \pmod n$. If A has an inverse modulo n of A^{-1}, then we can simply multiply both sides by the inverse and get our answer: $x \equiv BA^{-1} \pmod n$

But what if there is no A^{-1}? We must do this "the hard way":

$Ax - nk = B$

Now, let's change n (our modulo) so that g = $gcd(A, n)$. If g|B,

$(A/g)x - (n/g)k = (B/g)$

$(A/g)x \equiv (B/g) \pmod{n/g}$

Since A/g and n/g are now relatively prime, we know that we can divide them using the method above.

Therefore, it is always possible to solve a linear equation if g|B

Powers modulo n

Is it possible to determine the equivalence class of $z^a \pmod n$?

Do you remember Horner's rule for polynomial evaluation?

When we have a polynomial of the form $a_0+a_1 x+a_2 x^2+...+a_n x^n$, where the a_i' s are known, we can evaluate it for any given x by proceeding right-to-left. The trick is to rewrite the polynomial in the form $a_0+x(a_1+x(a_2+x(...+a_n(x^n))))$, start with $r=0$ and proceed by executing the operations "times x," "plus a_n ", "times x," "plus a_{n-1} ", ..., "times x", "plus a_0 ". At the end, r is equal to the value of the polynomial at x.

Successive Squaring

This way of evaluating has some advantages over the obvious method of taking powers of x, multiplying by the corresponding coefficient and adding. We are not interested in those right now. What we will do instead is replace all the additions with multiplications and all the multiplications with exponentiations. Also, we will set x=2 and all the a_i coefficients will be either 1 or z. This gives us the following problem: compute the value of $a_0 2^0 \times a_1 2^1 \times a_2 2^2 \times ... \times a_n 2^n$.

Horner's rule still works, only now it is called **successive squaring** because instead of starting with r=0, we will start with r=1, and instead of repeating "times x, plus a_i ," we will do "raise to the power of x, times a_i ". Raising to the power of x is easy because x=2, so that is a simple squaring operation – hence, the name "successive squaring".

See implementation.