Scribe Notes for *Algorithmic Number Theory*
Class 11—June 2, 1998

Scribes: Scott A. Guyer, Duxing Cai, and Degong Song

Abstract

Today's class covers methods for evaluating polynomials in a ring as well as solving systems of linear congruences. The Chinese Remainder theorem is covered in two forms and an algorithm for solving systems of linear congruences is presented.

1 Evaluating a Polynomial in a Ring

The task is to compute the value of

$$f(x) = \sum_{j=0}^{d} c_j x^j,$$

where $c_d \neq 0$, for some $x \in \mathbb{Z}/(n)$.

Let $a \in \mathbb{Z}/(n)$. There are two methods for calculating $f(a)$. The first is the straight-forward approach suggested by the summation above while the second makes an optimization based on Horner's rule.

Intuitive Polynomial Evaluation. To compute $f(a)$, we need to execute the following steps.

1. Generate a table of all the squares of a, i.e., $a, a^2, \ldots, a^{2^k}$ where $k = \lfloor \log_2 d \rfloor$.
2. For each j with $c_j \neq 0$, compute a^j using the entries in the previously computed table. This will take at most $\lg d$ multiplications.
3. Sum up all the $c_j x^j$.

To analyze the complexity, we introduce the value $r = |\{c_j \mid c_j \neq 0\}|$ to represent the sparsity of the polynomial. In other words, r represents the number of nonzero terms in the polynomial (which is frequently much less than d). Recall that s is the time it takes for multiplication in the ring. Then the bit complexity of polynomial evaluation in a ring is

$$O(rs \log d).$$

Evaluation Using Horner's Rule. Horner's rule makes use of the following observation:

$$\sum_{i=0}^{d} a_i x^i \quad = \quad a_0 + x(a_1 + x(a_2 + x(\cdots x(a_{d-1} + a_d x)\cdots))).$$

Using Horner's rule gives $O(d)$ multiplications. Hence, polynomial evaluation in this case has bit complexity

$$O(sd).$$

Comparing the Two Methods. The first method is preferable when the polynomial has high degree and is sparse. The second method is useful when the polynomial has low degree. However, if the polynomial has high degree and is dense, pick your poison. ;-)

2 Systems of Linear Congruences

In this section, we investigate the task of finding a common solution for a set of linear congruences. For example, the following system represents the most general case.

$$\begin{aligned} a_1x + b_1 &\equiv 0 \pmod{m_1} \\ a_2x + b_2 &\equiv 0 \pmod{m_2} \\ &\vdots \\ a_kx + b_k &\equiv 0 \pmod{m_k}, \end{aligned}$$

where $a_i, b_i, m_i \in \mathbb{Z}$ and $m_i > 0$. There are three questions for which we would like to find answers:

1. When does a system have a solution(s)?
2. How many solutions are there, and can we characterize all of them?
3. Is there an efficient algorithm to find a solution or characterize all solutions?

Before we investigate these questions, please note the following observations.

Observation 2.1 Consider $a_ix_i + b_i \equiv 0 \pmod{m_i}$. If we set $d_i = \gcd(a_i, m_i)$, then it is easy to see from the relation that $d_i \mid b_i$. If, however, $\gcd(a_i, m_i) \not| b_i$, then there is no solution. Otherwise, we can reduce the linear congruence to

$$\frac{a_i}{d_i}x_i + \frac{b_i}{d_i} \equiv 0 \pmod{\frac{m_i}{d_i}}.$$

Henceforth, we assume that all congruences have been reduced in this way such that $\gcd(a_i, m_i) = 1$.

Observation 2.2 Given the linear congruence $a_ix_i + b_i \equiv 0 \pmod{m_i}$ where $\gcd(a_i, m_i) = 1$, we know a_i has a multiplicative inverse $\overline{a}_i^{-1} \in \mathbb{Z}/(m_i)$ because a_i and m_i are relatively prime. Hence, the congruence $a_ix_i + b_i \equiv 0 \pmod{m_i}$ can be rewritten as

$$\begin{aligned} x_i + a_i^{-1}b_i &\equiv 0 \pmod{m_i}, \text{ or} \\ x_i &\equiv -a_i^{-1}b_i \pmod{m_i}. \end{aligned}$$

Henceforth, we will restrict our attention to systems of linear congruences where each congruence has the form $x \equiv x_i \pmod{m_i}$.

Example 2.1 Consider the following system of linear congruences.

$$\begin{aligned} 4x + 2 &\equiv 0 \pmod 6 \\ 3x + 4 &\equiv 0 \pmod 5 \end{aligned}$$

Based on the first observation, we can reduce the first equation because $\gcd(4,6) = 2$. Applying the second observation to the resulting equations yields the following equivalent system of linear congruences.

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 2 \pmod 5 \end{aligned}$$

Now we can proceed to answering the three questions stated above.

2.1 Chinese Remainder Theorem

The Chinese Remainder theorem answers all three of the questions posed with respect to systems of linear congruences.

THEOREM 1 (Chinese Remainder Theorem, version 1, Theorem 5.5.2 in [1]) *Let $m_1, m_2, \ldots, m_k$ be positive integers that are pairwise relatively prime. Let $m = m_1 m_2 \cdots m_k$. Then, given integers $x_1, x_2, \ldots, x_k$, there is an integer x such that*

$$x \equiv x_i \pmod{m_i}.$$

Moreover, x is unique modulo m.

Proof: Define $f_i = m/m_i$ for $1 \leq i \leq k$, then $f_i \equiv 0 \pmod{m_j}$ when $i \neq j$. Note that $\gcd(f_i, m_i) = 1$, so there is a multiplicative inverse $\overline{f}_i^{-1}$ for f_i in $(\mathbb{Z}/(m_i))^*$. Define $e_i = f_i f_i^{-1}$. Then e_i is congruent to 1 modulo m_i, yet congruent to 0 modulo m_j for all $j \neq i$. The required common solution is then,

$$x = \sum_{i=1}^{k} e_i x_i.$$

This is easy to see from the fact that $x \equiv x_i \pmod{m_i}$ for every i (by definition). This proves the existence of a common solution.

To prove the uniqueness of solutions, assume there is another solution $y \equiv x_i \pmod{m_i}$ for all i. Then $x - y \equiv 0 \pmod{m_i}$. Because the m_i are pairwise relatively prime, we also know that $x - y \equiv 0 \pmod{m}$ and hence, $x \equiv y \pmod{m}$.

Example 2.2 Consider the following system of linear congruences.

$$\begin{aligned} x &\equiv 7 \pmod{15} \\ x &\equiv 2 \pmod 4 \\ x &\equiv 6 \pmod 7 \end{aligned}$$

For this problem, we have $x_1 = 7$, $x_2 = 2$, $x_3 = 6$, $m_1 = 15$, $m_2 = 4$, and $m_3 = 7$. Note that the m_i are pairwise relatively prime and $m = m_1 m_2 m_3 = 420$. First, we compute the f_i and their inverses for $1 \leq i \leq 3$:

$$\begin{aligned} f_1 &= 4 \cdot 7 & f_2 &= 15 \cdot 7 & f_3 &= 15 \cdot 4 \\ &= 28 & &= 105 & &= 60 \\ \bar{f}_1^{-1} &= \bar{7} & \bar{f}_2^{-1} &= \bar{1} & \bar{f}_3^{-1} &= \bar{2}. \end{aligned}$$

Next, we compute the e_i for $1 \leq i \leq 3$:

$$\begin{aligned} e_1 &= f_1 f_1^{-1} = (28)(7) = 196, \\ e_2 &= f_2 f_2^{-1} = (105)(1) = 105, \\ e_3 &= f_3 f_3^{-1} = (60)(2) = 120. \end{aligned}$$

Finally, we add it all up to get x:

$$\begin{aligned} x &= e_1 x_1 + e_2 x_2 + e_3 x_3 \bmod 420 \\ &= (196 \cdot 7) + (105 \cdot 2) + (120 \cdot 6) \bmod 420 \\ &= 2302 \bmod 420 \\ &= 202 \end{aligned}$$

All solutions to this system are found by adding multiples of 420 to 202.

Example 2.3 Find all solutions in $\mathbb{Z}/(21)$ for

$$\overline{10}x^2 + \overline{11}x + \overline{3} = \overline{0}.$$

We can factor this to get

$$(2x+1)(5x+3) \equiv 0 \pmod{21}.$$

A solution exists whenever one or both of these binomials is congruent to 0 modulo 21. For example, we get a solution when

$$\begin{aligned} 2x+1 &\equiv 0 \pmod 3 \\ 5x+3 &\equiv 0 \pmod 7. \end{aligned}$$

To find the solution, we solve the following equivalent system.

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 5 \pmod 7 \end{aligned}$$

Applying the method used in the previous example, we find that $f_1 = 7$ and $f_2 = 3$. Continuing, we get $e_1 = 7$ and $e_2 = 15$. Hence, the solution is

$$\begin{aligned} x &= 7(1) + 15(5), \\ &= 82 \pmod{21}, \\ &= 19. \end{aligned}$$

Similarly, we can find other solutions.

2.2 Analysis

For the convenience of the reader, pseudocode for solving systems of linear congruences based on the Chinese Remainder theorem is presented in Figure 1.

The bit complexity of the Chinese Remainder algorithm is characterized by the following result.

```
ChineseRemainder(m_1, m_2, ..., m_k, a_1, a_2, ..., a_k)
 1  m ← m_1 m_2 ··· m_k
 2  x ← 0
 3  for i ← 1 to k
 4  do f_i ← m/m_i
 5     f_i^{-1} ← ExtendedEuclidean(m_i, f_i)_3
 6     ▷ Note that the subscripted 3 above indicates that we are looking for the coefficient
 7     ▷ of f_i found by running the extended Euclidean algorithm.
 8     e_i ← f_i f_i^{-1}  (mod m_i)
 9     x ← x + (e_i a_i)
10  return x
```

Figure 1: Computing the Chinese Remainder Theorem

COROLLARY 2 (5.5.3 in [1]) *Assume that all $m_i > 1$ and that all m_i are pairwise relatively prime. Then we can find a solution in $O((\lg m)^2)$ bit operations.*

Sketch of proof: Based on the assumptions, we know that $k = O(\lg m)$. Consider line 5 of the algorithm in Figure 1. Computing each of the f_i requires $O((\lg m_i)(\lg f_i))$ bit operations. Summing over the k iterations of the loop, we get

$$\begin{aligned}
O\left(\sum_{i=1}^{k} \lg m_i \ \lg f_i\right) &= O\left(\lg m \sum_{i=1}^{k} \lg m_i\right) \\
&= O\left(\lg m \sum_{i=1}^{k} (1 + \log_2 m_i)\right) \\
&= O\left(\lg m \left(k + \sum_{i=1}^{k} \log_2 m_i\right)\right) \\
&= O\left(\lg m \left(k + \log_2 \prod_{i=1}^{k} m_i\right)\right) \\
&= O\left(\lg m (k + \log_2 m)\right) \\
&= O((\lg m)^2).
\end{aligned}$$

Similarly, we can prove the bit complexity of the other steps in the algorithm are $O((\lg m)^2)$.

2.3 The Algebraic Form of the Chinese Remainder Theorem

THEOREM 3 (Chinese Remainder Theorem, version 2, Theorem 5.5.4 in [1]) *Let $m_1, m_2, \ldots, m_k$ be positive integers that are pairwise relatively prime. Then*

$$\mathbb{Z}/(m_1 m_2 \cdots m_k) \cong \mathbb{Z}/(m_1) \oplus \mathbb{Z}/(m_2) \oplus \cdots \oplus \mathbb{Z}/(m_k).$$

Proof: Define $f : \mathbb{Z} \longrightarrow \mathbb{Z}/(m_1 m_2 \cdots m_k) \cong \mathbb{Z}/(m_1) \oplus \mathbb{Z}/(m_2) \oplus \cdots \oplus \mathbb{Z}/(m_k)$ by

$$f(n) = (n \bmod m_1, n \bmod m_2, \ldots, n \bmod m_k).$$

By the Chinese Remainder theorem (Theorem 1), we know that f is onto. (It also tells us that f repeats with period m.) Furthermore, f is a ring homomorphism with $\ker f = (m) \subseteq \mathbb{Z}$. In other words, the kernel of f is the ideal generated by m in $\mathbb{Z}$. Hence, f is an isomorphism from $\mathbb{Z}/(m_1 m_2 \cdots m_k)$ to $\mathbb{Z}/(m_1) \oplus \mathbb{Z}/(m_2) \oplus \cdots \oplus \mathbb{Z}/(m_k)$.

References

[1] E. Bach and J. Shallit, *Algorithmic Number Theory*, The MIT Press, Cambridge, Massachusetts, 1996.