Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 15, Number 5 (2019), pp. 667-673
© Research India Publications
http://www.ripublication.com/gjpam.htm

Proof Methods, Computational Algorithms and Applications of the Chinese Remainder Theorem

Gavin Yu

COSMOS Program, University of California, Davis, California, USA 95616

Piedmont Hills High School, San Jose, California, 95132

Abstract

In this paper, two proof methods of Chinese Remainder Theorem are presented through Direct Proof and Induction on the number of moduli. Three computational numerical algorithms are performed through Direct Computation, Substitution, and Sieving. Also, applications of Chinese Remainder Theorem in Mignotte's threshold secret sharing field are discussed.

1. INTRODUCTION

The Chinese Remainder Theorem first appeared in *Sunzi Suanjing*, a book written by Chinese mathematician Sunzi in the third century. In the book, Sunzi asks for a number that leaves remainders of 2, 3, and 2 when divided by 3, 5, and 7 respectively. Using a special algorithm, he could quickly calculate the number of soldiers he had. Sunzi's book did not contain a proof nor a full method of computing the answer. Later, the theorem was also studied by Indian mathematicians Aryabhata in the sixth century. Aryabhata was the first to develop an efficient method of computing the solution to the congruences.

For Chinese Remainder Theorem, the classical studies dealt with coprime moduli and the existence and the uniqueness of the solution may be proven independently [1-4]. Chinese Remainder Theorem has many developed and protentional applications such as secret sharing, RSA, discrete logarithm problems, elliptic curves cryptography, limited number size computation, *etc* [5-8]. One of the most popular applications is used in Secret sharing field, which is discussed in detail in section four [9-11].

2. PROOF METHODS

In this section, firstly, the existence of a solution to the Chinese Remainder Theorem problem will be proven in two ways. Secondly, further to prove that the solution is unique modulo M, the product of all moduli in the congruences.

Let m_1, m_2, m_3 . . . m_r be r different pairwise relatively prime integers.

$$\begin{aligned} \text{Let } x &\equiv a_1 \bmod m_1 \\ &\equiv a_2 \bmod m_2 \\ &\equiv a_3 \bmod m_3 \\ &\vdots \\ &\equiv a_r \bmod m_r \end{aligned}$$

Let $M \equiv m_1 \times m_2 \times \ldots \times m_r$

To prove that there is exactly one solution of x mod M. Two methods are presented below.

2.1 Existence of a solution (Method 1): Direct Proof

For $1 \le p \le r$, let $M_p = M/m_p$

In other words, M_p is the product of all moduli except for m_p.

For all p such that $1 \le p \le r$, GCD (M_p, m_p) = 1, as all moduli are relatively prime, and M_p does not contain the factor m_p.

Using Bezout's identity, there are integers f_p and z_p such that $M_p \times f_p + m_p \times z_p = 1$. Rearranging and simplifying, $M_p \times f_p \equiv 1 \bmod m_p$ for some integer f_p and for all p such that $1 \le p \le r$.

Then, it is demonstrated that $x = a_1 \times M_1 \times f_1 + a_2 \times M_2 \times f_2 + \ldots + a_r \times M_r \times f_r$ is a solution to all of the congruences. For all q such that $1 \le q \le r$ and $q \ne p$, $a_q \times M_q \times f_q = 0 \bmod m_p$. This is true because by our definition, M_q is a multiple of m_p for $p \ne q$.

Therefore, for all p such that $1 \le p \le r$, $x \equiv a_p \times M_p \times f_p \equiv a_p \bmod m_p$, which means that x satisfies all of the congruences. Therefore, x is a valid solution.

2.2 Existence of a solution (Method 2): Induction on the Number of Moduli

The case of one modulus is trivial, as the solution is just one congruence.

A solution in the case of two moduli is presented below. The congruences are $x \equiv a_1 \bmod m_1$ and $x \equiv a_2 \bmod m_2$.

Since GCD (m_1, m_2) = 1, using Bezout's identity, there are integers n_1 and n_2 such that $m_1 \times n_1 + m_2 \times n_2 = 1$.

Then, it is demonstrated that $x = a_1 \times m_2 \times n_2 + a_2 \times m_1 \times n_1$ is a solution.

Further, $x \equiv a_1 \times m_2 \times n_2 + a_2 \times m_1 \times n_1 \equiv a_1 \times (1 - m_1 \times n_1) + a_2 \times m_1 \times n_1 \equiv a_{1\,+}\, m_1 \times$

$n_1 \times (a_2\text{-}a_1) \equiv a_1 \bmod m_1$. This works similarly for the second congruence.

Let us say that there is a solution if there are *n* congruences. In the case of $n + 1$ congruences, the problem can be reduced to one of *n* congruences by converting any two congruences into one using the process shown above.

This completes the inductive step and shows that there is a solution for any number of congruences greater than or equal to one.

With the above two proof methods of the existence of solution, the further next step proof of uniqueness of the solution (mod *M*) is presented below:

Let x_1 and x_2 be two solutions to the systems of congruences. For all *p*, $(x_1\text{-}x_2)$ is a multiple of m_p, and since all moduli are relatively prime, $(x_1\text{-}x_2)$ is a multiple of $m_1 \times m_2 \times m_3 \times \ldots \times m_r = M$. Then, we have $(x_1 - x_2) \equiv 0 \bmod M$ and $x_1 \equiv x_2 \bmod M$.

Therefore, it is completely proven that there is a solution, and that this solution is unique modulo *M*. Q.E.D.

3. COMPUTATIONAL NUMERICAL ALGORITHMS

In this section, three different computational algorithms of numerical computation, as well as their efficiency will be discussed.

Let us use the following concrete example to solve for x such that:

$$x \equiv 1 \bmod 7$$

$$x \equiv 2 \bmod 8$$

$$x \equiv 3 \bmod 9$$

For each algorithm, we will first present the algorithm solution, and then followed with the above example’s solution for ease of understanding.

3.1 Algorithm 1 - Direct Computation

Direct Computation is the simplest solution, but extremely inefficient. One can check all of the integers from 0 to *M* to see if they satisfy all of the inequalities and will be guaranteed to find the solution.

Example:

First, starting with 0, the number 0 fails to satisfy any of the congruences. Then, moving to 1, this satisfies the first congruence, but not the second or the third. The number 2 satisfies only the second congruence. Keep going, until it reaches 498, when it satisfies all of the congruences.

As seen, Direct Computation algorithm is tedious and very inefficient.

Based on direct computation, it can be further enhanced as an improved direct computation, which is presented below.

Per previous proof of the Chinese Remainder theorem, $x = a_1 \times M_1 \times f_1 + a_2 \times M_2 \times f_2 + \ldots + a_r \times M_r \times f_r$ is a solution to all of the congruences. For all p, M_p can easily be calculated by dividing M by m_p. f_p can be found by solving the congruence $M_\mathrm{p} \times f_p \equiv 1 \bmod m_p$. Then, x can be found by plugging in all known values.

Example:

$a_1 = 1$, $a_2 = 2$, and $a_3 = 3$.

$\mathrm{m}_1 = 7$, $\mathrm{m}_2 = 8$, and $\mathrm{m}_3 = 9$.

$M = 7 \times 8 \times 9 = 504$, so $M_1 = 72$, $M_2 = 63$, and $M_3 = 56$.

Solving the congruence $72 \times f_1 \equiv 1 \bmod 7$, we get $f_1 = 4$.

Solving the congruence $63 \times f_2 \equiv 1 \bmod 8$, we get $f_2 = 7$.

Solving the congruence $56 \times f_3 \equiv 1 \bmod 9$, we get $f_3 = 5$.

Plugging in all values, obviously, $x \equiv 1 \times 72 \times 4 + 2 \times 63 \times 7 + 3 \times 56 \times 5 \equiv 2010 \equiv 498 \bmod 504$.

Although in this way, the direct computation algorithm is improved, it is still inefficient.

3.2 Algorithm 2 - Substitution

Let us use p integers: $d_1, d_2, \ldots d_p$. Then, $x = d_1 \times m_1 + a_1$ and substitute it into the second congruence. After solving for d_1, a similar equation for d_2 in terms of d_1 is formed. Substitute it into the third congruence and keep going. Continue this process repeatedly until all d's are solved. Finally, x can be found by plugging back.

Example:

Using the first congruence, $x = 7 \times d_1 + 1$.

With $7 \times d_1 + 1 \equiv 2 \bmod 8$, the congruence is solved. $d_1 \equiv 7 \bmod 8$, or $d_1 = 8 \times d_2 + 7$.

With substitution into the next congruence, $x \equiv 7 \times (8 \times d_2 + 7) + 1 \equiv 3 \bmod 9$. Then, $d_2 \equiv 8 \bmod 9$, or $d_2 = 9 \times d_3 + 8$.

Now, x is solved. $x = 7 \times (d_1) + 1 = 7\times (8\times (d_2) +7) +1 = 7\times (8\times (9\times (d_3) +8) +7) +1 = 498 + 504 \times (d_3)$, so $x \equiv 498 \bmod 504$.

Obviously, the Substitution algorithm is more efficient than the Direct Computation algorithm.

3.3 Algorithm 3 - Sieving

Start with the congruence with the largest modulus, Sieving algorithm repeatedly add the modulus to the remainder until a value is found. That value satisfies the second congruence. In this way, two congruences can be converted into one. The two congruences can be solved at a time to reduce the number of congruences until only one congruence is left, which is the final solution.

Example:

Start with $x \equiv 3 \bmod 9$. Values of x mod 8 can be listed by adding 9 repeatedly (so it still satisfies the first congruence): 3, 12 (4), 21 (5), 30 (6), 39 (7), 48 (0), 57 (1), 66 (2).

First, it stops at 66, as it satisfies the congruence x ≡ 2 mod 8.

Now having x ≡ 66 mod 72 to replace the first two congruences. Values of x mod 7 can be listed by repeatedly adding 72: 66 (3), 138 (5), 210 (0), 282 (2), 354 (4), 426(6), 498 (1).

Then, it stops at 498, as it satisfies the congruence $x \equiv 1 \bmod 7$.

Finally, the solution comes out that $x \equiv 498 \bmod 504$.

Sieving Algorithm definitely works much more efficient than Direct Computation and Substitution algorithms.

4. APPLICATIONS IN MIGNOTTE'S THRESHOLD SECRET SHARING

In a general secret sharing scheme, each person in a group is given some information, and the solution can only be figured out if more than a certain number of people are present to combine the information which each of the individuals have.

For example, in a simple secret sharing scheme, the answer is locked in a box with many locks on it. Each person is given a certain number of keys, such that all of the locks can be opened only if the people present combined all needed locks. Using the Chinese Remainder theorem, each person is not given a physical key, but rather a congruence involving the answer, which is in the form of a number. Here are the requirements for the scheme: there is a group of n people; To find a solution for any number of people greater than k; Suppose the answer to be a number x.

In Mignotte's threshold secret sharing scheme, n congruences are formed, each one being the value of x mod m_p for $1 \leq p \leq n$. In this scheme, the moduli are determined by the (k, n) - Mignotte sequence. In these types of sequences, the product of the k smallest members of the sequence is larger than the product of the k - 1 largest members, which is exactly the condition necessary for the scheme to work.

For example, the sequence 11, 13, 14, 15, 17, 19 is a (4, 6) - Mignotte sequence, as the product of the four smallest numbers is larger than the product of the three largest numbers of the sequence. In fact, this is also a (1, 6); (2, 6); (3, 6); (5, 6); and (6, 6) Mignotte sequence.

In this scheme, the values of x that are allowed are in the range between the product of the k - 1 largest members and the k smallest members of the sequence. Also, the people are told that the number is smaller than the product of the k smallest moduli, which is essential for the group to reach a unique solution.

Since in this scenario, x is smaller than the product of the k smallest members, it will be smaller than the product of any k members of the sequence. Therefore, the unique solution, which is smaller than the product of these k numbers, can be determined. This means, according to the Chinese Remainder theorem, this solution is guaranteed to exist and be unique.

In another scenario that x is larger than the product of the $k - 1$ largest elements of the set, it will also be larger than the product of any $k - 1$ elements of the set. Since x is larger than the product of these moduli, there are multiple solutions to the given pieces of information and the group of $k - 1$ people cannot determine the exact answer because fewer than k people could get together and use their information to give hints about the solution.

For example, using the (4, 6) - Mignotte sequence 11, 13, 14, 15, 17, 19. Let us say that the answer is 12345. Now each person is given one of the six congruences: $x \equiv 3 \bmod 11$; $x \equiv 8 \bmod 13$; $x \equiv 11 \bmod 14$; $x \equiv 0 \bmod 15$; $x \equiv 3 \bmod 17$; and $x \equiv 14 \bmod 19$. Let us say that the first three people get together. They can't find a solution for certain, but by combining their information and using the Chinese Remainder theorem, they can figure out that the answer is congruent to 333 mod 2002. As they know that the answer is less than 30030, they are left with only 15 possibilities for solutions, rather than 30030. This improves the solution efficiency greatly using Chinese Remainder theorem.

5. CONCLUSION

The Chinese Remainder theorem originated from a problem more than a millennium ago but is still a fundamental piece of mathematics. The proof methods and computational algorithms of Chinese Remainder theorem inspired variance mathematic research and lead to numerous applications in modern math and computer science fields, such as the applications of Mignotte's threshold secret sharing scheme.

ACKNOWLEDGEMENT

This paper was completed with the support and guidance from COSMOS program at University of California, Davis, USA.

REFERENCES

[1] Jones, Burton Wadsworth. The Theory of Numbers. Holt, Rinehart and Winston, 1955.

[2] Ireland, Kenneth F., and Michael Ira Rosen. Elements of Number Theory Including an Introduction to Equations over Finite Fields. Bogden and Quigley, 1972.

[3] Hua, Loo Keng. Introduction to Number Theory. Springer, 1982.

[4] Das, Abhijit. Computational Number Theory. Chapman and Hall/CRC, 2013.

[5] Rivest, Ron., Shamir, Adi., and Adlemen, Leonard. A method for obtaining digital signatures and Public-Key cryptosystems. Communications of the ACM, vol.21,

pp.120–126, 1978.

[6] ElGamal, Taher. A Public-Key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, vol.31, pp.469–472, 1985.

[7] Koblitz, Neal. Elliptic curves cryptography. Mathematics of Computation, vol.48, pp.203-209, 1987.

[8] Miller Victor. Uses of elliptic curves in cryptography. Proc. Lecture Notes in Computer Science 218, Springer, pp.417-426, 1986.

[9] Kraft, James S., and Lawrence C. Washington. An Introduction to Number Theory with Cryptography. Chapman and Hall/CRC, 2018.

[10] Rosen, Kenneth H. Elementary Number Theory and Its Applications. 6th ed., Addison-Wesley, 2011.

[11] Ding, Changming., Pei, Dingyi., and Salomaa, Arto., Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography, World Scientific, Singapore, 1999.