アルゴリズム 12－20
(1989.11.21)

Pairwise Relatively Prime Generating Polynomials and Their Applications

C. C. Chang* and J. C. Shieh**

* Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan, Republic of China

**Institute of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

ABSTRACT

We propose a new linear function of the form T(x)=Dx+1 which can generate a set of pairwise relatively prime numbers from a finite set K. Furthermore, a very efficient algorithm to compute the only one parameter D is proposed. These pairwise relatively prime generating functions can be directly applied to "compute" minimal perfect hashing functions.

1. Introduction

Given a set of n distinct positive integers $K=\{k_1,k_2,\ldots,k_n\}$, theoretically, there exists a linear function of the form T(x)=Dx+E to transform the elements in K into n pairwise relatively prime numbers. Jaeschke [1981] proposed an exhaustive algorithm, named Algorithm DE, to find two integers D and E such that Dk_1+E, Dk_2+E, ..., and Dk_n+E are pairwise relatively prime to one another. Since Jaeschke's method is heuristic, in many cases, it may be unable to find such a function. Besides, the expected computing times for D and E are exponential in n, where n denotes the cardinality of key set K.

Up to now, there is still no efficient algorithm to compute such D and E satisfying that $T(k_1),T(k_2),\ldots,T(k_n)$ are pairwise relatively prime numbers.

In this paper, we shall consider pairwise relatively prime generating polynomials first. Then an efficient algorithm for "computing" these polynomials is presented.

Finally, we shall show that these pairwise relatively prime generating polynomials are useful by applying them to design minimal perfect hashing functions.

2. Pairwise Relatively Prime Generating Polynomials

In this section, a method is proposed for computing pairwise relatively prime generating polynomails of the form T(x)=Dx+1, where x belongs to a set $K=\{k_1,k_2,\ldots,k_n\}$ of n distinct positive integers. For the convenience, in the sequel, we assume that $k_1 < k_2 < \ldots < k_n$.

Before going further we give the formal definition of pairwise relatively prime generating polynomials as follows:

Definition 2.1

T(x) is called a pairwise relatively prime generating polynomial on a set K if T(x) is a polynomial and T(x) and T(y) are relatively prime to each other for x, y ∈ K and T(x) > T(y) if x > y.

For instance, let the set K={4,6,10}. Let T(x)=12x+1. The mapping values for T(x) are 49, 73,121. It is obvious that T(x) is a pairwise relatively prime generating polynomial. Consider the polynomial Dx+1, we ask a question:

For what integer D does the polynomial Dx+1 produce a set of pairwise relatively prime for an arbitrarily given set $K=\{k_1,k_2,\ldots,k_n\}$ of positive integers?

A value of D which answers the above question is $D=\ell cm(\{k_i-k_j \mid 1 \le j < i \le n\})$ denotes least common multiple.

Lemma 2.1

Let x and y be two positive integers and x > y. Let d be a multiple of (x-y). Then d(x-y) and dx+1 are relatively prime to each other.

Proof:

Let gcd(d(x-y), dx+1)=e, where gcd(a,b) means the greatest common divisor of x and y.

If e≠1, that is e ≥ 2, then there must exist a prime number p ≥ 2 such that p|e which means p is a factor of e. This implies p|d(x-y). Since p is a prime number, then we have p|d or p|(x-y). Because d is a multiple of (x-y). We conclude that p|d. This also implies

$$p \mid dx \qquad (1)$$

From the above discussion, it can be easily seen that

$$p \mid (dx+1) \qquad (2)$$

Combining (1) and (2), we have p=1, which is contradictory to p ≥ 2. So we can conclude that e=1. That is d(x-y) and dx+1 are relatively prime to each other.

Q.E.D.

Theorem 2.1

For a set $K=\{k_1,k_2,\ldots,k_n\}$ of n distinct positive integers, let $k_1 < k_2 < \ldots < k_n$, the integer $D=\ell cm(\{k_i-k_j \mid 1 \le j < i \le n\})$, satisfies that $Dk_1+1, Dk_2+1, \ldots, Dk_n+1$ are pairwise relatively prime numbers.

Proof:

For i > j,

$$gcd(Dk_i+1, Dk_j+1)=gcd(D(k_i-k_j), Dk_i+1).$$

Since D is the least common multiple of $\{k_i-k_j \mid 1 \le j < i \le n\}$, it is obvious that D is a multiple of (k_i-k_j), for $1 \le i, j \le n$ and i>j.

By Lemma 2.1, we have $D(k_i-k_j)$ and Dk_i+1 are relatively prime to each other.

That is, $gcd(Dk_i+1, Dk_j+1)=1$.

In other words, $Dk_1+1, Dk_2+1, \ldots, Dk_n+1$ are pairwise relatively prime numbers.

Q.E.D.

Example 2.1

Consider the finite set of keys {3,4,6,7}. Let T(x)=Dx+1

$$D=\ell cm(7-6,7-4,7-3,6-4,6-3,4-3)$$
$$=\ell cm(1,3,4,2,3,1)$$
$$=12.$$

From T(x)=Dx+1, we obtain

$$T(3)=3*12+1=37,$$
$$T(4)=4*12+1=49,$$
$$T(6)=6*12+1=73,$$
and $$T(7)=7*12+1=85.$$

It is clear that, in this case, T(x) is a pairwise relatively prime generating polynomial.

3. Minimal Perfect Hashing Functions

The design of minimal perfect hashing function has been studied by many researchers [Chang 1984, Cichelli 1980, Ghosh 1977, Jaeschke 1981, Sager 1985]. By a minimal perfect hashing function, we mean a one-to-one and onto mapping function h from a set of keys $K=\{k_1,k_2,\ldots,k_n\}$ in the key space to the address space $A=\{0,1,2,\ldots,n-1\}$. The following is an example.

Example 3.1

Assume that the set of keys is {5,33,62,14,21}. Then the function h(k)=k mod 5 is a minimal perfect hashing function. Its minimal perfectness can be checked on the following mapping diagram.

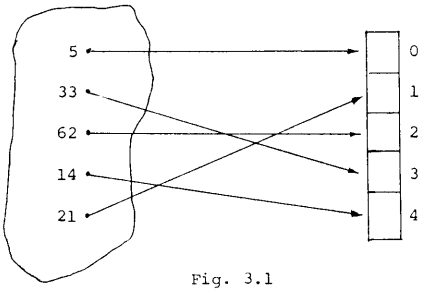

Fig. 3.1

Since a minimal perfect hashing function can be used to organize our records such that no collision occurs, so we may find the record having a given key k quickly. Besides, there is no waste of memory locations in storing these records. Therefore, minimal perfect hashing functions are desirable for many computer applications, such as compiler construction, assemblers design, operating systems design etc.

Recently, Jaeschke [1981] presented the following function $h(k)=\lfloor C/(Dk+E) \rfloor \bmod n$, where n denotes the total number of distinct keys, for minimal perfect hashing function construction. As mentioned in the previous section, Jaeschke [1981] proposed an exhaustive search algorithm, called algorithm DE, to find integers D and E for a given set $K=\{k_1,k_2,\ldots,k_n\}$ of positive integers such that $(Dk_i+E, Dk_j+E)=1$ for $1 \le i, j \le n$ and $i \ne j$. Thus, we may rewrite Jaeschke's hashing function as the form of $h(k)=\lfloor C/T(k) \rfloor \bmod n$, where T(k) is a pairwise relatively prime generating polynomial on K. Recently, Chang and Shieh [1985] proposed an efficient algorithm instead of an exhaustive search method proposed by Jaeschke [1981] to compute C values.

Chang [1984] proposed another minimal perfect hashing scheme of the form h(k)=C mod P(k) based upon the famous Chinese Remainder Theorem, where P(k) is a prime number function. A prime number function is defined as follows: A function P(x) for $a \le x \le b$, where x,a,b are all positive integers, is called a prime number function if P(x) is a prime number for $a \le x \le b$ and $P(x_1) > P(x_2)$ if $x_1 > x_2$. He also reported several prime number function of quadratic forms to transform some particular set of positive keys into their corresponding distinct prime numbers. However, Fendel [1985] showed that there are only seven quadratics can be qualified prime number functions, which are defined on small consecutive integer ranges. Theoretically, in fact, Chang's minimal perfect hashing scheme can be slightly revised into h(k)=C mod T(k), where T(k) is a pairwise relatively prime generating polynomial on K. In [Chang 1984], he presented an efficient way to calculate C values.

From the theoretical point of view, we see the fact that the problem of "computing" either Jaeschke's minimal perfect hashing function or Chang's minimal perfect hashing function is completely equivalent to the problem of "computing" pairwise relatively prime generating polynomials. Thus our pairwise relatively prime generating scheme is suitable for the construction of minimal perfect hashing functions.

4. Conclusions

We proved that T(x)=Dx+1 is a qualified pairwise relatively prime generating scheme. An efficient way is also presented for computing the only parameter D in T(x) on the given key set. We show that our pairwise relatively prime generating polynomials is indeed useful in the area of computing minimal perfect hashing functions. However, the size of the coefficient D used in the function grows exponentially in the cardinality of the given set of keys. This may render the approach impractical. Thus we believe that an research problem in number theory has been opened. Is it possible to have another approach to find the coefficient D such that the representation of its value is drastically smaller than our D value?

REFERENCES

1. Chang, C. C., (1984): The Study of an Ordered Minimal Perfect Hashing Scheme, Communications of the Association for Computing Machinery, Vol. 27, No. 4, April 1984, pp. 384-387.
2. Chang, C. C. and Shieh, J. C., (1985): A Fast Algorithm for Constructing Reciprocal Hashing Functions, the Proceedings of International Symposium on New Directions in Computing, Trondheim, Norway, August 1985, pp. 232-236.
3. Cichelli, R. J., (1980): Minimal Perfect Hash Function Made Simple, Communications of the Association for Computing Machinery, Vol. 23, No. 1, January 1980, pp. 17-19.
4. Fendel, D., (1985): Prime Producing Polynomials and Principal Ideal Domains, Mathematics Magazing, Vol. 58, No. 4, Sept. 1985, pp. 204-210.
5. Ghosh, S. P., (1977): Data Base Organization for Data Management, Academic Press, New York, 1977.
6. Jaeschke, G., (1981): Reciprocal Hashing: A Method for Generating Minimal Perfect Hashing Functions, Communications of the Association for Computing Machinery, Vol. 24, No. 12, December 1981, pp. 829-833.
7. Sager, T. J., (1985): A Polynomial Time Generator for Minimal Perfect Hash Functions, Communication of the Association for Computing Machinery, Vol. 28, No. 5, May 1985, pp. 523-532.