

Dynamical System Representation of Open Address Hash Functions

Gregory L. Heileman[†] Chaouki T. Abdallah[†] Bernard M. E. Moret[‡] Bradley J. Smith[†]
 heileman@eece.unm.edu chaouki@eece.unm.edu moret@cs.unm.edu bsmith@eece.unm.edu

Department of Electrical and Computer Engineering[†]
 Department of Computer Science[‡]
 University of New Mexico, Albuquerque, NM 87131

Introduction. Number theory and probabilistic analyses have typically been used to justify the use of various probing strategies for open address hashing. For example, Guibas and Szemerédi [1], and later Lueker and Molodowitch [3], have shown that double hashing offers a good approximation to uniform hashing if the hash function is assumed to select table entries with equal probability. In this paper we use an analysis based on nonlinear dynamical systems theory to provide a similar result, without making any probabilistic assumptions about the data distribution. The key to our approach is the ability to express hash functions as dynamical systems. Below we demonstrate for the first time how some important families of hash functions can be expressed as dynamical systems. After this, we provide an analysis of linear double hashing that demonstrates the potential usefulness of representing hash functions in this fashion.

Hashing Functions as Dynamical Systems. We will consider three important families of hash functions: quadratic hashing, linear double hashing, and exponential double hashing. Quadratic hashing is an extension of linear probing that makes the probe sequence nonlinearly dependent on the probe number n . For any ordinary hash function h that performs a mapping from a key space to a table space of size m , the family of quadratic hash functions is given by

$$H_{\mathcal{Q}_n} = (h(k) + c_1n + c_2n^2) \bmod m, \quad (1)$$

where k is the key, and c_1 and c_2 are positive constants. The specific values chosen for the constants are critical to the performance of this method [2]. To obtain a recurrence relation solution to (1) we note that

$$H_{\mathcal{Q}_{n+1}} = \left[(h(k) + c_1n + c_2n^2) + (c_1 + c_2(2n + 1)) \right] \bmod m,$$

which allows us to derive a complete state-space description of this system given by the following set of coupled linear (modulo m) time-invariant first-order difference equations:

$$x_{n+1} = (x_n + 1) \bmod m, \quad y_{n+1} = (y_n + 2c_2x_n + c_2 + c_1) \bmod m, \quad (2)$$

where $x_0 = 0$, $y_0 = h(k)$, and $H_{\mathcal{Q}_i} = y_n$ is the output equation used to iterate over the table space.

Given two hash functions g and h , the family $H_{\mathcal{LD}}$ of linear double hash functions is given by:

$$H_{\mathcal{LD}_n} = (g(k) + nh(k)) \bmod m. \quad (3)$$

where the initial probe $H_{\mathcal{LD}}(0) = g(k)$. Thus the probe sequence depends on k through both g and h , and is linear in $g(k)$ and $h(k)$.

A recurrence relation for the family $H_{\mathcal{LD}}$ is given by the state-space description of the system as a set of coupled linear (modulo m) time-invariant first-order difference equations:

$$x_{n+1} = x_n, \quad y_{n+1} = (x_n + y_n) \bmod m, \quad (4)$$

where $x_0 = h(k)$, $y_0 = g(k)$, and $H_{\mathcal{LD}_n} = y_n$ is the output equation used to iterate over the table space. A widely used member of $H_{\mathcal{LD}}$, proposed by Knuth [2], has $g(k) = k \bmod m$ and $h(k) = k \bmod (m - 2)$, where both m and $m - 2$ are prime.

We propose a new family of double hash functions $H_{\mathcal{E}}$, which we call exponential double hashing, that uses two ordinary hash functions g and h to compute a probe sequence according to

$$H_{\mathcal{E}_n} = (g(k) + h(k)^n) \bmod m. \quad (5)$$

The exponentiation does not have to be explicitly implemented, but can be computed via successive multiplications during the probing process. For this reason, the number of mathematical operations needed to implement (5) is identical to the number needed to implement (3). To obtain the desired transformation for the family $H_{\mathcal{E}}$ described in (5), we write

$$\begin{aligned} H_{\mathcal{E}_{n+1}} &= (h(k)h(k)^n + g(k)) \bmod m \\ &= [((h(k)^n + g(k)) \bmod m \cdot h(k) \bmod m) \bmod m + ((1 - h(k))g(k)) \bmod m] \bmod m \end{aligned}$$

and the transformation we obtain for $H_{\mathcal{E}}$ is

$$H_{\mathcal{E}_{n+1}} = (H_{\mathcal{E}_n} h(k) + (1 - h(k))g(k)) \bmod m \quad (6)$$

$$H_{\mathcal{E}_0} = g(k) + 1. \quad (7)$$

This is a first-order time-varying nonlinear system. A first-order time-invariant nonlinear system can be obtained by using a three-dimensional state space, as is demonstrated by the following triple of difference equations:

$$x_{n+1} = x_n, \quad y_{n+1} = y_n, \quad z_{n+1} = (y_n z_n + x_n (1 - y_n)) \bmod m, \quad (8)$$

where $x_0 = g(k)$, $y_0 = h(k)$, $z_0 = x_0 + 1$, and $H_{\mathcal{E}_n} = z_n$ is the output equation used to iterate over the table space. A particular member of $H_{\mathcal{E}}$, recently introduced by Smith et al. [4], has $g(k) = k \bmod m$, and $h(k) = k \bmod (m - 2)$, where $m = 2p + 1$ is selected so that both m and p are prime. Smith et. al [4] showed experimentally that on average this member of $H_{\mathcal{E}}$ tends to outperform any member of $H_{\mathcal{LD}}$. We suggest here that this may be due to the fact that $H_{\mathcal{E}}$ hash functions operates in a more complicated state space than $H_{\mathcal{LD}}$ hash functions.

Analysis. We now consider an analysis that makes use of the dynamical system representation of $H_{\mathcal{LD}}$. Given (4), the output equation can be rewritten in terms of the initial probes as $y_n = y_0 + nx_0 \pmod{m}$. Let us denote the joint density function for the initial probes using f_{x_0, y_0} , and without loss of generality (by appropriate scaling) we may assume $m = 2\pi$. In this case the characteristic function of the output equation is given by

$$\varphi_{y_n}(t) = \mathbb{E}e^{ity_n} = \mathbb{E}e^{it(y_0 + nx_0)} = \int_0^{2\pi} e^{it(v+nu)} f_{x_0, y_0}(u, v) du dv \quad (9)$$

By the Riemann-Lebesgue Lemma, it follows that for $t \neq 0$, $\varphi_{y_n}(t) \rightarrow 0$ weakly as $n \rightarrow \infty$. Thus, since f_{x_0, y_0} is a density function, for n sufficiently large $\varphi_{y_n}(t) = 1$ when $k = 0$, and $\varphi_{y_n}(t) = 0$ when $k \neq 0$, which is in fact the characteristic function for the uniform distribution on the circle. Since the characteristic function uniquely determines the density function for a given random variable, this demonstrates that for an appropriately chosen m , and after a sufficient number of probes, hash functions in the linear double hashing family will transform any initial density into the uniform distribution over the table space. \diamond

References.

- [1] L. Guibas and E. Szemerédi. The analysis of double hashing. *Journal of Computer and Systems Sciences*, 16:226–274, 1978.
- [2] D. E. Knuth. *Searching and Sorting*, volume 3, *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 1973.
- [3] G. Lueker and M. Molodowitch. More analysis of double hashing. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, pages 354–359, 1988.
- [4] B. J. Smith, G. L. Heileman, and C. T. Abdallah. The exponential hash function. *ACM Journal of Experimental Algorithmics*, 2(3):www.jea.acm.org/1997/SmithExponential/, 1997.