THE SOLUTION OF AN ITERATED RECURRENCE

D. S. MEEK and G. H. J. VAN REES*
University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2
(Sumbitted October 1981)

1. INTRODUCTION

Hofstadter [1, p. 137] defines the following iterated recurrence,

$$g(0) = 0,\ g(n) = n - g^r(n - 1),\ n = 1, 2, \ldots,$$

where $g^r(n)$ denotes the iterated function

$$\overbrace{g(g(\ldots(g(n))\ldots)}^{r\text{ levels}}.$$

He does not show how to determine the values of this irregular function. In this paper, we will show that the solution to the iterated recurrence can be given as a simple truncation function on numbers written in a generalized Fibonacci base.

First, for convenience, we will change the iterated recurrence by a translation of the origin. The iterated recurrence to be studied is the following:

$$g(0) = 0 \tag{1a}$$

$$g(n) = n - 1 - g^r(n - 1) \tag{1b}$$

The values of $g(n)$ for $r = 1, 2, 3,$ and 7 are tabulated below.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$g(n)$	0	0	1	1	2	2	3	3	4	4	5	5	6

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$g(n)$	0	0	1	2	2	3	3	4	5	5	6	7	7

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$g(n)$	0	0	1	2	3	3	4	4	5	6	6	7	8

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$g(n)$	0	0	1	2	3	4	5	6	7	7	8	8	9

If $r = 1$, it is clear that $g(n)$ is the integer part of $\frac{1}{2}n$, but for larger r it is more irregular. However, in the next section we will show that, if n is expressed in the appropriate Fibonacci base, then $g(n)$ is a simple truncation function.

This project was supported, in part, by NSERC Grant No. A4359 (Meek) and NSERC Grant No. A3558 (van Rees).

2. SOLUTION OF THE RECURRENCE

Define the "radix" b_i as:

$$b_i = i, \; i = 1, 2, \ldots, r \tag{2a}$$

$$b_i = b_{i-1} + b_{i-r}, \; i = r + 1, r + 2, \ldots . \tag{2b}$$

If r is 1, then $b_i = 2^i$, and we get the binary base in which all numbers have a unique positional representation of zeros and ones. If $r = 2$, then we have, for example, the following representations:

$$9 = 10001 = 1 \times 8 + 0 \times 5 + 0 \times 3 + 0 \times 2 + 1 \times 1$$

$$10 = 10010 = 1 \times 8 + 0 \times 5 + 0 \times 3 + 1 \times 2 + 0 \times 1.$$

And for $r = 3$:

$$9 = 100000 = 1 \times 9 + 0 \times 6 + 0 \times 4 + 0 \times 3 + 0 \times 2 + 0 \times 1$$

$$10 = 100001 = 1 \times 9 + 0 \times 6 + 0 \times 4 + 0 \times 3 + 0 \times 2 + 1 \times 1.$$

Note that if $r > 1$, the representation is not unique. When $r = 3$, for example, 10 could also be expressed as:

$$10 = 11000 = 1 \times 6 + 1 \times 4 + 0 \times 3 + 0 \times 2 + 0 \times 1.$$

However, the representation can be made unique as follows. To represent a positive number n, find the largest b_i that is less than or equal to n. The representation of n will have a one in the i^{th} digit. Now find the largest b_j less than or equal to $n - b_i$. The representation will also have a one in the j^{th} digit. This process of reduction is continued until n equals a sum of distinct "radix" numbers b_i. Then n will be represented in this base by $a_k a_{k-1} \ldots a_2 a_1$ where a_i, $i = 1, 2, \ldots, k$ is one or zero, depending on whether or not b_i is present in the sum. This will be called the normalized form of the number in this base.

The recurrence (2b) generates what have been called "generalized Fibonacci numbers." So we will call these bases "generalized Fibonacci bases."

A function which removes or truncates the last digit of a number n represented in a generalized Fibonacci base will be denoted by $T(n)$. If $n = a_k a_{k-1} \ldots a_2 a_1$ or, equivalently,

$$n = \sum_{i=1}^{k} a_i b_i,$$

then

$$T(n) = a_k a_{k-1} \ldots a_2 = \sum_{i=1}^{k-1} a_{i+1} b_i.$$

We will define $T(0)$ to be 0.

For example, if $n = 10$, then in the Fibonacci base with $r = 2$, $10 = 10001$ and

$$T(10001) = 1000 = 1 \times 5 + 0 \times 3 + 0 \times 2 + 0 \times 1 = 5.$$

In the binary base with $r = 1$,

$$T(10001) = 1000 = 8.$$

Theorem: The solution to the iterated recurrence

$$g(0) = 0$$
$$g(n) = n - 1 - g^r(n - 1),\ n \geqslant 1,$$

is $g(n) = T(n)$, where $T(n)$ is the truncation function.

3. PROOF OF THEOREM

The function $T(n)$ obviously satisfies the condition (1a). To satisfy (1b), $T(n)$ must equal $n - 1 - T^r(n - 1)$. The following lemma shows this equality.

Lemma: If m and $m + 1$ are written in a generalized Fibonacci base, then

$$T(m + 1) = m - T^r(m). \tag{3}$$

Proof: Let m be represented in normalized form by

$$a_k a_{k-1} \ldots a_{r+1} a_r a_{r-1} \ldots a_2 a_1 \text{ (k digits).} \tag{4}$$

Writing

$$m = \sum_{i=1}^{k} a_i b_i = \sum_{i=1}^{r} a_i b_i + \sum_{i=r+1}^{k} a_i b_i,$$

the relation (2b) can be used on the second sum to show

$$m = \sum_{i=1}^{r} a_i b_i + \sum_{i=r}^{k-1} a_{i+1} b_i + \sum_{i=1}^{k-r} a_{r+i} b_i.$$

Since the last sum is the value of $T^r(m)$, the right-hand side of (3) equals

$$a_k a_{k-1} \ldots a_{r+2}(a_{r+1} + a_r)a_{r-1} \ldots a_2 a_1 \text{ ($k - 1$ digits).} \tag{5}$$

Note that this number might not be in normalized form.

The representation of $m + 1$ can be found by first noting that at most one of the a_i, $i = 1, 2, \ldots, r$ is a 1 in (4). Three cases to consider are: $a_i = 0$ for all $i = 1, 2, \ldots, r$; $a_i = 1$ for some $i < r$; and $a_r = 1$. In the first case, the representation of $m + 1$ will be like (4) but with $a_1 = 1$. This representation is in normalized form, so $T(m + 1)$ is

$$a_k a_{k-1} \ldots a_{r+2} a_{r+1} a_r \ldots a_3 a_2 \quad \text{($k - 1$ digits).}$$

Since $a_i = 0$, $i = 2, 3, \ldots, r$, this is identical to (5). In the second case, $a_i = 1$ for some $i < r$, and $m + 1$ has a one in the $i + 1^{\text{st}}$ digit. Now $T(m + 1)$ can be found even though $m + 1$ as described is not necessarily normalized. It has representation (5). In the third case, where $a_r = 1$, $m + 1$ has the digits 1 to r all zeros and a one is added to the digit a_{r+1}. Thus $T(m + 1)$ is again as shown in (5). ∎

4. CONCLUDING REMARKS

If $g(n)$, for some large n, has to be calculated, the straightforward recursive method for doing so requires the calculation of all $g(i)$ numbers for $i \leqslant n$.

However, it can be done efficiently by calculating the "radix" b_i numbers using (2), finding the representation of the number as described in ¶2 of §2, using $T(n)$ to obtain the answer, and then converting the answer back to base 10. If

$$Z = \lim_{1 \to \infty} \frac{b_{i+1}}{b_i},$$

then this method takes approximately 3 $\log_Z n$ steps.

A closed form solution for (1) seems impossible to obtain for $r \geqslant 2$, but a good approximation to $g(n)$ is n/Z.

Finally, the theorem can be generalized by noting that the iterated recurrence

$$g(A) = A, \; A \text{ an integer}$$
$$g(n) = n - 1 + A - g^r(n - 1), \; n \geqslant A + 1$$

has solution $g(n) = T(n - A) + A$, for $n \geqslant A$. For $A = 1$, this gives a solution to Hofstadter's original problem.

ACKNOWLEDGMENT

The authors thank the referee for helpful comments on the presentation of this material.

REFERENCE

1. D. R. Hofstadter. *Godel, Esher, Bach: An Eternal Golden Braid*. New York: Basic Books Inc., 1979.

◆◇◆◇◆