MATHEMATICS OF COMPUTATION
VOLUME 54, NUMBER 189
JANUARY 1990, PAGES 377-394

# SOME QUESTIONS OF ERDÖS AND GRAHAM ON NUMBERS OF THE FORM $\sum g_n/2^{g_n}$

P. B. BORWEIN AND T. A. LORING

ABSTRACT. Erdös in 1975 and Erdös and Graham in 1980 raised several questions concerning representing numbers as series of the form $\sum_{n=1}^{\infty} g_n/2^{g_n}$. For example, does the equation

$$\frac{n}{2^n} = \sum_{k=1}^{T} \frac{g_k}{2^{g_k}}, \qquad T > 1,$$

have a solution for infinitely many $n$? The answer to this question is affirmative; in fact, we conjecture that the above equation is solvable for every $n$. This conjecture is based on a more general conjecture, namely that the algorithm

$$a_{n+1} = 2(a_n \bmod n)$$

with initial condition $a_m \in \mathbf{Z}$ always eventually terminates at zero. This, in turn, is based on an examination of how the "greedy algorithm" can be used to represent numbers in the form $\sum g_n/2^{g_n}$. The analysis of this, reformulated as a "base change" algorithm, proves surprising. Some numbers have a unique representation, as above, others have uncountably many. Also, from this analysis we observe that $\sum g_n/2^{g_n}$ is irrational if $\limsup_n((g_{n+1}-g_n)/\log(g_{n+1})) = \infty$ and conjecture that this is best possible.

## 1. INTRODUCTION

In [3] Erdös and Graham raise the following three questions. Does the equation

$$\frac{n}{2^n} = \sum_{k=1}^{T} \frac{g_k}{2^{g_k}}, \qquad T > 1, \tag{1.1}$$

have a solution for infinitely many $n$? For all $n$? (Here, $\{g_n\}$ is a strictly increasing sequence of positive integers.) Is there a rational $x$ for which

$$x = \sum_{k=1}^{\infty} \frac{g_k}{2^{g_k}} \tag{1.2}$$

has two solutions? Does there exist a rational $x$ for which

$$x = \sum_{k=1}^{\infty} \frac{g_k}{2^{g_k}} \tag{1.3}$$

Received December 8, 1988.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 11A63, 11D68, 11-04.
Research of the first author was supported by N.S.E.R.C. of Canada.
Research of the second author was supported in part by the Killam Trust.

and for which

$$\limsup_n (g_{n+1} - g_n) = \infty?$$

They speculate that the answer is positive to the last question. This would be best possible. In [1] Erdös resolves a twenty-year-old problem by showing that

$$(1.4) \qquad \sum \frac{g_n}{2^{g_n}} \quad \text{with } \lim(g_{n+1} - g_n) = \infty$$

is irrational. He asks in [2] whether the greedy algorithm always generates a representation for a rational $x$ with $g_{n+1} - g_n$ bounded. We discuss this further in §3.

We choose to approach these problems from the following point of view. Given

$$(1.5) \qquad \alpha = \sum_{n=1}^{\infty} \frac{b_n}{2^n}, \qquad b_n = 0, 1,$$

how can we represent $\alpha$ as

$$(1.6) \qquad \alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n}, \qquad d_n = 0, 1 ?$$

We call a representation of $\alpha$ as in (1.6) a $*$-binary representation and call $d_n$ the $n$th $*$-binary digit. It is in this sense that we think of the above problems as being "base change" problems. From this point of view the following questions are suggested.

**[Q1]** Does every $\alpha \in [0, 2]$ have a $*$-binary representation?

The answer, as observed in [1], is that the "greedy algorithm" always provides such a representation. In this context the greedy algorithm is the algorithm that, inductively, sets $d_N := 1$ if $\sum_{n=1}^{N} nd_n/2^n \leq \alpha$ and sets $d_N := 0$ otherwise. This algorithm, as we shall see, converges. In §2 we offer two reformulations of the greedy algorithm that are more amenable to analysis. The second natural question is:

**[Q2]** When does $\alpha \in [0, 2]$ have a unique $*$-binary representation?

It is apprarent from the existence of nontrivial solutions to (1.1) (see Proposition 1) that $*$-binary representations are not always unique. Uniqueness is, however, possible. For example, 1/72 has a unique $*$-binary representation. Other numbers have uncountably many different $*$-binary representations. Uniqueness questions will be primarily dealt with in §3. In particular, there is a systematic way of modifying the "base change" algorithm of §2 to generate all representations.

The third natural question is:

**[Q3]** When does $\alpha$ have a finite $*$-binary representation?

This is a question in Diophantine equations. It asks when we can solve

$$(1.7) \qquad \alpha = \sum_{n=1}^{N} \frac{nd_n}{2^n}, \qquad d_n = 0 \text{ or } 1 .$$

Clearly, $\alpha$ must be an exact binary fraction for such a solution to exist. We conjecture, perhaps surprisingly, that this is also sufficient. This, as we shall see, follows from the following conjecture.

**Conjecture 1.** Let $\eta$ be any integer. Let

$$a_m := \eta \quad \text{and} \quad a_{n+1} := 2(a_n \bmod n), \qquad n = m, m+1, \ldots$$

(where $(a \bmod n)$ is always chosen in the interval $[0, n-1]$). Then for some $N_m$, there holds $a_n = 0$, for $n \geq N_m$ (that is, the above iteration always terminates).

This conjecture, if true, also totally resolves the third question of Erdös ((1.3) above) by showing that every diadic rational has a representation of the form (1.3) with arbitrarily large gaps (i.e., strings of zeros). The evidence for Conjecture 1, and a discussion of its consequences, is the content of §5. Section 4 concerns arbitrary base analogues of these questions.

## 2. The base change algorithm

The two reformulations of the greedy algorithm we offer are:

**Algorithm 1.** *Let*

$$\alpha = \sum_{n=1}^{\infty} \frac{b_n}{2^n}, \qquad b_n = 0, 1.$$

*Let*

$$a_1 := b_1 \quad \textit{and} \quad a_{n+1} := 2(a_n \bmod n) + b_n$$

(*where* $(a_n \bmod n)$ *is chosen in* $[0, n-1]$). *Then*

$$\alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n},$$

*where*

$$d_n = \begin{cases} 0 & \textit{if } a_n < n, \\ 1 & \textit{if } a_n \geq n. \end{cases}$$

**Algorithm 2.** *Let* $\alpha \in [0, 2)$. *Let* $e_1 = 2\alpha$ *and*

$$e_{n+1} = \begin{cases} 2(e_n - n) & \textit{if } e_n \geq n, \\ 2e_n & \textit{if } e_n < n. \end{cases}$$

*Then*

$$\alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n},$$

*where*

$$d_n = \begin{cases} 0 & \textit{if } e_n < n, \\ 1 & \textit{if } e_n \geq n. \end{cases}$$

In fact, if $\alpha \in [0, 1)$ (and that in the event $\alpha$ is an exact binary fraction it is represented by its terminating representation) then Algorithms 1 and 2 have the same output. In this case, $a_n = \text{int part}(e_n)$.

We shall sometimes refer to this representation as the canonical $*$-binary representation. Observe that Algorithm 1 is very efficient, both practically and theoretically. The bit complexity of converting $n$ binary digits to $n$ $*$-binary digits is $O(n\log n)$ (which is better than the best known algorithm for converting base 2 to base 3 or, for that matter, multiplication [4]).

*Proof of Algorithm* 1. Let $\{\delta_n\}$ be any sequence of zeros and ones. Then

$$\alpha=\sum_{m=1}^{n-1}\frac{m\delta_m}{2^m}+\frac{a_n}{2^n}+\sum_{m=n+1}^{\infty}\frac{b_m}{2^m}, \tag{2.1}$$

where

$$a_1:=b_1$$

and

$$a_{n+1}:=2(a_n-\delta_n n)+b_{n+1}. \tag{2.2}$$

This is easily verified by induction on $n$ and the observation that

$$2n\delta_n=2a_n-a_{n+1}+b_{n+1}$$

or equivalently

$$\frac{n\delta_n}{2^n}+\frac{a_{n+1}}{2^{n+1}}=\frac{a_n}{2^n}+\frac{b_{n+1}}{2^{n+1}}.$$

If we now inductively define $\{\delta_n\}$ and $\{a_n\}$ by

$$a_1:=b_1,$$
$$\delta_n:=\begin{cases}0 & \text{if } a_n<n,\\ 1 & \text{if } a_n\ge n\end{cases}$$

and

$$a_{n+1}:=2(a_n \bmod n)+b_n,$$

then we derive Algorithm 1. The proof of convergence of Algorithm 1 now is reduced to showing that in (2.1)

$$a_n/2^n\to 0.$$

However, by construction,

$$a_{n+1}\le 2(n-1)+1=2n-1,$$

and we are done.

That this algorithm is actually the greedy algorithm (except for nonterminating representations of diadic rationals) follows from the inequality

$$\alpha-\sum_{m=1}^{n-1}\frac{m\delta_n}{2^m}=\frac{a_n}{2^n}+\sum_{m=n+1}^{\infty}\frac{b_m}{2^m}<\frac{2n-3}{2^n}+\frac{1}{2^n}=\frac{n-1}{2^{n-1}}, \tag{2.3}$$

which shows, again inductively, that at every $n$, every term that can be included in the representation has been included. □

*Proof of Algorithm* 2. Algorithm 2 is easily reduced to Algorithm 1 on writing $\alpha=\sum_{n=1}^{\infty}b_n/2^n$ (using the terminating representation if possible). One need only observe that the remainder $\sum_{n=N+1}^{\infty} b_n/2^n$ has no effect on the $N$th step of the algorithm. □

**Corollary 1.** *Conjecture* 1 *implies that every diadic rational has a terminating* $*$*-binary representation.*

*Proof.* If $\alpha = \sum_{n=1}^{M} b_n/2^n$, then for $m \geq M$, Algorithm 1 reduces to the algorithm of Conjecture 1. □

We now wish to exhibit an infinite class of $m$ for which

$$\frac{m-1}{2^{m-1}} = \sum_{k=m}^{T} \frac{k}{2^k},$$

thus resolving the first question of the introduction. The approach is the following. Since

$$\frac{m-1}{2^{m-1}} = \frac{2m-2}{2^m} = \frac{m}{2^m} + \frac{m-2}{2^m}, \tag{2.4}$$

we have

$$\frac{m-1}{2^{m-1}} = \frac{m}{2^m} + \sum_{n \geq m+1} \frac{nd_n}{2^n}, \tag{2.5}$$

where the $d_n$ are the output of Algorithm 1 applied to $(m-2)/2^m$. The corresponding $\{a_i\}$ are generated by

$$a_m := m - 2 \tag{2.6}$$

and

$$a_{n+1} := 2(a_n \bmod n).$$

Observe, for large $m$, that the initial few terms $a_m$, $a_{m+1}$, $a_{m+2}, \ldots$ are

$$m-2,\ 2m-4,\ 2m-10,\ 2m-24, \ldots, \tag{2.7}$$

where, if

$$a_{m+k} := 2m - \delta_k > m + k, \tag{2.8}$$

then

$$a_{m+k+1} = 2(2m - \delta_k \bmod m + k) = 2m - 2\delta_k - 2k.$$

So

$$\delta_{k+1} = 2\delta_k + 2k \quad \text{where } \delta_0 = 2. \tag{2.9}$$

Suppose for some $K$ that

$$a_{m+K} = m + K$$

while

$$a_{m+k} > m + k, \qquad k < K.$$

Then Algorithm 1 outputs a sequence of $d_l$ with

$$d_{m+1} = d_{m+2} = \cdots = d_{m+K-1} = 1$$

and all other $d_i = 0$ and produces a $*$-representation of $(m-1)/2^{m-1}$ of the requisite form. The trick now is to solve the recursion (2.9) for $\delta_k$. If

$$f(x) = \sum_{k=1}^{\infty} \delta_k x^k ,$$

then

$$\frac{f(x)}{x} = 2f(x) + 2\sum_{k=1}^{\infty} kx^k + \delta_1 \tag{2.10}$$

or, after some effort,

$$f(x) = \frac{4x+2}{1-2x} - \frac{2}{(1-x)^2} = \sum (2^{k+2} - 2(k+1))x^k . \tag{2.11}$$

In particular,

$$a_{m+K} = 2m + 2(K+1) - 2^{K+2} = m + K$$

exactly when

$$m = 2^{(K+2)} - (K+2) \tag{2.12}$$

for some $K$. Packaging this, yields:

**Proposition 1.** *For $m = 2^M - M$, $M \geq 2$, there holds*

$$\frac{m-1}{2^{m-1}} = \sum_{k=m}^{m+M-2} \frac{k}{2^k} . \tag{2.13}$$

This derivation of Proposition 1 also shows that no other identities of the form

$$\frac{c-1}{2^{c-1}} = \sum_{k=c}^{c+d} \frac{k}{2^k}$$

exist. Of course, (2.13), once discovered, can be proved directly. From the identity

$$\sum_{k=A}^{B-1} kx^{k-1} = \left(\frac{1-x^B}{1-x}\right)' - \left(\frac{1-x^A}{1-x}\right)'$$

we have

$$\sum_{k=A}^{B-1} kx^k = \frac{x^A(A - Ax + x) - x^B(B - Bx + x)}{(x-1)^2}$$

and

$$\sum_{k=A}^{B-1} \frac{k}{2^k} = \frac{A+1}{2^{A-1}} - \frac{B+1}{2^{B-1}} . \tag{2.14}$$

Proposition 1 now follows from (2.14) on setting $A := 2^M - M$, $B = 2^M - 1$ and $C = 2^M - M - 1$, whence

$$\frac{C}{2^C} = \frac{A+1}{2^{A-1}} - \frac{B+1}{2^{B-1}} = \sum_{k=A}^{B-1} \frac{k}{2^k} .$$

From Proposition 1 it is clear that $*$-binary representations are not unique, and thus not every $*$-binary representation arises as the output of the greedy algorithm. We shall see in §5 that for terminating representations the greedy algorithm does not necessarily generate the shortest representation.

**Proposition 2.** *Suppose $\alpha \in [0, 1)$ has the (canonical) $*$-binary and binary representations*

$$\alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n} = \sum_{n=1}^{\infty} \frac{b_n}{2^n}, \qquad d_n, b_n = 0, 1.$$

*Let $Z_m(\alpha)$ denote the length of the longest consecutive string of zeros among the first $m$ of the $d_i$, and let $O_m(\alpha)$ denote the longest such string of ones. Let $z_m(\alpha)$ and $o_m(\alpha)$ denote the corresponding counts for the $b_i$. Then,*

$$O_m(\alpha) \le 1 + \log_2 m + o_m(\alpha) \quad \textit{and} \quad Z_m(\alpha) \le 1 + \log_2 m + z_m(\alpha).$$

*Proof.* (We do not need to assume we are using the canonical representation, but then we must use Proposition 4 of the next section.) The longest possible string of zeros in the output of Algorithm 1 arises from

$$a_{m-s} = a_{m-s+1} = \cdots = a_{m-1} = 0 \quad \text{and} \quad a_m = 1, b_m = \cdots = b_{m+k} = 0$$

with

$$a_{m+1} = 2a_m < m + 1, \ldots, a_{m+k} = 2^k a_m < m + k.$$

The first sequence above is of length less than or equal $z_m(\alpha)$, since the $a_i$ can only stay at zero if the corresponding $b_i$ are zero. The number of terms following $a_m = 1$ for which $d_m = 0$ is less than the smallest $K$ for which $2^K > m + K$. Since $K = 1 + \log_2 m$ satisfies the above, we are done.

A similar argument applies for $O_m$. □

**Corollary 2.** *If $\alpha$ is rational, then for some constant $C$,*

$$O_m(\alpha) \le \log_2(m) + C,$$

*and if $\alpha$ is not a diadic rational, then for some constant $D$,*

$$Z_m(\alpha) \le \log_2(m) + D.$$

Corollary 2 has an easy direct proof: if $\alpha$ has gaps of length $\gg \log n$ at the $n$th $*$-digit, then

$$2^n\alpha = \text{integer } + O(1)$$

and $\alpha$ cannot be rational.

## 3. Multiple representations

Consider the numbers

$$B_M := \sum_{m=M}^{\infty} \frac{2^m - m - 1}{2^{2^m - m - 1}} = \sum_{m=M}^{\infty} \frac{g_m}{2^{g_m}}, \tag{3.1}$$

where $g_m := 2^m - m - 1$. Then there is a systematic way of replacing each $g_m/2^{g_m}$ term using the identity (2.13), namely

$$\frac{g_m}{2^{g_m}} = \sum_{k=g_m+1}^{g_m+m-1} \frac{k}{2^k}.$$

Since there are infinitely many independent choices to make, the total number of resulting representations is uncountable.

**Proposition 3.** (a) *There exists a dense set of irrationals, each with uncountably many different* $*$*-binary representations.*

(b) *If Conjecture* 1 *holds, then every diadic rational has infinitely many terminating representations* (*so that the Diophantine equation* (1.7) *has infinitely many solutions*).

(c) *Let* $A := [0, 1] - \{\textit{diadic rationals}\}$. *The set of* $\alpha \in A$ *with more than* $k$ *different* $*$*-binary representations is open and dense in* $A$ (*in the topology on* $A$).

(d) *If Conjecture* 1 *holds, then the set of* $\alpha \in [0, 1]$ *with more than* $k$ *different* $*$*-binary representations is open and dense in* $[0, 1]$.

*Proof.* Part (a) follows from the observation that any number of the form

$$\sum_{n=1}^{m} \frac{nd_n}{2^n} + B_M, \qquad m < M, \; B_M \text{ as in } (3.1), \tag{3.2}$$

has uncountably many different representations. Since by construction and Corollary 2, $B_M$ is irrational, it follows that the above numbers are irrational. The denseness of numbers of the form (3.2) follows from the fact that

$$\lim_{M\to\infty} B_M = 0$$

and the observation that (2.3) of Algorithm 1 guarantees that

$$\left\{ x := \sum_{n=1}^{m} \frac{nd_n}{2^n} \,\middle|\, d_n = 0, 1 \right\}$$

is dense in $[0, 1]$.

For part (b) we observe, as in (2.5), that

$$\frac{m-1}{2^{m-1}} = \frac{m}{2^m} + \sum_{n\geq m+1} \frac{nd_n}{2^n},$$

where the sum is the output of Algorithm 1 applied to $(m-2)/2^m$. If Conjecture 1 holds, this sum is finite. In particular, each finite representation of a diadic rational can be extended to a new finite representation just by using the above procedure to rewrite the highest nonzero term. The finiteness at each stage follows from the conjecture, as in Corollary 1.

Part (c) and part (d) would be the same if we knew that every diadic rational had infinitely many different $*$-binary representations. (Our problem is dealing with representations that end in an infinite string of ones, which by (2.14) can only occur for diadic rationals.) We prove openness as follows. Suppose $\alpha \in A$

has $k+1$ different $*$-representations. Then, for some $N_1$, these representations all differ at one of the first $N_1$ "digits". Furthermore, for some $N_2 > N_1$, each of these representations has a zero "digit" between $N_1$ and $N_2$ (here we have used the assumption that $\alpha$ is not diadic—if we knew that a diadic had infinitely many different $*$-representations we could proceed anyway). Thus for a particular one of these representations of $\alpha$, we have

$$\alpha := \sum_{n=1}^{N_3-1} \frac{nd_n}{2^n} + \sum_{n=N_3+1}^{\infty} \frac{nd_n}{2^n},$$

where $N_1 < N_3 < N_2$ is a zero term for this representation. We now show that if

$$|\alpha - \beta| < \frac{1}{2^{N_3}},$$

then

$$\beta = \sum_{n=1}^{N_3-1} \frac{nd_n}{2^n} + \sum_{n=N_3}^{\infty} \frac{nd_n^*}{2^n} \tag{3.3}$$

and thus $\beta$ has a representation with the same $N_1$ initial terms as $\alpha$. To do this, we observe that

$$\left|\beta - \sum_{n=1}^{N_3-1} \frac{d_n n}{2^n}\right| < \frac{1}{2^{N_3}} + \sum_{n=N_3+1}^{\infty} \frac{n}{2^n} < \frac{N_3+3}{2^{N_3}} < \frac{N_3-1}{2^{N_3-1}}.$$

Thus, Algorithm 1, applied to $\beta - \sum_{n=1}^{N_3-1} d_n n/2^n$, generates no nonzero terms until after the $(N_3-1)$st digit. In particular,

$$\beta - \sum_{n=1}^{N_3-1} \frac{d_n n}{2^n} = \sum_{n=N_3}^{\infty} \frac{d_n^* n}{2^n},$$

and we are done. □

We will show presently that there exist infinitely many (nondiadic) rationals with unique $*$-representations.

We now present a nondeterministic algorithm for constructing noncanonical $*$-binary representations. We shall in fact prove that all $*$-binary representations of $\alpha$, $0 \le \alpha \le 1$, arise as possible outputs of this algorithm. Therefore, this may be modified to be a deterministic algorithm which produces all possible initial segments of $*$-binary representations of $\alpha$.

**Algorithm 3.** *Let*

$$\alpha = \sum_{n=1}^{\infty} \frac{b_n}{2^n}, \qquad b_n = 0, 1,$$

*be a binary representation of* $0 \le \alpha \le 1$. *Let*

$$a_1 := b_1$$

*and, for* $n \geq 1$, *let*

$$d_n := \begin{cases} 0 & \textit{if}\, a_n \leq n-1, \\ 0 \textit{ or } 1 & \textit{if}\, a_n = n \textit{ or } a_n = n+1, \\ 1 & \textit{if}\, a_n \geq n+2 \end{cases}$$

*and*

$$a_{n+1} := 2(a_n - nd_n) + b_n .$$

*Then*

$$\alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n} .$$

*Proof.* As was true for Algorithm 1, equation (2.1) reduces the proof to showing that $a_n/2^n \to 0$ for all possible outputs of the algorithm. This follows from the inequality $0 \leq a_n \leq 2n+1$, which can be proved by an easy induction. □

Suppose at some stage during a run of the algorithm, $a_n = n$ or $a_n = n+1$. We shall refer to setting $d_n = 1$ as the *canonical choice*, since always making this choice reduces Algorithm 3 to Algorithm 1. We shall refer to setting $d_n = 0$ as *cheating*, since this does not correspond to following a greedy algorithm.

As an example, we run the algorithm on $1/4 = .01000\ldots$ three times: canonically, cheating at the first opportunity, and cheating at the first two opportunities. This yields

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $d_n$ | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| $a_n$ | 0 | 1 | 2 | 4 | 0 | 0 | ... |
| $b_n$ | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $d_n$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | ... |
| $a_n$ | 0 | 1 | 2 | 4 | 8 | 6 | 0 | 0 | ... |
| $b_n$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| | | | | $\widehat{\text{cheat}}$ | | | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $d_n$ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | ... |
| $a_n$ | 0 | 1 | 2 | 4 | 8 | 6 | 12 | 10 | 4 | 8 | 16 | 10 | 20 | 14 | 0 | ... |
| $b_n$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| | | | | | $\widehat{\text{cheat}}$ | $\widehat{\text{cheat}}$ | | | | | | | | | | |

Thus,

$$\frac{1}{4} = \frac{4}{2^4},$$

$$\frac{1}{4} = \frac{5}{2^5} + \frac{6}{2^6},$$

$$\frac{1}{4} = \frac{5}{2^5} + \frac{7}{2^7} + \frac{8}{2^8} + \frac{11}{2^{11}} + \frac{13}{2^{13}} + \frac{14}{2^{14}} .$$

Notice that the input to the algorithm is the given binary representation of $\alpha$, not $\alpha$. For diadic rationals, the choice of binary representation can affect the outcome of the algorithm. For example, running the algorithm on $\frac{1}{4} = .0011\ldots$ and cheating at the first opportunity, produces an infinite $*$-binary representation:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | ... |
| 0 | 0 | 1 | 3 | 7 | 5 | 11 | 9 | 19 | 21 | 23 | ... |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ... |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
| | | | | | | | cheat | | | | |

$$\frac{1}{4} = \frac{5}{2^5} + \frac{7}{2^7} + \sum_{n=9}^{\infty} \frac{n}{2^n}.$$

**Proposition 4.** *Suppose* $0 \le \alpha \le 1$.

(a) *If* $\alpha$ *is not a diadic rational, then every* $*$*-binary representation of* $\alpha$ *arises as a possible output of Algorithm* 3.

(b) *If* $\alpha$ *is a diadic rational, and*

$$\alpha = \sum_{n=1}^{\infty} g_n n/2^n, \qquad g_n = 0, 1,$$

*is a* $*$*-binary representation of* $\alpha$*, then either there exists an* $N$ *for which* $g_n = 1$ *for all* $n \ge N$*, or the sequence* $\{g_n\}$ *is a possible output of Algorithm* 3 *run on the terminating binary representations of* $\alpha$.

*Proof.* If $\alpha$ is not a diadic rational, it has a unique binary representation

$$\alpha = \sum_{n=1}^{\infty} b_n/2^n, \qquad b_n = 0, 1.$$

Let

$$\alpha = \sum_{n=1}^{\infty} g_n n/2^n, \qquad g_n = 0, 1,$$

be a given $*$-binary representation of $\alpha$. To simplify equations, we introduce the notation

$$B_n = \sum_{m=n+1}^{\infty} b_m/2^m, \qquad R_n = \sum_{m=n+1}^{\infty} g_m m/2^m.$$

If $\alpha > \frac{1}{2}$, then $a_1 = b_1 = 1$ and cheating is allowed. Cheating causes an output of $d_1 = 0$, while not cheating causes $d_1 = 1$. Thus we may arrange to have $d_1 = g_1$. If, on the other hand, $\alpha < \frac{1}{2}$, then no cheating is possible since $a_1 = b_1 = 0$, and the output is $d_1 = 0$. However, as $\frac{1}{2} > \alpha$, this term cannot occur in the $*$-binary representation, i.e., $g_1 = 0$.

Let us assume that some run of the algorithm has produced $d_k = g_k$ for $k = 1, \dots, n-1$. We shall endeavor to show that, by cheating if allowed and necessary, the algorithm will output $d_n = g_n$.

In the present context, equation (2.1) now reads

$$\alpha = \sum_{m=1}^{n-1} \frac{mg_m}{2^m} + \frac{a_n}{2^n} + B_n$$

or equivalently

$$a_n = 2^n R_{n-1} - 2^n B_n. \tag{3.4}$$

Since $\alpha$ is not a diadic rational, we have only to consider the following three cases:

(i) $R_{n-1} < n/2^n$,

(ii) $n/2^n < R_{n-1} < (n+2)/2^n$,

(iii) $R_{n-1} > (n+2)/2^n$.

In case (i), no cheating is possible because, by (3.4),

$$a_n = 2^n R_{n-1} - 2^n B_n < 2^n R_{n-1} < n.$$

The algorithm must produce $d_n = 0$. Also, $R_{n-1}$ is too small to contain the summand $n/2^n$, so $g_n = 0 = d_n$.

In case (ii), we add the inequalities

$$n < 2^n R_{n-1} < n+2, \qquad -1 < -2^n B_n < 0$$

and obtain, by equation (3.4),

$$n - 1 < a_n < n+2.$$

But $a_n$ is an integer, so $n \le a_n \le n+1$, and cheating is possible. Thus, we may arrange to have the output $d_n$ equal $g_n$.

In case (iii), we have

$$a_n = 2^n R_{n-1} - 2^n B_n > (n+2) - 1$$

and, since $a_n$ is an integer,

$$a_n \ge n+2.$$

No cheating is possible, and the output must be $d_n = 1$. If $g_n$ were to equal zero, then

$$R_{n-1} = R_n \le \sum_{m=n+1}^{\infty} \frac{m}{2^m} = \frac{n+2}{2^n},$$

a contradiction. Therefore, $g_n = 1 = d_n$. This completes the proof of (a).

Let

$$\alpha = \sum_{n=1}^{N} b_n/2^n, \qquad b_n = 0, 1, \ b_N = 1,$$

be the terminating binary representation of the diadic rational $\alpha$, and let

$$\alpha = \sum_{n=1}^{\infty} g_n n/2^n, \qquad g_n = 0, 1,$$

be a given $*$-binary representation.

Notice that

$$0 < 2^n B_n < 1 \quad \text{for } n = 1, \dots, N-1,$$

and since $a_n$ is an integer, equation (3.4) implies that

$$2^n R_{n-1} \notin \mathbf{Z} \quad \text{for } n = 1, \dots, N-1.$$

Therefore, for $n \leq N-1$, one of cases (i), (ii) or (iii) above must hold, and the proof of part (a) works here to show that some run of the algorithm produces $d_n = g_n$ for $n = 1, \dots, N-1$.

We now show that, unless the $g_n$ are eventually all ones, one may prove by induction that $g_n$, $n \geq N$, are possible outputs. Suppose the algorithm has produced $d_k = g_k$ for $k = 1, \dots, n$, for some $n \geq N-1$. At this stage, $B_n = 0$, so

$$R_{n-1} = a_n/2^n.$$

We must now consider four cases:

(i) $R_{n-1} < n/2^n$,

(ii) $R_{n-1} = n/2^n$ or $(n+1)/2^n$,

(iii) $R_{n-1} > (n+2)/2^n$,

(iv) $R_{n-1} = (n+2)/2^n$.

Cases (i), (ii) and (iii) are handled exactly as in the proof of part (a), except that now the required inequalities for $a_n$ are trivial to obtain.

In case (iv), the algorithm is forced to produce $d_n = 1$. If $g_n = 1$, the induction may continue. It is possible to have $g_n = 0$, but then

$$\frac{n+2}{2^n} = R_{n-1} = R_n = \sum_{m=n+1}^{\infty} \frac{g_m m}{2^m} \leq \sum_{m=n+1}^{\infty} \frac{m}{2^m} = \frac{n+2}{2^n},$$

which forces $g_{n+1} = g_{n+2} = \cdots = 1$. □

**Proposition 5.** *Any number of the form*

$$\sum_{k=N}^{\infty} \frac{2k}{2^{2k}} \quad \text{or} \quad \sum_{k=N}^{\infty} \frac{2k-1}{2^{2k-1}}, \qquad N > 2,$$

*has a unique $*$-binary representation* (*as above*). *In particular, the first few of these,* 5/24, 13/288, 1/72... *and* 17/72, 23/288, 29/1152..., *have unique $*$-binary representations.*

*Proof.* We use the identity

$$\sum_{k=N+1}^{\infty} \frac{2k}{2^{2k}} = \frac{2}{3}\left(\frac{N}{2^{2N}}\right) + \frac{8}{9}\left(\frac{1}{2^{2N}}\right) \tag{3.5}$$

to prove inductively that

$$\sum_{k=N}^{\infty} \frac{2k}{2^{2k}}$$

is the unique $*$-binary representation. Since, from (3.5),

$$\sum_{k=N+1}^{\infty} \frac{2k}{2^{2k}} < \frac{2N+1}{2^{2N+1}}, \qquad N > 2,$$

it is not possible to replace an odd $*$-digit by a 1. (This also shows why the initial $*$-digits must be zero.) Since

$$\frac{2N}{2^{2N}} + \sum_{k=N+1}^{\infty} \frac{2k}{2^{2k}} > \sum_{k=2N+1}^{\infty} \frac{k}{2^k}, \qquad N > 2,$$

it is not possible to replace any even digit by a zero. (What we have really done is construct numbers for which $a_n \neq n$ or $n+1$ and used the previous proposition.) The odd case is similar. □

We know, by Corollary 2, that if $\alpha$ is rational and

$$\alpha = \sum_{n=1}^{\infty} \frac{g_n}{2^{g_n}}$$

is a nonterminating $*$-representation, then

$$\limsup_n \left( \frac{g_{n+1} - g_n}{\log_2 g_n} \right) < \infty .$$

We want to show that, given Conjecture 1, this is best possible.

**Proposition 6.** *If Conjecture* 1 *holds, then every diadic rational* $\alpha \in (0, 1)$ *has a representation*

$$\alpha = \sum_{n=1}^{\infty} \frac{g_n}{2^{g_n}},$$

*where*

$$\limsup_n \left( \frac{g_{n+1} - g_n}{\log_2 g_n} \right) \geq 1 .$$

*Proof*. Consider the representation of $(2-\alpha)$ that is constructed by expanding $(2-\alpha)$ by Algorithm 1 and then systematically replacing the final 1 in the (terminating) representation as described in part (b) of Proposition 3. Note that, if a one in the $N$th place is modified in this fashion, it gives rise to a sequence of at least $\log_2 N$ ones following it. This gives a representation of $(2-\alpha)$ with "logarithmically long" sequences of ones. However, if

$$2 - \alpha = \sum_{n=1}^{\infty} \frac{nd_n}{2^n},$$

then

$$\alpha = \sum_{n=1}^{\infty} \frac{n(1-d_n)}{2^n},$$

and this provides the representation of $\alpha$ with logarithmically long sequences of zeros. □

It seems possible that many other rationals have logarithmically large gaps. We computed the first million $*$-binary digits of the canonical representation of $1/3$ and encountered exactly 2 strings of 17 consecutive zeros (starting at 287,658 and 969,239). We have not ruled out the possibility that all rationals have periodic $*$-binary representations, but this seems unlikely. If the canonical $*$-representation of $1/3$ is periodic, then either the period is greater than $1/2$-million or it starts after the $1/2$-millionth digit.

## 4. General bases

Let $c$ be a positive integer $(\geq 2)$. The base $c$ analogue of $*$-binary representations is contained in

**Algorithm 4.** *Let $c$ be an integer $(\geq 2)$. Let*

$$\alpha = \sum_{n=1}^{\infty} \frac{B_n}{2^n}, \qquad B_n = 0, 1, \dots, c-1.$$

*Let*

$$a_1 = B_1 \quad \textit{and} \quad a_{n+1} = c(a_n \bmod n) + B_n.$$

*Then*

$$\alpha = \sum_{n=1}^{\infty} \frac{nD_n}{c^n},$$

*where*

$$D_n = \begin{cases} 0 & \textit{if } a_n < n, \\ 1 & \textit{if } n \leq a_n < 2n, \\ 2 & \textit{if } 2n \leq a_n < 3n, \\ \vdots & \\ c-1 & \textit{if } (c-1)n \leq a_n. \end{cases}$$

Once again, this is the "greedy algorithm". The details are similar to those of §2. The analogue of Conjecture 1 is

**Conjecture 2.** Let $\eta$ be any integer and $c$ be any integer $\geq 2$. Let

$$a_m := \eta \quad \text{and} \quad a_{n+1} := c(a_n \bmod n), \qquad n = m, m+1, \dots$$

(where $(a \bmod n)$ is chosen in $[0, n-1]$). Then the above iteration always terminates at zero.

**Corollary 3.** *Conjecture 2 implies that every c-adic rational has a terminating representation as in Algorithm 4.*

The analogue of Proposition 1 is

**Proposition 7.** *For*

$$m = \frac{c^{M+2} - c}{c-1} - M, \qquad M = 1, 2, \dots,$$

*we have*

$$\frac{m-1}{c^{m-1}}=\sum_{k=m}^{m+M}\frac{(c-1)k}{c^k}. \tag{4.1}$$

This can be derived directly or by a generating function argument like that of §2. Proposition 3 has its obvious analogue. As before, this shows that $*$-representations base $c$ are not unique and that there exist numbers with uncountably many different representations. The limited numerical evidence for Conjecture 2 is presented in the next section.

## 5. Matters numerical

**Algorithm 5.** *Let* $c \geq 2$ *and* $M$ *be positive integers, and let*

$$a_{n+1} := c(a_n \bmod n),$$

*where the initial value is*

$$a_m := M.$$

*Here as before,* $(a_n \bmod n)$ *is the principal representation in* $[0, n-1]$.

We will denote this by $\mathrm{ALGO}_c(m, M)$. We say that $\mathrm{ALGO}_c(m, M)$ terminates if $a_h = 0$ for some $h \geq m$, and we say that $\mathrm{ALGO}_c(m, M)$ terminates at $H$ if $H$ is the smallest such $h$. The global termination function $T_c$ is defined so that $T_c(m)$ is the smallest integer (if it exists) so that $\mathrm{ALGO}_c(m, M)$ terminates for all $M$. Note that $T_c(m)$ is a nondecreasing function of $m$. The conjectures then become

**Conjecture 3.** For $c, m \geq 2$, $T_c(m)$ is finite.

This conjecture seems hard. Though not directly related, it has a similar feel to the $3x+1$ conjecture [5].

Some suggestive numbers follow.

$\underline{c=2}$

$$T_2(1)=2$$
$$T_2(2)=5$$
$$T_2(3)=9$$
$$T_2(4)=T_2(5)=15$$
$$T_2(6)=T_2(7)=25$$
$$T_2(8)=T_2(9)=33$$
$$T_2(10)\cdots T_2(53)=393$$
$$T_2(54)\cdots T_2(1000)=12,231$$

<u>$c = 3$</u>

$$\begin{aligned}
&T_3(1) = 2\\
&T_3(2) = 4\\
&T_3(3) \cdots T_3(5) = 10\\
&T_3(6) \cdots T_3(14) = 31\\
&T_3(15) \cdots T_3(20) = 43\\
&T_3(21) \cdots T_3(29) = 121\\
&T_3(30) \cdots T_3(41) = 424\\
&T_3(42) \cdots T_3(100) = 853
\end{aligned}$$

<u>$c = 10$</u>

$$\begin{aligned}
&T_{10}(1) = 2\\
&T_{10}(2) \cdots T_{10}(4) = 6\\
&T_{10}(5) \cdots T_{10}(9) = 11\\
&T_{10}(10) \cdots T_{10}(19) = 26\\
&T_{10}(20) \cdots T_{10}(29) = 51\\
&T_{10}(30) \cdots T_{10}(69) = 106\\
&T_{10}(80) \cdots T_{10}(74) = 111\\
&T_{10}(75) \cdots T_{10}(79) = 113\\
&T_{10}(80) \cdots T_{10}(150) = 261\,.
\end{aligned}$$

We collect some of this and some additional computational experience in the following proposition.

**Proposition 8.** *For $c = 3, 4, 5$ and $10$ and $m \le 100$, $\mathrm{ALGO}_c(m, M)$ terminates for all $M$.*

*For $c = 2$ and $m \le 1000$, $\mathrm{ALGO}_c(m, M)$ terminates for all $M$.*

Algorithm 3 (with Proposition 4) gives a very satisfactory algorithm for computing minimum length $*$-representations of diadic-rationals. We illustrate this on the question of minimum length rewritings of $(n-1)/2^{n-1}$ using the fact, once again, that

$$\frac{n-1}{2^{n-1}} = \frac{n}{2^n} + \frac{n-2}{2^n} \tag{5.1}$$

and considering minimum length rewritings of $(n-2)/2^n$. Note that, if Algorithm 1 applied to $(n-2)/2^n$ terminates at $N$ without generating an $a_n$, $n < N$, with $a_n = n + 1$, then this is, in fact, the minimal representation. If at some point, $a_n = n + 1$, then we branch as in Algorithm 3 by setting $d_n = 0$ instead of 1, and continue. In practice, very few branches are required until we have exceeded $N$. The following table collects some of the numerical experience. It presents the lengths (i.e., numbers of terms and termination term) for minimal representations of $(n-2)/2^n$ for various $n$. Often, though not always, the greedy algorithm provides this representation. Notice that, with (5.1), this provides minimum length solutions for the Diophantine equation (1.1). (The number of terms is one more than that in the table while the largest term is one less than the termination term.) It is perhaps surprising that such an easy algorithm exists for generating minimal representations.

$*$-binary expansions of $(n-2)/2^n$

| $n$ | Algorithm 2 terminates at | minimum rep. terminates | # of branches required |
|---|---|---|---|
| 15 | 393 (186 TERMS) | 393(180 TERMS) | 2 |
| 16 | 23 (4 TERMS) | ← | 0 |
| 17 | 393 (184 TERMS) | 47(16 TERMS) | 1 |
| 18 | 33 (7 TERMS) | ← | 0 |
| 19 | 33 (6 TERMS) | ← | 0 |
| 20 | 33 (7 TERMS) | ← | 0 |
| 122 | 12231 (6065 TERMS) | 321(101 TERMS) | 1 |
| 5000 | 7183 (1065 TERMS) | ← | 1 |
| 5001 | 12231 (3612 TERMS) | ← | 2 |
| 5002 | 5601 (286 TERMS) | ← | 1 |
| 5003 | 6273 (642 TERMS) | ← | 0 |
| 5004 | 12231 (3620 TERMS) | ← | 2 |
| 5005 | 12231 (3590 TERMS) | ← | 0 |

## Bibliography

1. P. Erdös, *Some problems and results on the irrationality of the sum of infinite series*, J. Math. Sci. **10** (1975), 1–7.
2. ——, *Sur l'irrationalité d'une certaine série*, C. R. Acad. Sci. Paris Ser. I Math. **292** (1981), 65–768.
3. P. Erdös and R. L. Graham, *Old and new results in combinatorial number theory*, Monograph No. 28 de d'Enseignement Mathématique, Genève, 1980.
4. D. Knuth, *The art of computer programming, vol.* 2: *Seminumerical algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1981.
5. J. C. Lagarias, *The* $3x+1$ *problem and its generalization*, Amer. Math. Monthly **92** (1985), 3–23.

Department of Mathematics, Statistics and Computing, Dalhousie University, Halifax, Nova Scotia B3H 3J5, Canada