

The Recursive Stopping Time Structure of the $3x + 1$ Function

Mike Winkler

Fakultät für Mathematik
Ruhr-Universität Bochum, Germany
mike.winkler@ruhr-uni-bochum.de
mikewinkler.co.nf

March 4, 2018

Abstract

The $3x + 1$ problem concerns iteration of the map $T : \mathbb{Z} \rightarrow \mathbb{Z}$ given by

$$T(x) = \begin{cases} \frac{x}{2} & \text{if } x \equiv 0 \pmod{2}, \\ \frac{3x+1}{2} & \text{if } x \equiv 1 \pmod{2}. \end{cases}$$

The $3x + 1$ Conjecture states that every $x \geq 1$ has some iterate $T^s(x) = 1$. The least $s \in \mathbb{N}$ such that $T^s(x) < x$ is called the stopping time of x . It is shown that the congruence classes $(\text{mod } 2^k)$ of the integers having finite stopping time are given by a recursive algorithm producing a directed rooted tree.

1 Introduction

The $3x + 1$ function is defined as a function $T : \mathbb{Z} \rightarrow \mathbb{Z}$ given by

$$T(x) = \begin{cases} \frac{x}{2} & \text{if } x \equiv 0 \pmod{2}, \\ \frac{3x+1}{2} & \text{if } x \equiv 1 \pmod{2}. \end{cases}$$

Let $T^0(x) = x$ and $T^s(x) = T(T^{s-1}(x))$ for $s \in \mathbb{N}$. Then we get for each $x \in \mathbb{N}$ a sequence $C(x) = (T^s(x))_{s=0}^\infty$. For example, the starting value $x = 11$ generates the sequence

$$C(11) = (11, 17, 26, 13, 20, 10, 5, 8, 4, 2, 1, 2, 1, 2, 1, \dots).$$

Any $C(x)$ can only assume two possible forms. Either it falls into a cycle or it grows to infinity. The $3x + 1$ Conjecture states that every $C(x)$ enters the trivial cycle $(2, 1, 2, 1, \dots)$.

2 The stopping time $\sigma(x)$ and the congruences (mod 2^k)

The $3x + 1$ Conjecture still holds if for each $x \in \mathbb{N}, x > 1$, there exists $s \in \mathbb{N}$ such that $T^s(x) < x$. The least $s \in \mathbb{N}$ such that $T^s(x) < x$ is called the stopping time of x , which we will denote by $\sigma(x)$.

Definition 1.

- (a) We define $C^a(x)$ to be a finite subsequence of $C(x)$, given by $C^a(x) = (T^s(x))_{s=0}^a$ for $a \in \mathbb{N}, a \geq 1$.
- (b) Let $n \in \mathbb{N}$ denote the number of odd terms in $C^{\sigma(x)-1}(x)$, where $T^0(x)$ is not counted.
- (c) We define σ_n to be $\lfloor 1 + (n + 1) \cdot \log_2 3 \rfloor$ for all $n \in \mathbb{N}$.

Theorem 2. *For each $n \geq 1$ there exists a finite set of congruence classes (mod 2^{σ_n}) with the property that all integers $x > 1$ of one of these congruence classes have finite stopping time $\sigma(x) = \sigma_n$. For each $n \geq 3$ the number of stopping time congruences (mod 2^{σ_n}) as listed in [A100982](#), and the number of remaining congruences (mod 2^k) as listed in [A076227](#) can be generated recursively from the two starting values 0 and 1.*

Proof. The proof will be divided into two parts. In the first part we prove that $\sigma(x) = \sigma_n$ for all $n \geq 1, x > 1$.

For the congruences (mod 2^k) in each case, k steps can be calculated. As long as a factor 2 is included only the congruence decides whether the next number is even or odd and this step can be done. If the powers of 2 are used up, they are replaced by a certain number of factors 3, which is less than or equal to the initial k , depending on how many $\frac{3x+1}{2}$ and $\frac{1}{2}$ steps have been occurred.

Let $r, q \in \mathbb{N}$, then in general $r \pmod{2^k}$ leads to $q \pmod{3^n}$ with $k \geq n$. We have $k = n$ exactly for $r = 2^k - 1$, which is also the deeper reason for the fact that more and more congruences remain, specifically the congruences of the form $2^k - 1$. If $2^k > 3^n$ then the stopping time is reached.

Let $\kappa(n)$ denote the highest k such that $2^k < 3^n$ as listed in [A056576](#), we have $\kappa(n) = \lfloor n \cdot \log_2 3 \rfloor$. Hence $\kappa(n+1) + 1 = \lfloor 1 + (n+1) \cdot \log_2 3 \rfloor$, which gives the stopping time $\sigma(x)$ for each $n \geq 1$. Because, according to Definition 1(b), $\sigma(x)$ is given exactly by $k + 1$ of the highest k such that $2^k < 3^{n+1}$ or $\kappa(n+1) + 1$. By Definition 1(c), it follows directly from $\kappa(n+1) + 1$ that $\sigma(x) = \sigma_n$ for each $n \in \mathbb{N}$, which completes the first part of the proof. It is not hard to verify that

$$\begin{aligned} \sigma(x) = \sigma_1 = 4 & \quad \text{if } x \equiv 3 \pmod{16}, \\ \sigma(x) = \sigma_2 = 5 & \quad \text{if } x \equiv 11, 23 \pmod{32}, \\ \sigma(x) = \sigma_3 = 7 & \quad \text{if } x \equiv 7, 15, 59 \pmod{128}, \\ \sigma(x) = \sigma_4 = 8 & \quad \text{if } x \equiv 39, 79, 95, 123, 175, 199, 219 \pmod{256}, \end{aligned}$$

and so forth. Note that $\sigma(x) = 1$ if $x \equiv 0 \pmod{2}$ and $\sigma(x) = 2$ if $x \equiv 1 \pmod{4}$.

The congruences $\pmod{2^k}$ can be evolved according to a binary tree. If we pass from a specific value k to the value $k + 1$, always *two* new values arise from the remaining candidates, so $r \pmod{2^k}$ became $r \pmod{2^{k+1}}$ or $(r + 2^k) \pmod{2^{k+1}}$. For one of them the result in the k -th step is even, for the other it is odd. Therefore one continues with the $\frac{3x+1}{2}$ step (power of 3 increases by one), the other continues with the $\frac{1}{2}$ step (power of 3 is retained).

Now we consider the number of congruences that lead to a specific power of 3. Let $\mathcal{R}(k, n)$ be the number of congruences $\pmod{2^k}$ which meet the condition $2^k < 3^n$ and lead to a congruence $\pmod{3^n}$. Each congruence $\pmod{2^{k+1}}$ comes from a congruence $\pmod{2^k}$, and either n is increased or n is retained, depending on the type of step performed. We thus get $\mathcal{R}(k + 1, n) = \mathcal{R}(k, n) + \mathcal{R}(k, n - 1)$ with the starting condition $\mathcal{R}(2, 2) = 1$ and $\mathcal{R}(2, 1) = 0$. Because $3 \pmod{2^2}$ is the only non-trivial starting value and leads to $8 \pmod{3^2}$. Hence the number of congruences $\pmod{2^k}$ can be calculated recursively in the fashion of a Pascal's triangle, whose left side is cut off by the stopping time condition $2^k > 3^n$. By the definition of $\kappa(n)$ it follows that for each $n \geq 2$ the last value $\mathcal{R}(k + 1, n)$ is given by $k + 1 = \kappa(n)$.

With our results so far we are able to develop an algorithm which generates [A100982](#) and [A076227](#) from the two starting values 0 and 1 for each $n \geq 3$. Appendix 9.1.1 and 9.1.2 give a program for the algorithm of Theorem 2, which completes the second part of the proof. \square

Now let $z(n)$ denote the number of congruences $\pmod{2^{\sigma_n}}$ as listed in [A100982](#) and let $w(k)$ denote the number of surviving congruences $\pmod{2^k}$ as listed in [A076227](#).

Table 1 illustrates the algorithm of Theorem 2. Note that no entry is equal to the value zero. The possible stopping times σ_n are listed in [A020914](#). The congruences $\pmod{2^{\sigma_n}}$ are listed in [A177789](#).

| | | | | | | | | | | | | | |
|---------------|---|---|---|---|----|----|----------|----------|----------|----------|----------|----------|----------|
| $\kappa(n) =$ | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 12 | 14 | 15 | 17 | ... | |
| $n =$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | ... | |
| $k =$ | | | | | | | | | | | | | $w(k) =$ |
| 2 | | 1 | | | | | | | | | | | 1 |
| 3 | | 1 | 1 | | | | | | | | | | 2 |
| 4 | | | 2 | 1 | | | | | | | | | 3 |
| 5 | | | | 3 | 1 | | | | | | | | 4 |
| 6 | | | | 3 | 4 | 1 | | | | | | | 8 |
| 7 | | | | | 7 | 5 | 1 | | | | | | 13 |
| 8 | | | | | | 12 | 6 | 1 | | | | | 19 |
| 9 | | | | | | 12 | 18 | 7 | 1 | | | | 38 |
| 10 | | | | | | | 30 | 25 | 8 | 1 | | | 64 |
| 11 | | | | | | | 30 | 55 | 33 | 9 | 1 | | 128 |
| \vdots | | | | | | | \vdots | \vdots | \vdots | \vdots | \vdots | \ddots | \vdots |
| $z(n) =$ | | 2 | 3 | 7 | 12 | 30 | 85 | 173 | 476 | 961 | 2652 | ... | |

Table 1: Triangle expansion of the number of congruences $\pmod{2^k}$.

Table 1 gives $z(n)$ for each $n \geq 2$ by the sum of the values in each column n . With

$\kappa(n) = \lfloor n \cdot \log_2 3 \rfloor$ we have $z(n) = \sum_{k=n}^{\kappa(n)} \mathcal{R}(k, n)$. For example, for $n = 4$ we see that exactly $1 + 3 + 3 = 7$ congruences $(\text{mod } 2^{\sigma_n})$ with stopping time $\sigma_4 = 8$ exist.

Table 1 gives $w(k)$ for each $k \geq 2$ by the sum of the values in each row k by $w(k) = \sum_{n=\lfloor 1+k \cdot \log_3 2 \rfloor}^k \mathcal{R}(k, n)$. For example, for $k = 6$ we see that relating to 2^6 exactly $3+4+1 = 8$ remaining congruences exist, where three lead to 3^4 , four lead to 3^5 and one leads to 3^6 . The values of $\lfloor 1 + k \cdot \log_3 2 \rfloor$ are listed in [A020915](#).

3 Subsequences $C^{\kappa(n)}(x)$ and a stopping time term formula

If an odd starting value $x > 1$ has stopping time $\sigma(x) = \sigma_n$, then it is shown in the proof of Theorem 2 that for each $n \geq 1$ the subsequence $C^{\kappa(n)}(x)$ represents *sufficiently* the stopping time of x . By definition, every $C^{\kappa(n)}(x)$ consists of $n + 1$ odd terms. Hence all terms $T^s(x)$ with $\kappa(n) < s < \sigma(x)$ are even.

If the succession of the even and odd terms in $C^{\kappa(n)}(x)$ is known, it is not hard to develop a term formula for the exact value of $T^{\sigma(x)}(x)$ with $\sigma(x) = \sigma_n$.

Theorem 3. *Let $C^{\kappa(n)}(x)$ consisting of $n + 1$ odd terms be given. Let $\alpha_i \in \mathbb{N}, \alpha_i \geq 0, i = 1, \dots, n + 1$. Now let $\alpha_i = s$, if and only if $T^s(x)$ in $C^{\kappa(n)}(x)$ is odd. Then we have*

$$T^{\sigma_n}(x) = \frac{3^{n+1}}{2^{\sigma_n}} \cdot x + \sum_{i=1}^{n+1} \frac{3^{n+1-i} 2^{\alpha_i}}{2^{\sigma_n}} < x. \quad (1)$$

For a proof of Theorem 3 we refer the reader to LAGARIAS[2]. A very similar term formula is also given by GARNER[1]. For example, for $n = 3$ we have $\sigma_3 = 7$. For $x = 59$ we get by (1)

$$T^7(59) = \frac{3^4}{2^7} \cdot 59 + \frac{3^3 2^0 + 3^2 2^1 + 3^1 2^3 + 3^0 2^4}{2^7} = 38 < 59.$$

Because, for $n = 3$ we have $\kappa(3) = 4$. $C^4(59) = (59, 89, 134, 67, 101)$ consists of $3 + 1 = 4$ odd terms given by 59, 89, 67, 101. The powers of two α_i we get as follows: $T^0 = 59$ is odd, so $\alpha_1 = 0$. $T^1 = 89$ is odd, so $\alpha_2 = 1$. $T^2 = 134$ is even. $T^3 = 67$ is odd, so $\alpha_3 = 3$. $T^4 = 101$ is odd, so $\alpha_4 = 4$. A comparison with $C^7(59) = (59, 89, 134, 67, 101, 152, 76, 38)$ confirms the solution $T^7(59) = 38$.

4 Parity vectors $v_n(x)$ and parity vector sets $\mathbb{V}(n)$

To simplify the distribution of the even and odd terms in $C^{\kappa(n)}(x)$ we define a *zero-one* sequence $v_n(x)$ by

$$v_n(x) = C^{\kappa(n)}(x) \quad \text{with} \quad T^s(x) = \begin{cases} 0 & \text{if } T^s(x) \equiv 0 \pmod{2}, \\ 1 & \text{if } T^s(x) \equiv 1 \pmod{2}, \end{cases} \quad (2)$$

which we will denote as the *parity vector* of x . In fact, $v_n(x)$ is a vector of $\kappa(n) + 1$ elements, where "0" represents an even term and "1" represents an odd term in $C^{\kappa(n)}(x)$.

For each $n \geq 1$ we define a *parity vector set* $\mathbb{V}(n)$ as the set of $z(n) \geq 1$ parity vectors $v_n(x)$, where $\sigma(x) = \sigma_n$ for each parity vector of the set. For Example, for $n = 3$ we have $\kappa(3) = 4$. For $x = 7, x = 15$ and $x = 59$ the subsequences $C^{\kappa(n)}(x)$ and their appropriate parity vectors $v_n(x)$ are given by

$$\begin{aligned} C^4(7) &= (7, 11, 17, 26, 13) & \text{and} & & v_3(7) &= (1, 1, 1, 0, 1), \\ C^4(15) &= (15, 23, 35, 53, 80) & \text{and} & & v_3(15) &= (1, 1, 1, 1, 0), \\ C^4(59) &= (59, 89, 134, 67, 101) & \text{and} & & v_3(59) &= (1, 1, 0, 1, 1). \end{aligned}$$

The parity vector set $\mathbb{V}(3)$ consists of these three parity vectors, because for $n = 3$ we have $\sigma(x) = \sigma_3 = 7$ only for the $z(3) = 3$ congruence classes $7, 15, 59 \pmod{2^7}$. We thus get

$$\mathbb{V}(3) := \left\{ \begin{array}{l} (1, 1, 0, 1, 1) \\ (1, 1, 1, 0, 1) \\ (1, 1, 1, 1, 0) \end{array} \right\}.$$

Remark 4. According to Theorem 3, for each $n \geq 1$ there exists for each parity vector of $\mathbb{V}(n)$ a *unique* Diophantine equation from (1) given by

$$y = \frac{3^{n+1}}{2^{\sigma_n}} \cdot x + \sum_{i=1}^{n+1} \frac{3^{n+1-i} 2^{\alpha_i}}{2^{\sigma_n}}, \quad (3)$$

whose *only* positive integer solutions (x, y) are for x the congruence classes $\pmod{2^{\sigma_n}}$. Note that the positive integer solutions $x < 2^{\sigma_n}$ from (3) for each parity vector of $\mathbb{V}(n)$ are equal to the congruences as listed in [A177789](#).

5 Generating the parity vectors of $\mathbb{V}(n)$

With our results so far we are able to build the parity vectors of $\mathbb{V}(n)$ for each $n \geq 2$.

Theorem 5. For each $n \geq 2$ the parity vectors of $\mathbb{V}(n)$ can be generated algorithmically producing a directed rooted tree.

Proof. Let $d(n)$ denote the number of 2^k between 3^{n-1} and 3^n as listed in [A022921](#). Then with $\kappa(n) = \lfloor n \cdot \log_2 3 \rfloor$ we have

$$d(n) = \kappa(n) - \kappa(n-1) = \lfloor n \cdot \log_2 3 \rfloor - \lfloor (n-1) \cdot \log_2 3 \rfloor. \quad (4)$$

From (4) we have $d(n) \in \{1, 2\}$ for each $n \geq 2$. From Section 3 and (2) it follows that for each $n \geq 2$ each parity vector of $\mathbb{V}(n)$ consists of $(n+1)$ ones and $(\kappa(n) - n)$ zeros. Now let the root of the tree be given for $n = 1$ by $v_1(x) = (1, 1)$, then the parity vectors of $\mathbb{V}(n)$ for each $n \geq 2$ are given by a *three-step-algorithm* as follows.

1. Build one new vector of $\mathbb{V}(n)$ by adding the vector of $\mathbb{V}(n-1)$ on the right by "1" if $d(n) = 1$, and by "0, 1" if $d(n) = 2$.
2. Build $j \geq 1$ new vectors of $\mathbb{V}(n)$, if the new vector in step 1 contains $j \geq 1$ zeros in direct progression from the right-sided penultimate position to the left. In this case the last right-sided one will change its position with each of its $j \geq 1$ left-sided zeros in direct progression, only one change for each new vector from the right to the left.
3. Repeat step 1 and step 2 until the new vector of $\mathbb{V}(n)$ begins with $(n+1)$ ones followed by $(\kappa(n) - n)$ zeros.

Appendix 9.1.3 and 9.1.4 give a program for the algorithm of Theorem 5, which completes the proof. \square

The above algorithm produces a directed rooted tree with two different directions, a horizontal and vertical (cf. Figure 3). This construction principle gives the tree a triangular form which extends ever more downwards with each column. Figure 4 displays the beginning of the tree up to $\mathbb{V}(4)$.

Now let $h \geq 2$ denote the number of the first ones in direct progression in a parity vector and let $\mathcal{P}(h, n)$ denote the number of such parity vectors with same h for each $n \geq 1$. Table 2 gives the first values for $\mathcal{P}(h, n)$ as generated by the algorithm of Theorem 5.

| $n =$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | ... |
|----------|---|---|---|---|----|----|----|-----|-----|-----|------|----------|
| $h =$ | | | | | | | | | | | | |
| 2 | 1 | 1 | 1 | 2 | 3 | 7 | 19 | 37 | 99 | 194 | 525 | ... |
| 3 | | 1 | 1 | 2 | 3 | 7 | 19 | 37 | 99 | 194 | 525 | ... |
| 4 | | | 1 | 2 | 3 | 7 | 19 | 37 | 99 | 194 | 525 | ... |
| 5 | | | | 1 | 2 | 5 | 14 | 28 | 76 | 151 | 412 | ... |
| 6 | | | | | 1 | 3 | 9 | 19 | 53 | 108 | 299 | ... |
| 7 | | | | | | 1 | 4 | 10 | 30 | 65 | 186 | ... |
| 8 | | | | | | | 1 | 4 | 14 | 34 | 103 | ... |
| 9 | | | | | | | | 1 | 5 | 15 | 50 | ... |
| 10 | | | | | | | | | 1 | 5 | 20 | ... |
| 11 | | | | | | | | | | 1 | 6 | ... |
| 12 | | | | | | | | | | | 1 | ... |
| \vdots | | | | | | | | | | | | \ddots |
| $z(n) =$ | 1 | 2 | 3 | 7 | 12 | 30 | 85 | 173 | 476 | 961 | 2652 | ... |

Table 2: Triangle expansion of the number of parity vectors.

The triangle structure in Table 2 follows directly from the construction principle and gives an idea of the size of the tree. Note the peculiarity here that the first three rows are identical for each $n \geq 3$.

As well as Table 1, Table 2 gives $z(n)$ for each $n \geq 1$ by the sum of the values in each column n by $z(n) = \sum_{h=2}^{n+1} \mathcal{P}(h, n)$. For example, for $n = 4$ we see that exactly $2 + 2 + 2 + 1 = 7$ congruences classes (mod 2^{σ_n}) with stopping time $\sigma_4 = 8$ exist (cf. Figure 4). The "1" entries at the lower end of each column refer to the "one" parity vector beginning with $(n + 1)$ ones followed by $(\kappa(n) - n)$ zeros as mentioned in the third step of the algorithm of Theorem 5. The values from Table 2 or $\mathcal{P}(h, n)$ can also be generated recursively. Appendix 9.1.5 gives a program for this algorithm.

6 The order of the generated parity vectors

The way the algorithm of Theorem 5 generates the parity vectors represents the *exact* order as given by all *permutations in lexicographic ordering*¹ of a *zero-one word*² with $(\kappa(n) - n)$ zeros and $(n - 1)$ ones, given by

$$\underbrace{(0, \dots, 0)}_{\kappa(n) - n}, \underbrace{(1, \dots, 1)}_{n - 1}, \tag{5}$$

where the first two ones must be added on the left side. Let $L(n)$ denote for each $n \geq 1$ the number of all permutations in lexicographic ordering of a zero-one word (5), then we have $L(n) = \frac{(\kappa(n)-1)!}{(\kappa(n)-n)! \cdot (n-1)!}$, which generates the sequence 1, 2, 3, 10, 15, 56, 210, 330, ..., listed in [A293308](#).

In regard to Theorem 3 and Chapter 4, by interpreting the zero-one words (5) with the first two added ones on the left as a simplification for the even and odd terms in $C^{\kappa(n)}(x)$, as same as the parity vectors, *only* for the $L(n) \geq 1$ zero-one words the conditions of Theorem 3 and (3) are complied. Note there are *no other* possibilities for an integer solution (x, y) , but not for all of them we have $\sigma(x) = \sigma_n$. This applies only to the zero-one words which are identical to the parity vectors of $\mathbb{V}(n)$. For all others we have $\sigma(x) < \sigma_n$. For Example, for $n = 5$ we have $\kappa(5) = 7$ and $L(5) = 15$. The left side in Table 3 gives the 15 permutations in lexicographic ordering of the zero-one word (5). The right side gives these zero-one words with the first two added ones and their integer solution (x, y) for (3).

¹As given by the algorithm in Appendix 9.1.6.

²We use *word* instead of *vector* to exclude confusion regarding to Chapter 4.

| | | | |
|----|--------------------|--------------------------|------------|
| 1 | (0, 0, 1, 1, 1, 1) | (1, 1, 0, 0, 1, 1, 1, 1) | (595, 425) |
| 2 | (0, 1, 0, 1, 1, 1) | (1, 1, 0, 1, 0, 1, 1, 1) | (747, 533) |
| 3 | (0, 1, 1, 0, 1, 1) | (1, 1, 0, 1, 1, 0, 1, 1) | (507, 362) |
| 4 | (0, 1, 1, 1, 0, 1) | (1, 1, 0, 1, 1, 1, 0, 1) | (347, 248) |
| 5 | (0, 1, 1, 1, 1, 0) | (1, 1, 0, 1, 1, 1, 1, 0) | (923, 658) |
| 6 | (1, 0, 0, 1, 1, 1) | (1, 1, 1, 0, 0, 1, 1, 1) | (823, 587) |
| 7 | (1, 0, 1, 0, 1, 1) | (1, 1, 1, 0, 1, 0, 1, 1) | (583, 416) |
| 8 | (1, 0, 1, 1, 0, 1) | (1, 1, 1, 0, 1, 1, 0, 1) | (423, 302) |
| 9 | (1, 0, 1, 1, 1, 0) | (1, 1, 1, 0, 1, 1, 1, 0) | (999, 712) |
| 10 | (1, 1, 0, 0, 1, 1) | (1, 1, 1, 1, 0, 0, 1, 1) | (975, 695) |
| 11 | (1, 1, 0, 1, 0, 1) | (1, 1, 1, 1, 0, 1, 0, 1) | (815, 581) |
| 12 | (1, 1, 0, 1, 1, 0) | (1, 1, 1, 1, 0, 1, 1, 0) | (367, 262) |
| 13 | (1, 1, 1, 0, 0, 1) | (1, 1, 1, 1, 1, 0, 0, 1) | (735, 524) |
| 14 | (1, 1, 1, 0, 1, 0) | (1, 1, 1, 1, 1, 0, 1, 0) | (287, 205) |
| 15 | (1, 1, 1, 1, 0, 0) | (1, 1, 1, 1, 1, 1, 0, 0) | (575, 410) |

Table 3: The 15 zero-one words for $n = 5$.

Note that the right sided zero-one words (or parity vectors) 1, 2 and 6 are not in $\mathbb{V}(5)$, but their order is exactly the same as generated by the algorithm of Theorem 5. (cf. Appendix 9.2).

7 The Diophantine equations and their integer solutions

As mentioned in Chapter 4, the algorithm of Theorem 5 generates an infinite set of unique Diophantine equations (3) whose only positive integer solutions x gives the stopping time congruence classes (mod 2^{σ_n}). Therefore the $3x + 1$ Conjecture is also a *Diophantine equation problem*, because the conjecture holds, if the set of congruence classes (mod 2^{σ_n}) is equal to the set of congruences $3 \pmod{4}$.

Remark 6. The $x < 2^{\sigma_n}$ of the integer solution (x, y) in (3) for a parity vector of $\mathbb{V}(n)$ we will denote briefly as "the solution x for a parity vector".

In regard to (3) there is a direct connectedness between the elements in direct progression in a parity vector and its solution x . Therefore the algorithm of Theorem 5 allows us to make accurate statements about the solutions x without solving the Diophantine equations explicitly. The following Corollaries are precise implications from Theorem 3, Remark 4 and Theorem 5. We will use the symbol \boxplus to denote the end of a Corollary.

Corollary 7. Regarding to the first $h \geq 2$ ones in direct progression in a parity vector, for each $n \geq 1$ for the solution x of a parity vector we have

$$x \equiv (2^h - 1) \pmod{2^{h+1}}. \quad (6)$$

\boxplus

From now on we need an *individual identification* for each parity vector and its solution x . Let $p \in \mathbb{N}$, $p = 1, \dots, \mathcal{P}(h, n)$, denote the *enumeration value* for the order of the parity vectors with same $h \geq 2$ as generated by the algorithm of Theorem 5. Then for each $n \geq 1$ the individual identification for a parity vector $v_n(x)$ and its solution x we will denote by

$$v_{n,h,p} \quad \text{with} \quad x_{n,h,p}. \quad (7)$$

To make the equations easier to read we will only write the indexes which change. For example, if n and h are fixed, we only write v_p and x_p . Let $v_{n-1,p'}$ denote the *predecessor-parity vector* of $v_{n,p}$ in regard to the first step of the algorithm of Theorem 5.

Corollary 8. Regarding to the *first step* of the algorithm of Theorem 5, for each $n \geq 2$, $h \geq 2$, for the solution x of a parity vector $v_{n,p}$ which last element is "1" we have

$$x_{n,p} \equiv x_{n-1,p'} \pmod{2^{\kappa(n)}}, \quad (8)$$

and $x_{n,p}$ is explicit given with $\lambda \in \{1, 3, 5, 7\}$ by the recurrence relation

$$x_{n,p} = x_{n-1,p'} + \lambda \cdot 2^{\kappa(n)} \quad \text{if} \quad x_{n-1,p'} + \lambda \cdot 2^{\kappa(n)} < 2^{\sigma_n}, \quad (9)$$

or

$$x_{n,p} = x_{n-1,p'} + \lambda \cdot 2^{\kappa(n)} - 2^{\sigma_n} \quad \text{if} \quad x_{n-1,p'} + \lambda \cdot 2^{\kappa(n)} > 2^{\sigma_n}. \quad (10)$$

Note: There exist only four possibilities for $x_{n,p}$ to solve (3) in \mathbb{N} . And if and only if $x > 2^{\sigma_n}$ we have $x_n = x - 2^{\sigma_n}$. \boxplus

Appendix 9.1.7 gives a program for the recurrence relation of Corollary 8.

Corollary 9. Regarding to the *second step* of the algorithm of Theorem 5, for each $n \geq 2$, $2 \leq h \leq n$, for the solution x of a parity vector v_p which last element is "0" we have

$$x_p \equiv x_{p-1} \pmod{2^{\kappa(n)-j}}, \quad (11)$$

where $j \geq 1$ is the number of the last *zeros* in direct progression in v_p .

For each $n \equiv 1 \pmod{2}$ we have

$$x_p = x_{p-1} + \delta \cdot 2^{\kappa(n)-j} \quad \text{with} \quad \delta = 1 \pm 8b, \quad b \in \mathbb{N}. \quad (12)$$

For each $n \equiv 0 \pmod{2}$ we have

$$x_p = x_{p-1} + \delta \cdot 2^{\kappa(n)-j} \quad \text{with} \quad \delta = 3 \pm 8b, \quad b \in \mathbb{N}. \quad (13)$$

For each $2 \leq n \leq 8$, $2 \leq h \leq n$, the solution x_p is explicit given as follows.

For each $n \equiv 1 \pmod{2}$ we have

$$x_p = x_{p-1} + 2^{\kappa(n)-j} + (2 - d(n)) \cdot 2^{\kappa(n)-j+3} \quad (14)$$

if the right side of (14) $< 2^{\sigma_n}$, or

$$x_p = x_{p-1} + 2^{\kappa(n)-j} + (2 - d(n)) \cdot 2^{\kappa(n)-j+3} - 2^{\sigma_n} \quad (15)$$

if the right side of (14) $> 2^{\sigma_n}$.

For each $n \equiv 0 \pmod{2}$ we have

$$x_p = x_{p-1} + 2^{\kappa(n)-j} + 2^{\kappa(n)-j+1} + (2 - d(n)) \cdot 2^{\kappa(n)-j+3} \quad (16)$$

if the right side of (16) $< 2^{\sigma_n}$, or

$$x_p = x_{p-1} + 2^{\kappa(n)-j} + 2^{\kappa(n)-j+1} + (2 - d(n)) \cdot 2^{\kappa(n)-j+3} - 2^{\sigma_n} \quad (17)$$

if the right side of (16) $> 2^{\sigma_n}$.

The rules from equations (14)–(17) also work for each $n \geq 9$ with $j = 1$ for all $d(n) = 2$ and almost all $d(n) = 1$. Unfortunately, for $n \geq 9$ with $j \geq 2$ the rules for constructing the solution x_p from x_{p-1} are not so clear defined as for $2 \leq n \leq 8$. There exist explicit rules for each $n \geq 9$, but they are depending on the value of j and λ . At this point we cannot specify these explicit rules in an easy general manner. \boxplus

Corollary 10. Regarding to the *third step* of the algorithm of Theorem 5, for each $n \geq 2$ the solution x of each parity vector $v_{n,n+1,1}$ is given by

$$x_{n,n+1,1} = 2 \cdot x_{n,n,\mathcal{P}(n,n)} + 1 \quad \text{if} \quad 2 \cdot x_{n,n,\mathcal{P}(n,n)} + 1 < 2^{\sigma_n}, \quad (18)$$

or

$$x_{n,n+1,1} = 2 \cdot x_{n,n,\mathcal{P}(n,n)} + 1 - 2^{\sigma_n} \quad \text{if} \quad 2 \cdot x_{n,n,\mathcal{P}(n,n)} + 1 > 2^{\sigma_n}. \quad (19)$$

Note: $v_{n,n+1,1}$ is the last parity vector of each $\mathbb{V}(n)$ given by $x_n = 23, 15, 95, 575, \dots$ and the child of $v_{n,n,\mathcal{P}(n,n)}$ given by $x_n = 11, 7, 175, 287, \dots$ (cf. Figure 2) \boxplus

Corollary 11. Regarding to the *third step* of the algorithm of Theorem 5, for each $n \geq 2$ the solution x of a parity vector $v_{n,n+1,1}$ is explicit given with $\beta \in \{0, 1, 2, 3, 4, 5\}$ by the recurrence relation

$$x_n = \frac{2 \cdot x_{n-1} - 1 + \beta \cdot 2^{\kappa_n+2}}{3} \in \mathbb{N}, \quad (20)$$

if and only if

$$y_n = \frac{y_{n-1} + \beta \cdot 3^n}{d(n+1)} \in \mathbb{N}. \quad (21)$$

Note: There exist only six possibilities for x_n, y_n to solve (3) in \mathbb{N} . \boxplus

Appendix 9.1.8 gives a program for the recurrence relation of Corollary 11 and further information about the connectedness between Corollary 10 and 11.

Corollary 12. Regarding to the algorithm of Theorem 5 and especially to the *first step*, for each $n \geq 2, h \geq 2$, the solution x of each parity vector v_h which last element is "1" has a special arithmetic relationship to the solution x of a parity vector v_{h+1} , if v_{h+1} arises from v_h by the cyclic permutation of length 2 of its $E = \kappa(n) + 1$ elements given by

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \dots & E-1 & E \\ E & 1 & 2 & 3 & \dots & E-2 & E-1 \end{pmatrix}. \quad (22)$$

For Example, for $n = 4$ the cyclic permutation (22) applies to three parity vectors, given by

$$\begin{array}{lll} v_2(123) = (1, 1, 0, 1, 1, 0, 1) & \longrightarrow & v_3(39) = (1, 1, 1, 0, 1, 1, 0), \\ v_3(199) = (1, 1, 1, 0, 1, 0, 1) & \longrightarrow & v_4(175) = (1, 1, 1, 1, 0, 1, 0), \\ v_4(79) = (1, 1, 1, 1, 0, 0, 1) & \longrightarrow & v_5(95) = (1, 1, 1, 1, 1, 0, 0). \end{array}$$

The arithmetic relationship is given by

For each $n \equiv 1 \pmod{2}$ we have

$$x_h = \frac{3 \cdot x_{h+1} + 1}{2} - 2^{\kappa(n)} \pm 2^{\kappa(n)+1} \text{wehave} \quad (23)$$

if the right side of (23) < 0 , or

$$x_h = \frac{3 \cdot x_{h+1} + 1}{2} - 2^{\kappa(n)} \pm 2^{\kappa(n)+1} - 2^{\sigma_n} \quad (24)$$

if the right side of (24) $> 2^{\sigma_n}$.

\boxplus

8 Conclusion

At first sight the stopping time congruence classes $(\text{mod } 2^{\sigma_n})$, as listed in Chapter 2, convey the impression of randomness. There seems to be no regularity. The congruences seem to obey no law of order.

We have shown that this impression is deceptive. The finite stopping time behavior of the $3x + 1$ function is exactly defined by an algorithmic structure according to a directed rooted tree, whose vertices are the congruence classes $(\text{mod } 2^{\sigma_n})$. And there exists explicit arithmetic relationships between the parent and child vertices given by the Corollaries 8, 9 and 10. (cf. Figure 1 and 2)

Up to this point, our results on the congruences $(\text{mod } 2^{\sigma_n})$ are absolutely precise and clear. These results are given without the use of any statistical and probability theoretical methods. Even though Corollary 8 and 9 are not precise enough at this time to generate all solutions x precisely, from this point, statistical and probability theoretical methods could be used to show that the congruences $(\text{mod } 2^{\sigma_n})$ and the congruences $0 \pmod{2}$ and $1 \pmod{4}$ build the set of the natural numbers.

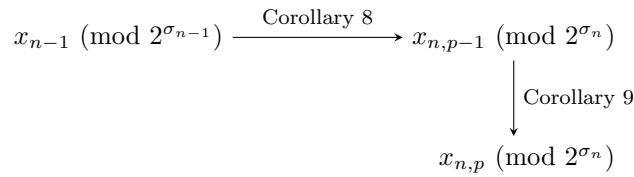


Figure 1: Tree construction principle (cf. Figure 3)

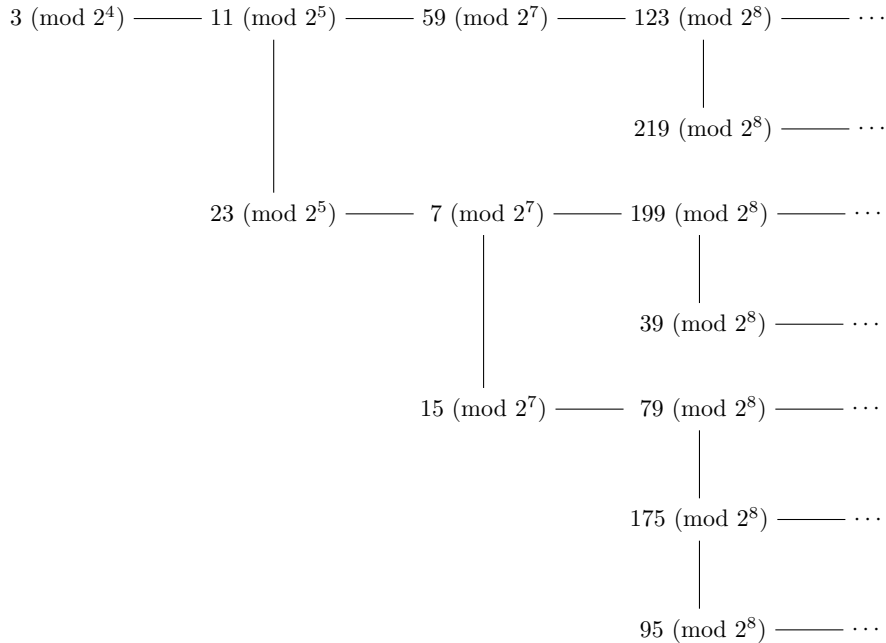


Figure 2: Directed rooted tree for the congruences $(\text{mod } 2^{\sigma_n})$. (cf. Figure 4)

One possibility to prove the $3x + 1$ Conjecture would be the following: Let us assume the most extreme case for Corollary 8 and 9. In regard to Remark 6, the values for x are thus as large as possible, whereby most of the small values (residual classes) are skipped. The equations (10), (15), (17) and (19) are the reason why there must still exist very small solutions for x , even if the values for n become very large. Thus it could be shown that there exist bounds for n such that all x smaller than a specific value have a finite stopping time.

9 Appendix

9.1 Algorithms in PARI/GP [4]

9.1.1 Program 1

Program 1 gives the algorithm for Theorem 2. It generates the values from Table 1 especially [A100982](#). It outputs the values from column n and their sum $z(n)$ for each $n \geq 2$.

```

1  {
2  limit=20; /* or limit>20 */
3  R=matrix(limit,limit);
4  R[2,1]=0;
5  R[2,2]=1;
6
7  for(n=2, limit,
8    print; print1("For n="n" in column n: ");
9    Kappa=floor(n*log(3)/log(2));
10   Zn=0;
11   for(k=n, Kappa,
12     R[k+1,n]=R[k,n]+R[k,n-1];
13     print1(R[k+1,n]", ");
14     Zn=Zn+R[k+1,n];
15   );
16   print; print(" and the sum is z(n)="Zn);
17 );
18 }
```

9.1.2 Program 2

Program 2 gives the algorithm for Theorem 2. It generates the values from Table 1 especially [A076227](#). It outputs the values from row k and their sum $w(k)$ for each $k \geq 2$.

```

1  {
2  limit=20; /* or limit>20 */
3  R=matrix(limit,limit);
4  R[2,1]=0;
5  R[2,2]=1;
6
7  for(n=2, limit,
8    if(n>2, print; print1("For k="n-1" in row k: "));
9    Kappa=floor(n*log(3)/log(2));
10   for(k=n, Kappa,
11     R[k+1,n]=R[k,n]+R[k,n-1];
12   );
13   t=floor(1+(n-1)*log(2)/log(3)); /* cf. A020915 */
14   Wk=0;
15   for(i=t, n-1,
16     print1(R[n,i]", ");
17     Wk=Wk+R[n,i];
18   );
19   if(n>2, print; print(" and the sum is w(k)="Wk));
20 );
21 }
```

9.1.3 Program 3

Program 3 gives the algorithm for Theorem 5. It generates the parity vectors of $\mathbb{V}(n)$ for $n \geq 2$ from the one initial parity vector of $\mathbb{V}(1)$. It outputs the parity vectors with h, p and its counting number which last value is equal to $z(n)$.

```

1  {
2    k=3;
3    Log32=log(3)/log(2);
4    limit=14; /* or limit>14 */
5    V=matrix(limit,60000);
6    xn=3;
7
8    /* initial parity vector of V(1) */
9    A=[]; for(i=1, 2, A=concat(A,i)); A[1]=1; A[2]=1;
10   V[1,1]=A;
11
12   for(n=2, limit,
13     print("n="n);
14     Sigma=floor(1+(n+1)*Log32);
15     d=floor(n*Log32)-floor((n-1)*Log32);
16     Kappa=floor(n*Log32);
17     Kappa2=floor((n-1)*Log32);
18
19     r=1; v=1;
20     until(w==0,
21       A=[]; for(i=1, Kappa2+1, A=concat(A,i));
22       A=V[n-1,v];
23       B=[]; for(i=1, Kappa+1, B=concat(B,i));
24       for(i=1, Kappa2+1, B[i]=A[i]);
25
26       /* step 1 */
27       if(d==1, B[k]=1; V[n,r]=B; r++; v++);
28       if(d==2, B[k]=0; B[k+1]=1; V[n,r]=B; r++; v++);
29
30       /* step 2 */
31       if(B[Kappa]==0,
32         for(j=1, Kappa-n,
33           B[Kappa+1-j]=B[Kappa+2-j]; B[Kappa+2-j]=0;
34           V[n,r]=B; r++;
35           if(B[Kappa-j]==1, break(1));
36         );
37       );
38
39       /* step 3 */
40       w=0; for(i=n+2, Kappa+1, w=w+B[i]);
41     );
42     k=k+d;
43
44     p=1; h2=3;
45     for(i=1, r-1,
46       h=0; B=V[n,i]; until(B[h]==0, h++);
47       if(h>h2, p=1; h2++; print);
48       print(V[n,i] "    "h-1"    "p"    "i);
49       p++;
50     );
51     print;
52   );
53 }

```

9.1.4 Program 4

Program 4 gives the same algorithm for Theorem 5 as Program 3, but it outputs the values from Table 2 column by column.

```

1  {
2    k=3;
3    Log32=log(3)/log(2);
4    limit=14; /* or limit>14 */
5    V=matrix(limit,60000);
6    xn=3;
7
8    /* initial parity vector of V(1) */
9    A=[]; for(i=1, 2, A=concat(A,i)); A[1]=1; A[2]=1;
10   V[1,1]=A;
11
12   for(n=2, limit,
13     print1("n="n" ");
14     Sigma=floor(1+(n+1)*Log32);
15     d=floor(n*Log32)-floor((n-1)*Log32);
16     Kappa=floor(n*Log32);
17     Kappa2=floor((n-1)*Log32);
18
19     r=1; v=1;
20     until(w==0,
21       A=[]; for(i=1, Kappa2+1, A=concat(A,i));
22       A=V[n-1,v];
23       B=[]; for(i=1, Kappa+1, B=concat(B,i));
24       for(i=1, Kappa2+1, B[i]=A[i]);
25
26       /* step 1 */
27       if(d==1, B[k]=1; V[n,r]=B; r++; v++);
28       if(d==2, B[k]=0; B[k+1]=1; V[n,r]=B; r++; v++);
29
30       /* step 2 */
31       if(B[Kappa]==0,
32         for(j=1, Kappa-n,
33           B[Kappa+1-j]=B[Kappa+2-j]; B[Kappa+2-j]=0;
34           V[n,r]=B; r++;
35           if(B[Kappa-j]==1, break(1));
36         );
37       );
38
39       /*step 3 */
40       w=0; for(i=n+2, Kappa+1, w=w+B[i]);
41     );
42     k=k+d;
43
44     p=1; h2=3; zn=0;
45     for(i=1, r-1,
46       h=0; B=V[n,i]; until(B[h]==0, h++);
47       if(h>h2, print1(" "p-1); zn=zn+p-1; p=1; h2++);
48       p++;
49     );
50     print1(" "p-1"      z(n)="zn+1); print;
51   );
52 }

```

9.1.5 Program 5

Program 5 gives an algorithm for generating $\mathcal{P}(h, n)$ for a fixed $h \geq 4$ and $n = 5, \dots, \text{limit}$. It outputs the values from Table 2 for a given row h .

```

1  {
2  h=4; /* or h>4 */
3  limit=20; /* or limit>20 */
4  Log32=log(3)/log(2);
5  if(h>7, h++);
6  if(h>8, print1("h="h-1:"), print1("h="h:"));
7  P1=P2=vector(limit);
8  a=2;
9  b=1;
10 n=4;
11 P1[1]=1;
12 P2[2]=n-(h-1);
13
14 until(n>limit-1,
15     n++;
16     value=1;
17     d1=floor(n*Log32)-floor((n-1)*Log32);
18     d2=floor((n-1)*Log32)-floor((n-2)*Log32);
19     b++;
20     if((d1==1) && (d2==2), a=0);
21     if((d1==2) && (d2==1), a=-1);
22     if((d1==2) && (d2==2), a=0; b--);
23     if(a+b-b2==2, b--);
24     b2=a+b;
25     for(a=2, a+b,
26         if((n>6) && (n==h-1), P2[a]=0);
27         P1[a]=P1[a-1]+P2[a];
28         value=value+P1[a];
29         a2=a;
30     );
31     if(d1==2, P1[a2+1]=P1[a2]; value=value+P1[a2+1]);
32     if((n>6) && (n==h-1), print1(" "1));
33     if(n>=h-1, print1(" "value));
34     for(i=2, b+1, P2[i]=P1[i]);
35 );
36 }
```

9.1.6 Program 6

The function NextPermutation(a) generates all permutations in lexicographic ordering of a *zero-one word* (5) as given by Table 3.

```

1  NextPermutation(a)=
2  {
3  i=#a-1;
4  while(!(i<1 || a[i]<a[i+1]), i--);
5  if(i<1, return(0));
6  k=#a;
7  while(!(a[k]>a[i]), k--);
8  t=a[k];
9  a[k]=a[i];
10 a[i]=t;
11 for(k=i+1, (#a+i)/2,
12     t=a[k];
13     a[k]=a[#a+1+i-k];
14     a[#a+1+i-k]=t;
15 );
16 return(a);
17 }
```

9.1.7 Program 7

Program 7 gives the algorithm for Corollary 8, especially for the first parity vector $v_{n,2,1}$ of each $\mathbb{V}(n)$ for $n \geq 2$. It outputs the integer solution x for these parity vectors with λ .

```

1  {
2    j=3;
3    limit=20; /* or limit>20; */
4    Log32=log(3)/log(2);
5    xn=3;
6
7    /* initial parity vector of V(1) */
8    B=[]; for(i=1, j+1, B=concat(B,i)); B[1]=1; B[2]=1;
9
10   for(n=2, limit,
11     Sigma=floor(1+(n+1)*Log32);
12     d=floor(n*Log32)-floor((n-1)*Log32);
13     Kappa=floor(n*Log32);
14
15     /* generate the new parity vector for n */
16     if(n>2, B=[]; for(i=1, Kappa+1, B=concat(B,i));
17       for(i=1, j-1, B[i]=A[i]);
18     );
19     if(d==2, B[j]=0; B[j+1]=1, B[j]=1);
20     j=j+d;
21     A=[]; for(i=1, Kappa+1, A=concat(A,i));
22     for(i=1, Kappa+1, A[i]=B[i]);
23
24     /* determine the n+1 values for Alpha[i] */
25     Alpha=[]; for(i=1, n+1, Alpha=concat(Alpha,i));
26     for(i=1, n+1, Alpha[i]=0);
27     i=1; for(k=1, Kappa+1, if(B[k]==1, Alpha[i]=k-1; i++));
28
29     /* calculate Lamda from Diophantine equation */
30     Lamda=1;
31     until(Lamda>7,
32       x=xn+Lamda*2^Kappa;
33       Sum=0; for(i=1, n+1, Sum=Sum+3^(n+1-i)*2^Alpha[i]);
34       y=(3^(n+1)*x+Sum)/2^Sigma;
35       if(frac(y)==0,
36         if(x>2^Sigma, x=x-2^Sigma);
37         print(n" "x" "Lamda);
38         xn=x;
39       );
40     Lamda=Lamda+2;
41   );
42 );
43 }
```


9.1.8 Program 8

Program 8 gives the algorithm for Corollary 11. It generates by a recurrence relation the solution x of the parity vector $v_{n,n+1,1}$ of $\mathbb{V}(n)$ for each $n \geq 2$. It outputs the integer solution x for these parity vectors with β .

```

1  {
2  limit=20; /* or limit>20; */
3  Log32=log(3)/log(2);
4  x2=y2=1;
5
6  for(n=1, limit,
7    Kappa=floor(n*Log32);
8    d=floor((n+1)*Log32)-floor(n*Log32); /* d(n+1) */
9
10   for(Beta=0, 5,
11     x=(x2+Beta*2^(Kappa+2))/3;
12     y=(y2+Beta*3^n)/d;
13     if(frac(x)==0 && frac(y)==0,
14       print(n" "x" "Beta);
15       break;
16     );
17   );
18   x2=2*x-1;
19   y2=y;
20 );
21 }
```

Note: If the program starts in line 6 with $n = 2$, the generated values for x give the solution of the parity vectors $v_{n,n,\mathcal{P}(n,n)}$. In this case the solution x of the parity vector $v_{n,n+1,1}$ is given by (18) or (19). (cf. Corollary 10).

9.2 The directed rooted tree produced by the algorithm of Theorem 5

Figure 3 illustrates the two directions of the construction principle for the tree.

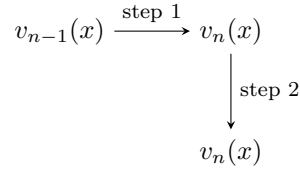


Figure 3: Tree construction principle

Figure 4 displays the beginning of the tree using the above construction principle. Each column of this tree indicates the parity vectors of $\mathbb{V}(n)$.

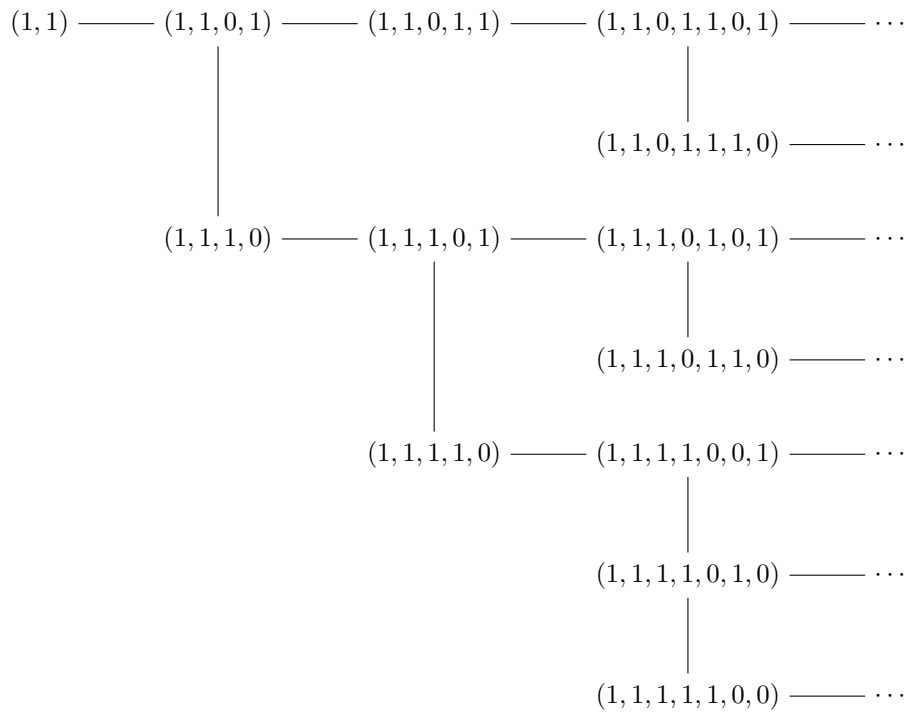


Figure 4: Beginning of the directed rooted tree for the parity vectors.

10 References

- [1] Lynn E. Garner, *On the Collatz $3n + 1$ Algorithm*, Proc. Amer. Math. Soc., Vol. 82, No. 1 (May, 1981), pp. 19 – 22.
(<http://www.jstor.org/stable/2044308>)
- [2] Jeffrey C. Lagarias, *The $3x + 1$ problem and its Generalizations*, The American Mathematical Monthly Vol. 92, No. 1 (January, 1985), pp. 3 – 23.
(<http://www.cecm.sfu.ca/organics/papers/lagarias/paper/html/paper.html>)
- [3] The On-Line Encyclopedia of Integer Sequences (OEIS), A020914, A020915, A022921, A056576, A076227, A100982, A177789, A293308
(<http://oeis.org>)
- [4] The PARI Group - PARI/GP Version 2.9.0
(<http://pari.math.u-bordeaux.fr>)
- [5] Eric Roosendaal, *On the $3x + 1$ problem*, web document.
(<http://www.eric.nl/wondrous/terras.html>)
- [6] Mike Winkler, *New results on the stopping time behaviour of the Collatz $3x + 1$ function*, March 2015.
(<http://arxiv.org/pdf/1504.00212v1.pdf>)
- [7] Mike Winkler, *On a stopping time algorithm of the $3n + 1$ function*, May 2011.
(http://mikewinkler.co.nf/collatz_algorithm.pdf)
- [8] Mike Winkler, *On the structure and the behaviour of Collatz $3n + 1$ sequences - Finite subsequences and the role of the Fibonacci sequence*, November 2014.
(<http://arxiv.org/pdf/1412.0519v1.pdf>)