THE PURE NUMBERS GENERATED BY THE COLLATZ SEQUENCE

Douglas J. Shaw
317 Wright Hall, University of Northern Iowa, Cedar Falls, IA 50614
(Submitted April 2004-Final Revision January 2005)

ABSTRACT

The Collatz Conjecture has been vexing mathematicians for over seventy years. It is the only mathematics problem that is comprehensible to a fourth-grader, yet has caused Paul Erdos to say, "Mathematics is not yet ready for such a problem." This paper divides the integers into "pure" and "impure" numbers, where the "impure" numbers occur in the trajectories of smaller numbers. It develops an infinite set of theorems characterizing the "impure" numbers, and establishes bounds on their density in the integers.

1. INTRODUCTION

The Collatz conjecture, also known as the $3n+1$ problem, the Syracuse problem, the Hailstone problem, Kakutani's problem, Hasse's algorithm and Ulam's problem, is tantalizingly simple to state and thus has seduced mathematical minds since the 1930's. Since then, many articles about it and its generalizations have appeared but a complete solution still evades the mathematical community. (See Lagarias [1] and Wirsching [2]) In this paper, we allow the Collatz sequence to lead us to dividing the integers into "pure" and "impure" numbers, and present a theorem-schema which allows us to generate a set of theorems, each theorem characterizing a different set of numbers as impure. The Collatz conjecture is true if and only if it is true on the set of pure numbers.

The Collatz sequence can be stated in several ways. This paper uses a traditional formulation. For n a positive integer, we define the function C by

$$C(n) = \begin{cases} 3n+1 & n \text{ odd} \\ \frac{n}{2} & n \text{ even} \end{cases} \tag{1.1}$$

and denote its iterates $C_k(n)$. The Collatz conjecture says that for all positive n, there exists k such that $C_k(n) = 1$.

2. PURE AND IMPURE NUMBERS

Consider the following Collatz sequence, starting with $n = 3$:

n	3
$C(n)$	10
$C_2(n)$	5
$C_3(n)$	16
$C_4(n)$	8
$C_5(n)$	4
$C_6(n)$	2
$C_7(n)$	1
$C_8(n)$	4
$C_9(n)$	2
$C_{10}(n)$	1

Notice that in the above example, once one verified that the conjecture held for $n = 3$, it would no longer be necessary to check $n = 4, 5, 8, 10$ and 16 since they have appeared in its orbit. (One would technically never need to check even numbers anyway, but the upcoming classification of integers has interest beyond this application)

Definition 2.1: A positive integer n is *pure* if its entire tree of preimages under the Collatz function C are greater than or equal to it; otherwise n is *impure*. Equivalently, a positive integer n is *impure* if there exists $r < n$ such that $C_k(r) = n$ for some k.

The complexity of the $3x + 1$ problem has to do with the iterates of the Collatz map increasing or decreasing in a complicated way. The notion of "purity" reflects certain properties of this dynamic, in focusing on numbers which are minimal in their entire tree of inverse iterates. This justifies studying the notion of "purity" for its own sake.

We start with a trivial theorem on pure numbers:

Theorem 2.1: If $n \equiv 0 \pmod 3$, then n is pure.

Proof: If $n \equiv 0 \pmod 3$ then its preimages are all of the form $2^k n$. $\square$

We will show that if $n \equiv 2 \pmod 3$ then n is impure, reducing our field of study to $n \equiv 1 \pmod 3$. Our analysis will center on proving certain classes of numbers to be impure, leaving the pure numbers to be the ones left over.

This paper will develop a theorem schema to generate sufficiency theorems to show a number is impure, such as the ones below:

Theorem 2.2: If $n \equiv 2 \pmod 3$ then n is impure.

The remaining case, $n \equiv 1 \pmod 3$, is not so simple.

Theorem 2.3: Let $n \equiv 1 \pmod 3$.

(1) If $n \equiv 4 \pmod 6$ then n is impure.
(2) If $n \equiv 4 \pmod 9$ then n is impure.
(3) If $n \equiv 10 \pmod{81}$ then n is impure.

These theorems will be developed by applying an algorithm to certain kinds of $\{0, 1\}$ vectors, called worthwhile vectors. To define worthwhile vectors, we use the following function:

Definition 2: Let $\overrightarrow{v} = (v_0, v_1, \ldots, v_{m-1})$ be a $\{0,1\}$ m-vector. Let $|\overrightarrow{v}| = \sum_{i=0}^{m-1} v_i$. We define the real-valued function f by

$$f\left(\overrightarrow{v}\right) = \frac{3^{|\overrightarrow{v}|}}{2^{\left(n-|\overrightarrow{v}|\right)}}. \tag{2.1}$$

Definition 3: Let $\overrightarrow{v} = (v_0, v_1, \ldots, v_{m-1})$ be a $\{0,1\}$ m-vector. We say that $\overrightarrow{v}$ is *worthwhile* if:

1. It is successor-free (i.e. there is no k for which $v_k = v_{k+1} = 1$)
2. $v_0 = 1$
3. $f(\overrightarrow{v}) > 1$ (equivalently: $|\overrightarrow{v}| > \frac{m \ln 2}{\ln 6}$).

Before developing the algorithm, we show its results for all worthwhile $\{0,1\}$ vectors with $m \leq 5$.

Vector	$f(\overrightarrow{v})$	Resultant sufficiency theorem
1	3	$n \equiv 4 \pmod 6$ is impure (Theorem 2.2)
10	3/2	$n \equiv 2 \pmod 3$ is impure (Theorem 2.1)
100	3/4	sequence not worthwhile
101	9/2	$n \equiv 16 \pmod{18}$ is impure (weaker form of Theorem 2.2)
1000	3/8	sequence not worthwhile
1001	9/4	$n \equiv 4 \pmod{18}$ is impure (weaker form of Theorem 2.2)
1010	9/4	$n \equiv 8 \pmod 9$ is impure (weaker form of Theorem 2.1)
10000	3/16	sequence not worthwhile
10001	9/8	$n \equiv 16 \pmod{18}$ is impure (weaker form of Theorem 2.2)
10010	9/8	$n \equiv 2 \pmod 9$ is impure (weaker form of Theorem 2.1)
10100	9/8	$n \equiv 4 \pmod 9$ is impure (Theorem 2.3)
10101	27/4	$n \equiv 52 \pmod{54}$ is impure (weaker form of Theorem 2.2)

3. THE THEOREM GENERATING ALGORITHM

We now present the algorithm by which a worthwhile vector is transformed into a impure-number sufficiency theorem. We let $\overrightarrow{v} = (v_0, v_1, \ldots, v_{m-1})$ be a $\{0,1\}$ worthwhile m-vector.

Step 0: Set $a_0 = 2, b_0 = 1$

Step i: (This step is iterated from $i = 1$ to m)

Case 1: If a_{i-1} is even and b_{i-1} is even

If v_{i-1} then the sequence was not worthwhile.

If $v_{i-1} = 0$ then set $a_i = \frac{a_{i-1}}{2}$ and $b_i = \frac{b_{i-1}}{2}$.

Case 2: If a_{i-1} is even and b_{i-1} is odd

If $v_{i-1} = 1$ then set $a_i = 3a_{i-1}$ and $b_i = 3b_{i-1} + 1$.

If $v_{i-1} = 0$ then the sequence was not worthwhile.

Case 3: If a_{i-1} is odd and b_{i-1} is even

If $v_{i-1} = 1$ then set $a_i = 6a_{i-1}$ and $b_i = 3a_{i-1} + 3b_{i-1} + 1$.

If $v_{i-1} = 0$ then set $a_i = a_{i-1}$ and $b_i = \frac{b_{i-1}}{2}$.

Case 4: If a_{i-1} is odd and b_{i-1} is odd

If $v_{i-1} = 1$ then set $a_i = 6a_{i-1}$ and $b_i = 3b_{i-1} + 1$.

If $v_{i-1} = 0$ then set $a_i = a_{i-1}$ and $b_i = \frac{a_{i-1}+b_{i-1}}{2}$.

After the algorithm is completed, we wind up with a pair (a_m, b_m) which are used in the following theorem:

Theorem 3.1: **(Sufficiency schema)** Let $\vec{v} = (v_0, v_1, \ldots, v_{m-1})$ be a $\{0,1\}$ worthwhile m-vector. Let (a_m, b_m) be the result of applying the above algorithm to $\vec{v}$. Then it is true that if $n \equiv b_m \pmod{a_m}$ then n is impure.

Proof: For a given integer n, define a sequence of 0-1 valued quantities $x_i(n)$ by

$$C_i(n) \equiv x_i(n) \pmod 2 \tag{3.1}$$

as in [1]. It can be shown that if $\vec{v}$ is a successor-free $\{0,1\}$ m vector, then $\vec{v} = (x_0(n), x_1(n), \ldots, x_{m-1}(n))$ for some n. (This can be shown by applying Theorem B in [1], and using the fact that a $(\ldots, 1, 0, \ldots)$ in the traditional formulation of the conjecture corresponds to a $(\ldots, 1, \ldots)$ in Lagarias'.)

We now can prove the Sufficiency Schema theorem. We start with a successor-free $\{0,1\}$ m-vector $\vec{v}$, and generate its associated n. We first show that $C_m(n) \equiv b_m \pmod{a_m}$. Then we show that the process is reversible, that if $q \equiv b_m \pmod{a_m}$, where a_m and b_m are the result of applying the algorithm to a vector $\vec{v}$, then there is a p such that $C_m(p) \equiv q$, and $\vec{v} = (x_0(p), x_1(p), \ldots, x_{m-1}(p))$. Finally, we show that if $\vec{v}$ is a worthwhile vector, then it is true that $p < q$ and thus q is impure.

Lemma 3.1: If $\vec{v} = (x_0(p), x_1(p), \ldots, x_{m-1}(p))$ for some p, and $x_0(p) = 1$, then $q = C_m(p) \equiv b_m \pmod{a_m}$.

Proof: Since $x_0(p) = 1$ we know that p is odd. So we can write

$$p = a_0 k_0 + b_0 \tag{3.2}$$

where $a_0 = 2, b_0 = 1$ and k_0 is a nonnegative constant. Now

$$C_1(p) = 3p + 1 = 6k_0 + 4. \tag{3.3}$$

So we can say that $a_1 = 6$ and $b_1 = 4$. We can also set $k_1 = k_0$ to write

$$C_1(p) = a_1 k_1 + b_1. \tag{3.4}$$

We now iterate this procedure. Note that if $v_i = 1$ then $C_{i+1}(p) = 3C_i(p) + 1$ and if $v_i = 0$ then $C_{i+1}(p) = \frac{C_i(p)}{2}$.

Case 1: If a_i is even and b_i is even

In this case, $a_i k_i + b_i$ is always even, and thus $C_i(p) = \frac{a_i}{2} k_i + \frac{b_i}{2}$. We set $a_{i+1} = \frac{a_i}{2}, b_{i+1} = \frac{b_i}{2}$ and $k_{i+1} = k_i$ to obtain $C_{i+1}(p) = a_{i+1} k_{i+1} + b_{i+1}$

Case 2: If a_i is even and b_i is odd

In this case, $a_i k_i + b_i$ is always odd, and thus $C_i(p) = 3a_i k_i + 3b_i + 1$. We set $a_{i+1} = 3a_i, b_{i+1} = 3b_i + 1$ and $k_{i+1} = k_i$ to obtain $C_{i+1}(p) = a_{i+1} k_{i+1} + b_{i+1}$

Case 3: If a_i is odd and b_i is even

Here, $a_ik_i + b_i$ may be odd or even, depending on the parity of k_i. If $v_i = 0$ then k_i is even and we set $k_i = 2k_{i+1}$. We now have $C_{i+1}(p) = \frac{2a_ik_{i+1}+b_i}{2} = a_ik_{i+1} + \frac{b_i}{2}$. We set $a_{i+1} = a_i, b_{i+1} = \frac{b_i}{2}$ to obtain $C_{i+1}(p) = a_{i+1}k_{i+1} + b_{i+1}$.

If $v_i = 1$ then k_i is odd and we set $k_i = 2k_{i+1} + 1$. We now have $C_{i+1}(p) = 3(a_i(2k_{i+1} + 1) + b_i) + 1$. We set $a_{i+1} = 6a_i, b_{i+1} = 3a_i + 3b_i + 1$ to obtain $C_{i+1}(p) = a_{i+1}k_{i+1} + b_{i+1}$.

Case 4: If a_i is odd and b_i is odd

Here, $a_ik_i + b_i$ may be odd or even, depending on the parity of k_i. If $v_i = 0$ then k_i is odd and we set $k_i = 2k_{i+1} + 1$. Applying the same process as above, we obtain $a_{i+1} = a_i, b_{i+1} = \frac{a_i+b_i}{2}$ such that $C_{i+1}(p) = a_{i+1}k_{i+1} + b_{i+1}$.

If $v_i = 1$ then k_i is even and we set $k_i = 2k_{i+1}$. In this case we obtain $a_{i+1} = 6a_i, b_{i+1} = 3b_i + 1$ such that $C_{i+1}(p) = a_{i+1}k_{i+1} + b_{i+1}$.

After the algorithm is completed, we have that $q = C_m(p) = a_mk_m + b_m$ and thus that $q \equiv b_m \pmod{a_m}$. $\square$

Lemma 3.2: If $q \equiv b_m \pmod{a_m}$, where a_m and b_m are the results of applying the algorithm to a vector $\vec{v}$, then there is a p such that $C_m(p) = q$ and $\vec{v} = (x_0(p), x_1(p), \ldots, x_{m-1}(p))$.

Proof: For a given $\vec{v}$, we can think of k_n as a function of k_0, obtained as a composition of the functions $2x, 2x + 1$ and the identity function. This is an invertible function. Thus we can write $q = a_nK + b_n$ for some K, and apply the inverse function to obtain a k_0 such that $K = k_n$. Now let $p = 2k_0 + 1$. Applying the algorithm gives $C_m(p) = q$. $\square$

We now have that, for any $\{0, 1\}$ successor-free vector $\vec{v}$ (with $v_0 = 1$) if q is the appropriate parity, it appears as part of a number p's orbit. We now need show that if $\vec{v}$ is worthwhile, then $p < q$.

Lemma 3.3: Let $\vec{v} = (x_0(p), x_1(p), \ldots, x_{m-1}(p))$ for some odd p. Then $C_m(p)$ is given explicitly by

$$C_m(p) = \left(2^{\left(m-|\vec{v}|\right)}\right)^{-1} \left(p3^{|\vec{v}|} + \sum_{k=0}^{m-1} v_k 2^{\left(k-\sum_{i=0}^{k-1} v_i\right)} 3^{\left(\sum_{i=k+1}^{m-1} v_i\right)}\right). \tag{3.5}$$

Notice that the expressions immediately to the right of the 2s and the 3s are their exponents, not multiplicands.

Proof: By induction on n.

If $n = 2$, then $\vec{v} = (1, 0)$ and the result follows. $\left(C_2(p) = \frac{3p+1}{2}\right)$.

Assume true for n, and consider $n + 1$. If $\vec{v}_{n+1} = 0$, we want to leave the numerator of equation (3.5) unchanged, and to increase the denominator by a factor of 2. It is easy to check that this occurs.

If $\vec{v}_{n+1} = 0$, we want to leave the denominator of equation (3.5) unchanged. This is also easy to check.

In going from x to $3x+1$, we want to multiply the numerator by 3, and then add $2^{\left(m-|\vec{v}|\right)}$ to it. We see that the summations that appear as exponents of 3 are increased by one. A new non-zero term is added at the end, $2^{\left(m-|\vec{v}|\right)}$. $\square$

Lemma 3.4: Let $\vec{v}$ be a worthwhile $\{0, 1\}$ m vector. Let p be an integer such that $x_i(p) = v_i$. If $q = C_m(p)$ then $p < q$.

Proof: By the previous lemma, we have an explicit formula for q in terms of p. This formula could be written $q = kp + b$ where b is always positive and k is given explicitly by

$$k = \left(2^{\left(m-|\vec{v}|\right)}\right)^{-1}\left(3^{|\vec{v}|}\right). \tag{3.6}$$

So if $k > 1$, then $p < q$. □

As previously stated, Lemmas 3.1 - 3.4 prove the sufficiency theorem, Theorem 3.1.

We have shown that we can create sufficiency theorems, such as Theorems 2.2 - 2.3, that characterize certain classes of numbers as impure. We now show the converse, that every impure number can be shown to be impure by one of the theorems generated by the scheme.

Theorem 3.2: For every impure number n there is a sufficiency theorem, generated by a worthwhile vector, that shows n to be impure.

Proof: If n is impure, there is a number $p < n$ with $n = C_k(p)$. Let $\vec{v}$ be the associated $\{0, 1\}$ vector and apply the algorithm to $\vec{v}$. □

4. OBSERVATIONS AND EXTENSIONS

The theorems above dealt with branding certain numbers as impure. Is it possible to characterize the pure numbers? We can nicely summarize the findings of Theorems 2.1, 2.2, 2.3 (1) and 2.3 (2) in the following table:

$n \equiv 0 \pmod{18}$	n is pure
$n \equiv 1 \pmod{18}$	n may be pure or impure
$n \equiv 2 \pmod{18}$	n is impure
$n \equiv 3 \pmod{18}$	n is pure
$n \equiv 4 \pmod{18}$	n is impure
$n \equiv 5 \pmod{18}$	n is impure
$n \equiv 6 \pmod{18}$	n is pure
$n \equiv 7 \pmod{18}$	n may be pure or impure
$n \equiv 8 \pmod{18}$	n is impure
$n \equiv 9 \pmod{18}$	n is pure
$n \equiv 10 \pmod{18}$	n is impure
$n \equiv 11 \pmod{18}$	n is impure
$n \equiv 12 \pmod{18}$	n is pure
$n \equiv 13 \pmod{18}$	n is impure
$n \equiv 14 \pmod{18}$	n is impure
$n \equiv 15 \pmod{18}$	n is pure
$n \equiv 16 \pmod{18}$	n is impure
$n \equiv 17 \pmod{18}$	n is impure

The inclusion of Theorem 2.3 (3) would not add a lot to this table; it would take the cases where n is congruent to 1 or 7 mod 18, and classify as impure those numbers that are also 10 modulo 81, i.e. adding information to the special case of $n \equiv 91 \pmod{162}$.

Observation 4.1: Theorems 2.1 - 2.3 cover all of the impure numbers from 1 - 500 except for 61, 205, 325 and 433. Surprisingly, the shortest vector that generates the theorem to cover 61 has length 85. This is because 61 appears very late in the orbit of 27, and so the vector linking 27 to 61 is very long.

Observation 4.2: If $\vec{v}$ is a worthwhile m vector, the condition $|\vec{v}| > \frac{m \ln 2}{\ln 6} \approx 0.387m$ gives a lower bound on $|\vec{v}|$. Similarly, the condition that $\vec{v}$ is successor-free means that $|\vec{v}| < \frac{m+1}{2} \approx 0.5m$, which gives an upper bound on $|\vec{v}|$. So, for a given large m, the set of worthwhile m vectors have a relatively narrow range of admissible sizes: $0.386m < |\vec{v}| < \frac{m+1}{2}$.

We can also categorize the theorems that result from the generating algorithm.

Theorem 4.2: If a sufficiency theorem (b mod a is impure) comes from a worthwhile m vector $\vec{v}$, with $|\vec{v}| = k$, then $a = 3^k$ if $\vec{v}_{m-1} = 0$, and $a = 2(3^k)$ if $\vec{v}_{m-1} = 1$.

Proof: At step i, if $\vec{v}_i = 1$ then a is multiplied by 3 or 6. If it is multiplied by 6, b will be divided by 2 if there is a next iteration. $\square$

Observation 4.3: If the Collatz Conjecture is true, then there is an algorithm to decide if a given integer n is pure. All we need do is test every number $n' < n$, to see if n appears in the trajectory of n'. These tests are guaranteed to terminate if the conjecture is true.

5. THE DENSITY OF IMPURE NUMBERS IN $\mathbb{R}$

Let I be the set of impure numbers. The asymptotic density of I is given by

$$d = \lim_{N \to \infty} \frac{1}{N} | \{n : n \in I, 1 \leq n \leq N\} . \tag{5.1}$$

Clearly, if this limit exists, then $d \leq \frac{2}{3}$, by Theorem 4.1.

Theorem 5.1: There exists a set of sufficiency theorems (b mod a is impure) generated by the algorithm, that induce a partition $P = \{P_1, P_2, \dots\}$ of I, where P_i are sets of the form $\{n | n \equiv b \pmod{k}, n > 0, k = 3^l \text{ or } 2 \cdot 3^l \text{ for some } l > 0\}$.

Proof: We construct P as follows. The first element is $P_1 = \{2, 5, 8, 11, 14, \dots\}$, corresponding to theorem 2.2.

Assume that P is not yet a partition of I. Let i be the smallest element of I not covered. Theorem 3.2 guarantees a sufficiency theorem, $i \pmod{a}$, covering i. If $\{i, i+a, i+2a, \dots\}$ is disjoint from the elements of P, then let $\{i, i+a, i+2a, \dots\}$ be the next element of P and $i \pmod{a}$.

If $\{i, i+a, i+2a, \dots\}$ is not disjoint from all the elements of P, then let k be the lcm of the modulii of the elements of the partion that intersect it. (Theorem 4.2 guarantees that k will be of the form $2^r 3^s$, where $r \in \{0, 1\}$). Now the subsequence $\{i, i+ka, i+2ka, \dots\}$ covers i, and is disjoint from all the elements of P. $\square$

Corollary 5.1: The set P of pure numbers and the set $I = \mathbb{N} \backslash P$ of impure numbers each have a natural density.

Proof: We will show $d \lim_{N \to \infty} \frac{1}{N} |\{n : n \in I, 1 \leq n \leq N\}|$ exists. This will be the density of the impure numbers, and the density of the pure numbers will be $1 - d$.

Let $P_i = \{b_i, b_i + a_i, b_i + 2a_i, \dots\}$. Then $\lim_{N\to\infty} \frac{1}{N}|\{n : n \in P_i, 1 \le n \le N\} = \frac{1}{a_i}|$. We thus obtain that $d = \sum_{i=1}^{\infty} \frac{1}{a_i}$. Now the sequence of partial sums of this series is increasing and bounded by $\frac{2}{3}$ (Theorem 2.1), therefore d exists and is less than or equal to $\frac{2}{3}$. □

Observation 5.1: $\frac{91}{162} < d < \frac{2}{3}$.

The following table shows the first four elements of the partition, and hence the first four partial sums of d.

Partition	Sufficiency Theorem	Partial Sum
$P_1 = \{2, 5, 8, 11, 14, \dots\}$	$n \equiv 2 \pmod 3$ is impure	$\frac{1}{3}$
$P_2 = \{4, 10, 16, 22, 28, \dots\}$	$n \equiv 4 \pmod 6$ is impure	$\frac{1}{3} + \frac{1}{6} = \frac{1}{2}$
$P_3 = \{13, 31, 49, 67, 85, \dots\}$	$n \equiv 13 \pmod{18}$ is impure	$\frac{1}{3} + \frac{1}{6} + \frac{1}{18} = \frac{5}{9}$
$P_4 = \{91, 253, 415, 577, \dots\}$	$n \equiv 91 \pmod{162}$ is impure	$\frac{1}{3} + \frac{1}{6} + \frac{1}{18} + \frac{1}{162} = \frac{91}{162}$

Computer analysis of the first million positive integers gives a lower bound on d of 0.567636. For comparison, $\frac{91}{162} \approx 0.561728$.

The following is a list of all the pure numbers, not congruent to 0 mod 3, less than 1000:

1	7	19	25	37	43	55	73	79	97	109	115	127
133	145	151	163	169	181	187	199	217	223	235	241	259
271	277	289	295	307	313	331	343	349	361	367	379	385
397	403	421	439	451	457	469	475	487	493	505	511	523
529	541	547	559	565	583	595	601	613	619	631	649	655
667	673	685	691	703	709	721	727	745	757	763	775	781
793	799	811	817	829	835	847	853	865	871	883	889	907
925	937	943	955	961	973	979	997					

Notice that, as predicted earlier, all the pure numbers that are not congruent to 0 mod 3 are congruent to 1 or 7 mod 18.

ACKNOWLEDGMENTS

The author would like to thank the reviewers for their comments and suggestions.

REFERENCES

[1] J. C. Lagarias. "The 3x+1 Problem and Its Generalizations." *Amer. Math. Monthly* **92** (1985): 3-23 (MR86I:11043).

[2] G. J. Wirsching. *The Dynamical System Generated by the 3n+1 Function*, Lecture Notes in Math. No. 1681. Springer-Verlag, Berlin, 1998, pp. 158.

AMS Classification Numbers: 11B37, 11B83

✠ ✠ ✠