The "$3x + 1$" Problem and Finite Automata

Jeffrey Shallit*
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
shallit@graceland.waterloo.edu

David Wilson
4 Colby Brook Estates
Rt. 2 Box 78
Epsom, NH 03234
USA
dwilson@cvbnet.prime.com

Abstract

Let $f(x) = 3x + 1$ if x odd, and $x/2$ if x even. The "$3x + 1$" conjecture states that for all integers $n \geq 1$, there exists an $i \geq 0$ such that $f^i(n) = 1$. In this note we give a relationship between this famous conjecture and finite automata.

1 Introduction.

Let f be a function from the positive integers to the positive integers, defined as follows:

$$f(x) = \begin{cases} 3x + 1, & \text{if } x \text{ odd;} \\ x/2, & \text{if } x \text{ even.} \end{cases}$$

The Collatz, or "$3x + 1$" conjecture states that for all integers $n \geq 1$, there exists an $i \geq 0$ such that $f^i(n) = 1$. By $f^i(n)$ we mean f iterated i times with itself, evaluated at n, i.e.

$$\overbrace{f(f(f(\cdots f}^{i}(n)\cdots))).$$

The sequence of iterates $(x, f(x), f^2(x), \ldots)$ is called the *trajectory* of x. Following Lagarias [1985], we classify the possible trajectories as follows:

*Supported in part by a grant from NSERC.

(i) *convergent*: there exists $i \geq 0$ such that $f^i(x) = 1$;

(ii) *non-trivial cyclic*: the sequence $f^i(x)$ is eventually periodic, and $f^i(x) \neq 1$ for all $i \geq 0$;

(iii) *divergent*: $\lim_{i \to \infty} f^i(x) = \infty$.

Thus, we can rephrase the "$3x+1$" conjecture as follows: every trajectory is convergent.

Although the "$3x+1$" problem has been studied since at least 1952, little progress has been made. For a detailed survey, see the article of Lagarias [1985].

In this note we give a connection between the "$3x+1$" problem and finite automata, first noticed by the second author.

2 Representations and k-automatic Sets

Let $w = w_1 w_2 \cdots w_k \in (0+1)^*$ be a string. If

$$\sum_{i=1}^{k} w_i 2^{k-i} = n,$$

we say that w is a *representation* for n, and we write $[w] = n$. For example, $[0101] = 5$.

If $w = \epsilon$, the empty string, or if $k \geq 1$ and $w_1 \neq 0$, we say that w is the *canonical representation* for n, and we write $n_{(2)} = w$. For example, $5_{(2)} = 101$. Note that $[\epsilon] = 0$, and $0_{(2)} = \epsilon$. The set of all canonical representations is $C = \epsilon + 1(0+1)^*$.

Let S be a set of non-negative integers. Then we say S is 2–*automatic* (or 2–*recognizable*) if the set $r(S)$ of canonical representations for elements of S is regular. Formally,

$$r(S) = \{n_{(2)} \mid n \in S\}.$$

For example, the set

$$P = \{1, 2, 4, 8, 16, \ldots\}$$

of powers of 2 is 2–regular; its set of canonical representations $r(P)$ is 10^*. For more on these concepts, see Perrin [1990].

The following is a useful trick: to show that a set of integers S is 2–automatic, it suffices to show that *any* set whatsoever of representations for members of S is regular, not just the set of canonical representations. More precisely, if $L \subseteq (0+1)^*$ is a language such that

$$S = \{[w] \mid w \in L\},$$

then $r(S)$ is regular if L is regular. This is easy to see, since

$$r(S) = (L^R/0^*)^R \cap C.$$

Here w^R denotes the *reversal* of the word w, L^R denotes the language whose words are reversals of the words in L, and L_1/L_2 denotes the language-theoretic quotient, i.e.

$$L_1/L_2 = \{x \mid \exists y \in L_2 \text{ such that } xy \in L_1\}.$$

3 The Main Result

We now define a function to count the number of times the map $x \to 3x+1$ is used in the trajectory of n. More precisely, if the trajectory of n is convergent, we define

$$g(n) = \begin{cases} g(3n+1)+1, & \text{if } n > 1 \text{ is odd;} \\ g(n/2), & \text{if } n \text{ is even;} \\ 0, & \text{if } n = 1. \end{cases}$$

If the trajectory is non-trivial cyclic, or divergent, we define $g(n) = \infty$.

We let

$$S_i = g^{-1}[i] = \{k \geq 1 \mid g(k) = i\}.$$

Thus S_i is the set of integers n whose trajectories (up to $f^j(n) = 1$) contain i instances of the map $x \to 3x+1$. For example,

$$S_0 = \{1, 2, 4, 8, 16, \ldots\}$$

and

$$S_1 = \{5, 10, 20, 21, 40, 42 \ldots\}.$$

Remark.

It is easy to see that each S_i is infinite. For define $a_1 = 5$ and a_{i+1} as follows. If $a_i \equiv 1 \pmod 3$, then

$$a_{i+1} = \begin{cases} (4a_i - 1)/3, & \text{if } 4a_i \not\equiv 1 \pmod 9; \\ (16a_i - 1)/3, & \text{if } 4a_i \equiv 1 \pmod 9. \end{cases}$$

If $a_i \equiv 2 \pmod 3$, then

$$a_{i+1} = \begin{cases} (8a_i - 1)/3, & \text{if } 8a_i \not\equiv 1 \pmod 9; \\ (32a_i - 1)/3, & \text{if } 8a_i \equiv 1 \pmod 9. \end{cases}$$

Then we leave it to the reader to show that $a_{i+1} > a_i$, $a_i \in S_i$, and hence $a_i 2^j \in S_i$ for all $j \geq 0$.

We now state the main result of this note.

Theorem 3.1 *S_i is 2–automatic for all $i \geq 0$.*

We prove the result by induction on i. The result is clearly true for $i = 0$, since $S_0 = P$, the set of powers of 2.

Now we assume the theorem is true for i, and we prove it for $i+1$.

First, we introduce some notation: if S and T are sets of integers, we define

$$ST = \{st \mid s \in S,\ t \in T\}$$

and

$$S + T = \{s + t \mid s \in S,\ t \in T\}.$$

We also introduce the following function, inspired by the programming language APL: if $w = w_1 w_2 \cdots w_k$ is a string and n is a non-negative integer, then

$$\text{take}(n, w) = \begin{cases} w_1 w_2 \cdots w_n, & \text{if } n \le k; \\ w_1 w_2 \cdots w_k 0^{n-k}, & \text{if } n > k. \end{cases}$$

Now it is easy to see that, for $i \ge 0$, we have

$$S_{i+1} = (A_i \cap B)P,$$

where

$$B = \{3, 5, 7, 9, 11, \ldots\},$$

the odd numbers ≥ 3, and

$$A_i = \{x \mid 3x + 1 \in S_i\}.$$

Thus $r(S_{i+1}) = (r(A_i) \cap r(B))0^*$; since $r(B) = 1(0 + 1)^*1$, it suffices to see that $r(A_i)$ is regular.

The idea of the construction is very simple, although the details are a bit messy. Since $r(S_i)$ is regular, it is accepted by some finite automaton. On input w with $[w] = n$, we would like to simulate that automaton on $(3n + 1)_{(2)}$. There are two minor problems, however. The first is that we cannot easily compute $3n + 1$ digit-by-digit unless we start with the *least* significant digit first. The second is that the representation for $3n + 1$ may be up to two bits longer than the representation for n.

Each problem is easy to handle, however. For the first, we work with the reversed representation, where the least significant digit appears first. For the second, we use representations in which leading zeros (actually, trailing zeros, since the representation is reversed!) are allowed. Here are the details:

By induction, $r(S_i)$ is regular, and therefore the set $T = r(S_i)^R 0^*$ is also regular. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton for T. (We use the conventions from Hopcroft and Ullman [1979].) We now construct a finite automaton M' that, on input $x \in (0 + 1)^*$, simulates M on input w, where

$$w = \text{take}(|x|, (3[x^R] + 1)^R_{(2)}).$$

For example, on input 11100, M' simulates M on input 01101. M' accepts x iff M accepts w and $[w^R] = 3[x^R] + 1$.

More formally, let $M' = (Q', \Sigma, \delta', q_0', F')$. Each element of Q' is a pair $[s, c]$, where s represents a state of Q in the simulated computation, and c denotes the "carry" left over from the computation of $3x + 1$ digit-by-digit. Then $q_0' = [q_0, 1]$,

$$\delta'([s, c], a) = [\delta(s, (3a + c) \bmod 2), \lfloor (3a + c)/2 \rfloor],$$

and $F' = \{[s, 0] \mid s \in F\}$.

Let L be the language accepted by M'. We leave it to the reader to show that that $A_i = \{[x] \mid x \in L^R\}$. Hence by the trick at the end of Section 2, A_i is 2-automatic.

This completes the proof. ■

Remark.

The same technique can be used to prove that if S and T are 2–automatic sets, then so is $S + T$. Note, however, that there exist 2–automatic sets S, T such that ST is not 2–automatic; there is a simple counterexample, which the reader may enjoy finding.

References

[Hopcroft and Ullman 1979]
J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[Lagarias 1985]
J. C. Lagarias. The $3x + 1$ problem and its generalizations. *Amer. Math. Monthly* **92** (1985), 3–23.

[Perrin 1990]
D. Perrin. Finite automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Vol. B: Formal Models and Semantics, pages 1–57. MIT Press/Elsevier, 1990.