COUNTING THE NUMBER OF WINNING BINARY STRINGS IN THE 1-DIMENSIONAL SAME GAME

Chris Burns

Department of Mathematics, The University of British Columbia,
1984 Mathematics Rd, Vancouver, B.C., Canada V6T 1Z2

Benjamin Purcell

Department of Mathematics, The University of British Columbia,
1984 Mathematics Rd, Vancouver, B.C., Canada V6T 1Z2
e-mail: benjamin.purcell@gmail.com

(Submitted December 2005-Final Revision September 2006)

ABSTRACT

A simple matching game played with binary strings is related to the Fibonacci numbers. Using a counting argument, we show that the number of strings of length n that cannot result in a win is an integer multiple of the $(n-2)$'nd Fibonacci number, not counting certain trivial strings.

This paper is concerned with what is called the *same game*, a game played with binary strings. The object of the game is to eliminate a binary string by removing runs of consecutive identical digits. Stephan [2] conjectured that the number of winning strings of length n is

$$2^n - 2nF_{n-2} - (-1)^n - 1,$$

or equivalently, the number of losing strings is

$$2nF_{n-2} + (-1)^n + 1. \tag{1}$$

At heart, our proof is this: by excluding two trivial losing strings that only occur when n is even, we will show that under an equivalence relation, the losing strings can be partitioned into F_{n-2} sets of size $2n$, thus establishing Stephan's conjecture.

We define the same game as follows. Let S be a binary n-string with a run of $k > 1$ consecutive identical digits. Then we define a *reduction rule* by removing those consecutive identical digits thus producing an $(n-k)$-string. Strings that can be reduced to null by a sequence of reduction rules are called *winning strings*. *Losing strings* are those strings that are not winning strings. The reduction rules along with the binary strings constitute the same game. When n is even, we call the two losing strings which repeat '10' $n/2$ times or '01' n/2 times the trivial losing strings or just trivial strings. These are just the strings we exclude from account in formula (1). All other strings are non-trivial.

Let Q_n be the set of binary strings of length n. We proceed by defining two functions $r, c : Q_n \to Q_n$. The function r rotates a string; it is defined by

$$a_1a_2\cdots a_n \mapsto a_2a_3\cdots a_na_1.$$

The function c maps a string to its complement, i.e., each element a_i is mapped to 0 if $a_i = 1$ and 1 if $a_i = 0$. We now define an equivalence relation $\sim$ on Q_n as the least equivalence relation such that $S_1 \sim S_2$ when $rS_1 = S_2$ or $cS_1 = S_2$ for two binary strings $S_1, S_2 \in Q_n$. An equivalence class that contains a binary string S is called the *orbit* of S. We now come to our first proposition.

Proposition 1: *Every orbit of Q_n consists entirely of winning strings or entirely of losing strings.*

Proof: Assume S is a winning string of length n that begins with 0. Then there is some sequence of removals that reduces S to null, one of which removes the first 0 in S along with at least one other 0. Notice that removing this 0 cannot bring together a run of consecutive identical digits. Therefore, we can choose to save this run of 0's for the end, making its removal the final removal. Similarly, we can choose to save any 0's at the end of the string for our final removal.

Now consider rS. This string resembles S, except that the 0 at the beginning is now on the end. Notice that we can perform moves on rS analogous to those performed on S above except for the last move that takes S to null. This is because in the previous paragraph we "protect" any 0's on the ends of S. Thus, if we follow the same sequence of removals as before, but this time on rS, we will be left with a single run of 0's. We can remove these so that rS is also a winning string. What is more, this argument can be repeated for S beginning with 1.

It is obvious that cS must also be a winning string. Thus we have shown that, given a winning string S, cS and rS also win, proving that any orbit containing a winning string contains only winning strings since the functions c and r generate our equivalence relation. This implies that any orbit containing a losing string contains only losing strings. □

Thus we may now refer unambiguously to losing orbits and winning orbits. Moreover, we will use Proposition 1 for a corollary, which concerns a related game with nicer properties. Consider a variant of the same game we call the *wraparound same game* (or *wraparound game*). In this variant all reduction rules in the same game are valid and we also allow removal of runs of consecutive identical digits that wrap around the end of the string. Thus

$$0010 \mapsto 1$$

is a reduction rule in the wraparound game.

Corollary 2: *S is a winning string in the same game if and only if S is a winning string in the wraparound game.*

Proof: We prove the non-trivial direction by induction on n the length of a winning string. So assume that any winning string in the wraparound game of length $j < n$ is a winning string in the standard same game. Assume S is a winning n-string in the wraparound game. Consider the case where we remove a run of length k that wraps around. Then we can rotate S to $r^i S$ for some integer i so that this run does not wrap around and eliminate it in the same game. This produces a winning $(n-k)$-string in the wraparound game which by our induction hypothesis is a winning string in the same game. Thus $r^i S$ is a winning string in the same game so that Proposition 2 implies that S is a winning string in the same game. □

For what follows, it is advantageous to use the wraparound game and Corollary 2 allows us to do this. That is, we may unambiguously refer to a winning or losing string without reference to a particular game and we also consider the first and last digits of a string as consecutive.

Lemma 3: *Given any non-trivial losing binary n-string S, there is a rotation r^iS that can be reduced to a single digit in the standard same game.*

Proof: Assume S is a non-trivial losing n-string so that S must contain a run of two or more consecutive identical digits. When we remove this run in the wraparound game, the two digits on either side are identical, so we bring them together and remove them and any other identical digits within the run. This brings together two more identical digits which we remove in a similar fashion. Eventually this must terminate with a single digit a_i in S for $1 \leq i \leq n$ by our assumption that S is a losing string.

Consider the rotation $r^{i-1}S$ making a_i the first digit. Because we do not remove a_i, placing it as the first digit ensures that no run we removed in S will wrap around in $r^{i-1}S$. Thus, all the removals performed on S become allowable removals in the standard same game when performed on $r^{i-1}S$. □

We now consider the losing orbits: the following two propositions establish that every losing orbit of Q_n has size $2n$. We will see that this relates to the $2n$ term in formula (1).

Proposition 4: *Let S be a non-trivial binary n-string. Assume that $r^kS = S$ for some k, $0 < k < n$. Then S is a winning string.*

Proof: Choose the smallest positive k for which $r^kS = S$. Then any two digits separated by k places (mod n) are identical. Because $r^n(S) = S$ it follows by properties of the greatest common denominator that $r^{\gcd(k,n)}(S) = S$. Thus, because of our choice of k, we have that $\gcd(k, n) = k$ and S repeats its first k digits n/k times.

Let the string T of length k be the first k digits of S. If T is a winning string, then it follows that S is a winning string. So assume T is a losing string. Because S is non-trivial, T must also be non-trivial. By Lemma 3 there is a rotation of T, r^jT which can be reduced to a single digit using the standard same game rules. Then the first k digits of r^jS are r^jT which repeat n/k times. Now reduce each copy of r^jT to a single digit, leaving a single run of n/k consecutive identical digits. Finally, we remove this run of digits, proving that r^jS, and therefore S, is a winning string. □

Proposition 5: *Let S be a non-trivial binary n-string. Assume that $cr^kS = S$ for some integer k. Then S is a winning string.*

Proof: Let the integer k satisfy $cr^kS = S$. By applying r^k twice, it follows that $r^{2k}S = S$. Let ρ denote the remainder of $2k$ upon division by n so that $0 < \rho < n$, if $2k$ is not a multiple of n. In that case we also have that $r^\rho S = S$, hence we may apply the previous proposition. Thus we may assume that $2k$ is a multiple of n.

Notice that the first k digits of S must be the complement of the next k digits. Similarly, the next k digits must be the complement of these, meaning they are the same as the first k digits. Therefore, the first $2k$ digits of S will be repeated over the length of S. If $n > 2k$, then this pattern is repeated at least twice and by the previous result S must be a winning string. So we reduce to the case where $n = 2k$, and the pattern appears only once.

We proceed by induction on n. So assume any non-trivial string of length $j < n$ composed of a string followed by its complement is a winning string. Because S is non-trivial, we can rotate it to r^iS for some i so that the first k digits begin with a run of consecutive identical digits. Let T be the first k digits of r^iS, so that r^iS consists of T followed by cT. From our construction, T and cT begin with a run of consecutive identical digits, so we remove those runs to form a string S' of length of less than n. If S' is trivial, then r^iS must consist of a single run of 1's and a single run of 0's and so it is a winning string. Our induction hypothesis covers the other case. □

These propositions imply that the cardinality of the orbit of a losing n-string S is $2n$. This is the case since we must have $r^kS \neq S$ and $cr^kS \neq S$ for any k such that $0 < k < n$, hence the strings r^iS, cr^iS for i ranging from 0 to $n-1$ are all different, and there are $2n$ such strings. Thus all that remains is to find F_{n-2} losing orbits of Q_n, excluding the orbit of trivial strings in order to establish formula (1). We proceed by introducing a string that indexes where a string of the same game alternates. As it turns out, we can tell a losing string from the indexing string from a very simple inspection; no moves of the game are necessary.

Definition: Given a binary string $S = a_1 \cdots a_n$ of length n, we define $I(S)$ to be the *indexing string* of S as follows. For $i < n$, S is a binary n-string such that the i'th element is 1 if $a_i \neq a_{i+1}$ and 0 otherwise. For the n'th element we compare a_n and a_1.

Notice the following properties of indexing strings. First, for any binary string S, $I(S) = I(cS)$. Second, $I(rS) = rI(S)$. We also have the following proposition.

Proposition 6: *Given S, The number of* 1*'s in the indexing string $I(S)$ is even.*

Proof: The number of 1's counts the number of times that S alternates from 0 to 1 or 1 to 0. If S alternates from 0 to 1 it must at some point alternate back from 1 to 0 because we consider S as wrapping around at the end. □

We now define a third game, the *indexing same game* (or *index game*). The moves of this game are performed on finite binary strings containing an even number of 1's. Let S be an n-string in the index game with a run of $k \geq 1$ consecutive 0's (where the first digit is consecutive to the n'th). Then this run of 0's is flanked by two 1's, one on each side. We define a *indexing reduction rule* by removing the run of 0's and replacing the two flanking 1's with a single zero, producing an $(n-k-1)$-string (there is an example below). A *winning indexing string* is a string in the index game that can be reduced to a single run of $k > 1$ consecutive 0's.

Proposition 7: *Let T be a binary n-string with an even number of* 1*'s. Then $T = I(S)$ for some binary string S. T is a winning string in the index game if and only if S is a winning string in the wraparound game.*

Proof: Let T be defined as above. It is trivial to note that $T = I(S)$ for some binary n-string S. We will reduce $I(S)$ and S simultaneously using corresponding reduction rules. Notice that corresponding to a run of length k in S, we have $(k-1)$ 0's in T. If our run takes up the entire string S, then T is just 0's. Then S and $T = I(S)$ are both winning strings in their respective games. If our run ends, then it alternates to the opposite digit at both ends. Thus, we must have 1's flanking our corresponding 0's in $I(S)$. When we eliminate the run in S to create a binary $(n-k)$-string S', we bring flanking digits together to create another run. Thus, removing the $(k-1)$ 0's in $I(S)$ in the index game will reduce $T = I(S)$ to $I(S')$. If S is a winning string, then some sequence of removals takes a S to a single run of identical digits. Then some sequence of removals takes $I(S)$ to a single run of two or more 0's and the converse is also true. □

Example: The winning string $S = 11101100$ can be reduced to null in the same game by the steps

$$1110\{11\}00 \mapsto 111\{000\} \mapsto \{111\} \mapsto \emptyset.$$

In each step above the digits within brackets are simply removed. The transform $I(S) =$ 00110101 can be reduced analogously in the index game with the steps

$$001\{101\}01 \mapsto 00\{1001\} \mapsto \{000\}.$$

In the above the brackets are replaced with a single 0 until we have a run of 0's.

Notice that an indexing string composed of a single run of 1's and a single run of 0's is always a losing string in the index game and that the even indexing string consisting of all 1's corresponds to the trivial losing strings. We now present a very simple condition for telling when S is a losing string by looking at $I(S)$.

Proposition 8: *Let S be a non-trivial binary n-string so that $I(S)$ is a binary n-string that contains an even number $2m$ of 1's, where $0 \leq m < n/2$. Then S is a losing string if and only if there is a run of consecutive 1's in $I(S)$ strictly greater than m.*

Proof: ($\Leftarrow$) If a string has a run of $m+1$ or more 1's we call this a main run. So assume $I(S)$ has a main run. Notice that there are at most $(m-1)$ 1's not in the main run. Each elimination step that removes a 1 from the main run also removes a 1 that is not in the main run. Since there are more 1's in the main run than 1's outside the main run, we cannot eliminate the main run and leave a run of two or more 0's.

($\Rightarrow$) We prove this by induction on n, the length of our indexing string. Thus assume any indexing string of length $j < n$ without a main run is a winning string. Say an n-string $I(S)$ has no run of $m+1$ or more 1's where $2m$ is the number of 1's in $I(S)$. We wish to prove that $I(S)$ is a winning string. Find a run q of 1's in $I(S)$ that has the maximal number of 1's. Reduce $I(S)$ to $I(S')$ by removing a 1 from this maximal run and a 1 from some other run. Then $I(S')$ has $(2m-2)$ 1's.

Suppose (for the sake of contradiction) $I(S')$ has a run s' of 1's of length $m-1+1=m$ or greater so that $I(S')$ is a losing string. Then we did not remove from a run s in $I(S)$ to produce s'. Otherwise, s would have $m+1$ or more 1's. Then s has m 1's (it cannot have more), must be a maximal run, and must be different from q which we now know must have m 1's. Then there are only two runs because there are only $2m$ 1's. But when we removed from our maximal run, we must have removed from s as well which gives our contradiction. □

Thus given an indexing string with $2m$ 1's, we know that it is a losing string just by looking at the configuration of its 1's. It loses if it has a run of $m+1$ or more 1's and wins if it has no such run. Also notice that a losing string cannot have more than one main run because it only has $2m$ 1's.

Any losing indexing string must have at least two 1's. Thus, we can choose our losing indexing n-strings by picking an even number $2m$, $0 < m < n/2$, of 1's, making sure we have at least $m+1$ gathered together in a group and throwing the rest of the 1's anywhere else in the string. This group of $m+1$ or more 1's we once again call the main run. At this point we now present our main theorem.

Theorem 9: *The number of non-trivial losing strings of length n in the same game is $2nF_{n-2}$ for all n.*

Proof: We define an *oriented losing string* to be an indexing losing string with $2m$ 1's such that the first $m+1$ digits are all 1's and the last digit is a 0. That is, we place the main run at the beginning of the string. Let T be an oriented losing string. Then $T = I(S)$ for some losing string S in the same game. Notice that $T = I(cS)$ and $rT = I(rS)$. Thus, we use $T = I(S)$ as the unique representative of the orbit of S under G. There is a way to count these oriented losing strings that gives the Fibonacci recurrence.

Because we exclude the trivial strings, we exclude the case where an even indexing string is entirely composed of 1's. Consider all the oriented losing indexing n-strings where we have a 1 placed two slots to the right of our main run (there is a 0 separating this 1 from the end of our main group). We claim that the number of losing strings of this form is equal to the number of losing strings of length $n-2$. The one-to-one correspondence is given by removing the 1 that is two slots to the right of our main run and removing a 1 in our main run to produce a losing $(n-2)$-string. By removing a 1 both from outside and inside the main run, we ensure that the resulting $(n-2)$-string is still a losing string (that it has a main run of requisite length). The inverse of this is taking an $(n-2)$-losing string, inserting a 1 to the main run and inserting a 1 two slots to the right of the main run. The above is illustrated with the following corresponding strings,

$$1111\{1\}0\{1\}0110 \leftrightarrow 111100110.$$

The first is a losing oriented indexing 11-string, the second is its corresponding losing oriented indexing 9-string. We remove or add the bracketed digits depending on the direction of the correspondence.

It is trivial to note that the number of losing indexing n-strings with a 0 placed two slots the right of the main group is the number of losing $(n-1)$-strings. We simply add or remove that 0. Thus we have the recurrence. Notice that the number of non-trivial oriented indexing losing 2-strings is $F_0 = 0$ and there is only $F_1 = 1$ oriented indexing losing 3-string: that is 110. Thus we have our theorem. □

ACKNOWLEDGMENT

The authors would like to thank Michael Henle and Elizabeth Wilmer for their comments and advice.

REFERENCES

[1] N. J. A. Sloane. The On-line Encyclopedia of Integer Sequences, published electronically at http://www.research.att.com/~njas/sequences/, 2005.

[2] Ralf Stephan. Prove or Disprove: 100 Conjectures from the OEIS. arXiv:math.CO/0409509 v4 27 Nov 2004, published electronically at http://arxiv.org/pdf/math.CO/0409509/

AMS Classification Numbers: 05A15

✠ ✠ ✠