GENERALIZED TWO-PILE FIBONacci NIM

JIM FLANIGAN
University of California at Los Angeles, Los Angeles, CA 90024

1. INTRODUCTION

Consider a take-away game with one pile of chips. Two players alternately remove a positive number of chips from the pile. A player may remove from 1 to \( f(t) \) chips on his move, \( t \) being the number removed by his opponent on the previous move. The last player able to move wins.

In 1963, Whinihan [3] revealed winning strategies for the case when \( f(t) = 2t \), the so-called Fibonacci Nim. In 1970, Schwenk [2] solved all games for \( f \) nondecreasing and \( f(t) \geq t \forall t \). In 1977, Epp & Ferguson [1] extended the solution to the class where \( f \) is nondecreasing and \( f(1) \geq 1 \).

Recently, Ferguson solved a two-pile analogue of Fibonacci Nim. This motivated the author to investigate take-away games with more than one pile of chips. In this paper, winning strategies are presented for a class of two-pile take-away games which generalize two-pile Fibonacci Nim.

2. THE TWO-PILE GAME

Play begins with two piles containing \( m \) and \( m' \) chips and a positive integer \( \omega \). Player I selects a pile and removes from 1 to \( \omega \) chips. Suppose \( t \) chips are taken. Player II responds by taking from 1 to \( f(t) \) chips from one of the piles. We assume \( f \) is nondecreasing and \( f(t) \geq t \forall t \). The two players alternate moves in this fashion. The player who leaves both piles empty is the winner. If \( m = m' \), Player II is assured a win.

Set \( d = m' - m \). For \( \omega \geq 1 \), define \( L(m, \omega ) \) to be the least value of \( \omega \) for which Player I can win. Set \( L(m, 0) = \infty \forall m \geq 0 \). One can systematically generate a tableau of values for \( L(m, d) \). Given the position \( (m, d, \omega ) \), the player about to move can win if he can:

1. take \( t \) chips, \( 1 \leq t \leq \omega \), from the large pile, leaving the next player in position \( (m, d - t, f(t)) \) with \( f(t) < L(m, d - t) \); or
2. take \( t \) chips, \( 1 \leq t \leq \omega \), from the small pile, leaving the next player in position \( (m - t, d + t, f(t)) \) with \( f(t) < L(m - t, d + t) \).

(See Fig. 2.1.) Consequently, the tableau is governed by the functional equation

\[
L(m, d) = \min \{ t > 0 \mid f(t) < L(m, d - t) \text{ or } f(t) < L(m - t, d + t) \}
\]

subject to \( L(m, 0) = \infty \forall m \geq 0 \). Note that \( L(m, d) \leq d \forall d \geq 1 \). Dr. Ferguson has written a computer program which can quickly furnish the players with a \( 60 \times 40 \) tableau. As an illustration, Figure 2.2 gives a tableau for the two-pile game with \( f(t) = 2t \), two-pile Fibonacci Nim.
Fig. 2.1 The Tableau

Given \( f \), one can construct a strictly-increasing infinite sequence \( \langle H_k \rangle \) as follows: \( H_1 = 1 \) and for \( k \geq 1 \), \( H_{k+1} = H_k + H_j \) where \( j \) is the least integer such that \( f(H_j) \geq H_k \). For example, \( \langle H_k \rangle \) is the Fibonacci sequence when \( f(t) = 2t \), and \( H_k = 2^{k-1}, k \geq 1 \) when \( f(t) = t \). Schwenk [2] showed that each positive integer \( d \) can be represented as a unique sum of the \( H_k \)'s

\[
(2.1) \quad d = \sum_{i=1}^{s} H_{n_i} \text{ such that } f(H_{n_i}) < H_{n_i+1} \text{ for } i = 1, 2, \ldots, s - 1.
\]

Moreover, for the take-away game with a single pile of \( d = m' - 0 \) chips, Player I can win iff he can remove \( H_{n_1} \) chips from the pile (i.e., iff \( H_{n_1} \leq u \)). So for the two-pile game with one pile exhausted,

\[
(2.2) \quad L(0, d) = H_{n_1}.
\]

For the one-pile game with \( d = H_{n_1} + \cdots + H_{n_s}, s \geq 1 \), chips, \( H_{n_1} \) is the key term. It turns out that for the two-pile game where \( d = m' - m = H_{n_1} + H_{n_2} + \cdots + H_{n_s}, s \geq 1 \), \( H_{n_1} \) (when it exists) as well as \( H_{n_1} \) plays a decisive role. Denote \( n_1 = n \) and \( n_2 = n \) (when it exists) = \( m + r \). Thus, we shall write

\[
(2.3) \quad f(H_{k - \ell(k)}) \geq H_k.
\]

Note that \( \ell(1) = 0, \ell(k) \geq 0 \), and \( H_{k+1} = H_k + H_{k - \ell(k)} \) \( \forall k \geq 1 \).

In the sequel, we present winning strategies for the class of two-pile games for which \( \ell(k) \in \{0, 1\} \forall k \). We refer to such games as generalized two-pile Fibonacci Nim.

It would be nice if one could find some NASC on \( f \) such that \( \ell(k) \in \{0, 1\} \forall k \), the following partial results have been obtained:

(1) If \( f(t) < (5/2)t \forall t \), then \( \ell(k) \in \{0, 1\} \forall k \).

In particular, for \( f(t) = ct \),

(a) if \( 1 \leq c < 2 \), then \( \ell(k) = 0 \) \( \forall k \geq 1 \);

(b) if \( 2 \leq c < 5/2 \), then \( \ell(k) = 1 \) \( \forall k \geq 2 \);

(c) if \( c \geq 5/2 \), then \( \ell(3) = 2 \) or \( \ell(4) = 2 \).

(2) If \( \ell(k) \in \{0, 1\} \forall k \), then \( f(t) < 6t \forall t \).

(3) A NASC such that \( \ell(k) = 0 \) \( \forall k \) is \( f(2^k) < 2^{k+1} \forall k \geq 0 \).
(4) A NASC such that \( \&(k) = 1 \) \( \forall k \geq 2 \) is \( F_k \leq \sqrt{F_{k-1}} < F_{k+1} \) \( \forall k \geq 2 \), where \( \langle F_k \rangle \) is the Fibonacci sequence 1, 2, 3, 5, 8, 13, ...
3. SOME GOOD AND BAD MOVES

**Lemma:** For the position \((m, d)\), \(d = H_n + \cdots + H_{n_s} \geq 1\), it is never a winning move to take

1. \(t\) chips from the large pile if \(0 < t < H_n\).
2. \(t\) chips from the small pile if \(0 < t < H_n\), \(t \neq H_{n - k(n)}\).

It is always a winning move to take

3. \(H_n\) chips from the large pile with the possible exception of the special case: \(d = H_n + H_{n+2} + \cdots + H_{n_s} \geq 2\), \(k(n + 1) = k(n + 2) = 1, m \geq H_{n+1}\).
4. \(H_{n-k(n)}\) chips from the small pile when \(d = H_n + H_{n+2} + \cdots + H_{n_s}, s \geq 2\), \(k(n + 1) = k(n + 2) = 1, m \geq H_{n-k(n)}\). (This contains the special case.)

**Proof:** The statements (1)-(4) imply that \(L(m, d) \in \{H_n, H_{n-k(n)}\}\) \(\forall m \geq 0\).

We shall use this observation and double induction in our argument.

Schwenk [2] proved the assertions for the positions \((0, d)\), \(\forall d \geq 1\) (see equation 2.2). Suppose they hold for the positions \((m, d)\) \(\forall m \leq M - 1, \forall d \geq 1\) for some \(M \geq 1\). We must show that (1)-(4) hold for the positions \((M, d)\) \(\forall d \geq 1\).

The claim is trivial for position \((M, 1)\). Suppose it is true for \((M, d)\)
\(\forall d \leq D - 1\) for some \(D \geq 2\). Consider the two types of moves which can be made from position \((M, D)\), \(D = H_n + H_{n+2} + \cdots + H_{n_s}, s \geq 1\).

### A. Taking from the big pile:

Take \(t\) chips, \(0 < t < H_n\), from the big pile. Then \(D - t = H_k + \cdots + H_{n_s}\)
where \(k < n\), \(t \geq H_{k-1}\) if \(k(k) = 1\), and \(t \geq H_k\) if \(k(k) = 0\). By the inductive assumption \(L(M, D - t) \leq H_k\).
Hence,

- \(f(t) \geq f(H_{k-1}) \geq H_k \geq L(M, D - t)\) if \(k(k) = 1\),
- \(f(t) \geq f(H_k) \geq H_k \geq L(M, D - t)\) if \(k(k) = 0\).

Statement (1) follows.

Suppose you take \(t = H_n\) chips from the big pile. Consider the following cases.

1. \(D = H_n\). Taking \(H_n\) chips from the large pile is obviously a winning move.
2. \(D > H_n\). Write \(D - H_n = H_{n+2} + \cdots + H_s\).
   - \(r = 1\). Necessarily, \(k(n + 1) = 0\). By the inductive assumption, \(L(M, D - H_n) \leq H_{n+1}\). Thus, \(f(H_n) < L(M, D - H_n)\) and it is a good move to take \(H_n\) chips from the large pile.
   - \(r \geq 3\). By the inductive assumption, \(L(M, D - H_n) \leq H_{n+2}\). Thus, \(f(H_n) < H_{n+2} \leq L(M, D - H_n)\) and it is a good move to take \(H_n\) chips from the large pile.
   - \(r = 2\).
     - \(k(n + 1) = 0\). \(f(H_n) < f(H_{n+1})\) and, by the inductive assumption, \(L(M, D - H_n) \geq H_{n+1}\). A good move is to take \(H_n\) chips from the big pile.
     - \(k(n + 1) = 1\) and \(k(n + 2) = 0\). By the second equation and the inductive assumption, \(L(M, D - H_n) = H_{n+2}\). Thus, \(f(H_n) < H_{n+2} \leq L(M, D - H_n)\), so taking \(H_n\) chips from the large pile wins.
     - \(k(n + 1) = 1\) and \(k(n + 2) = 1\). Here \(f(H_n) \geq H_{n+1}\). By the inductive assumption, it is possible that \(L(M, D - H_n) = H_{n+1}\). If \(L(M, D - H_n) = H_{n+1}\), then \(M \geq H_{n+1}\) follows from (1) of the Lemma. The possibility of \(f(H_n) \geq L(M, D - H_n)\) signifies that taking \(H_n\) chips from the large pile might be a bad move. Thus, (3) holds.
B. Taking from the small pile:

If \( t \) chips, \( 0 < t < H_n, t \neq H_{n-k-1}(n) \), are removed from the small pile, the resulting position is \((M - t, D + t)\), \( D + t = H_{n-k} + \cdots + H_n \) for some \( k \geq 1 \) and \( t \geq H_{n-k} \). But \( L(M - t, D + t) \leq H_{n-k} \) by assumption. Since

\[ f(t) \geq t \geq H_{n-k} \geq L(M - t, D + t), \]

this is a bad move. Thus, (2) holds.

C. Case A2.c.(iii) revisited:

Here \( D = H_n + H_{n+2} + \cdots + H_n \), \( l(n + 1) = l(n + 2) = 1 \). Suppose taking \( H_n \) chips from the large pile is not a good move. Then, \( L(M, D - H_n) = H_{n+1} \).

For position \((M, D)\), \( M \geq H_{n-k-1}(n) \), take \( H_{n-k-1}(n) \) chips from the small pile to get \((M - H_{n-k-1}(n), D + H_{n-k-1}(n))\). \( D + H_{n-k-1}(n) = H_{n-k} + H_n + H_{n+2} + \cdots + H_{n+1} + H_{n+2} + \cdots + H_n = H_{n+k} + \cdots + H_{n+m} \) for some \( k \geq 3 \) and \( m \geq n_{n+2} \), since \( l(n + 2) = 1 \). By the inductive assumption, \( L(M - H_{n-k-1}(n), D + H_{n-k-1}(n)) \geq H_{n+2} \). But \( f(H_{n-k-1}(n)) \leq f(H_n) < H_{n+1} \). Thus,

\[ f(H_{n-k-1}(n)) < L(M - H_{n-k-1}(n), D + H_{n-k-1}(n)) \]

Taking \( H_{n-k-1}(n) \) chips from the small pile is a good move, so (4) holds.

In A, B, and C we established that (1)-(4) hold for the position \((M, D)\), which completes the induction on \( D \). Hence, they hold for \((M, d)\) \( \forall d \geq 1 \). This in turn completes the induction on \( m \). Thus, (1)-(4) hold for \((m, d)\) \( \forall m \geq 0 \), \( \forall d \geq 1 \). Q.E.D.

Corollary 1: \( L(m, d) \in \{H_n, H_{n-k-1}(n)\} \) \( \forall m \geq 0 \).

Observe that if \( l(n) = 0 \), then \( L(m, d) = H_n \) \( \forall m \geq 0 \). But when \( l(n) = 1 \), there are two possible values \( L(m, d) \) might assume. However, if \( m < H_{n-1} \), then \( L(m, d) = H_n \).

Corollary 2—How to win (if you can) when you know \( L(m, d) \):

1. If \( L(m, d) = H_{n-1} \), take \( H_{n-1} \) chips from the small pile to win.
2. If \( L(m, d) = H_n \), a winning move is to take \( H_n \) chips from the large pile, except possibly for the special case cited in the Lemma. In the special case, take \( H_n \) chips from the small pile to win.

4. HOW TO WIN IF YOU CAN

Knowing \( L(m, d) \) at the beginning of play reveals whether Player I has a winning strategy. Compare \( L(m, d) \) and \( \omega \). If Player I knows the value of \( L(m, d) \) and \( \omega \geq L(m, d) \), he can use Corollary 2 to determine a winning move.

Which of the two possible values \( L(m, d) \) assumes is not obvious under certain circumstances. The position \((m, d, \omega)\) defies immediate classification when \( L(m, d) = H_n \) is unknown and \( H_{n-1} \leq \omega < H_n \).

Fortunately, not knowing whether one can win at the beginning of play does not prevent one from describing a winning strategy, provided such a strategy exists. A strategy of play, constructed from the Corollaries, is presented in Table 4.1. This table tells how to move optimally in all situations in which there exists a possibility of winning. An \( H(d) \) represents a position for which there exists a winning move for Player I (II).

The only case in which the status of a position is now known at the start of play arises in 2(b) of the table. There, the player about to move is an optimist and pretends \( L(m, d) = H_{n-1} \). This dictates taking \( H_{n-1} \) chips from the small pile. The outcome of the game will reveal the value of \( L(m, d) \) depending on who wins.
Table 4.1. How To Win (If You Can) Without Knowing \( \ell(m, d) \)

(1) If \( \ell(n) = 0 \) [so necessarily \( L(m, d) = H_n \)] and

(a) \( d = H_n + H_{n+2} + \cdots + H_{n_s}, \ s \geq 2, \ \ell(n + 2) = \ell(n + 1) = 1 \)

\[
\begin{array}{c|c|c}
\omega \geq H_n & \text{\( L(m, d) = H_n \)} & \text{\( m \geq H_n \)}
\end{array}
\]

\[
\begin{array}{c|c|c}
\omega < H_n & \text{\( m < H_n \)}
\end{array}
\]

Table: For \( \omega \geq H_n \), play \( H_n \) from the smaller pile; for \( \omega < H_n \), play \( H_n \) from the larger pile.

(b) not as in (a)

\[
\begin{array}{c|c|c}
\omega \geq H_n & \text{\( m \geq H_n \)} & \text{\( m < H_n \)}
\end{array}
\]

\[
\begin{array}{c|c|c}
\omega < H_n & \text{\( m < H_n \)}
\end{array}
\]

(2) If \( \ell(n) = 1 \) and

(a) \( d = H_n + H_{n+2} + \cdots + H_{n_s}, \ s \geq 2, \ \ell(n + 2) = \ell(n + 1) = 1 \)

\[
\begin{array}{c|c|c}
\omega \geq H_n & \text{\( L(m, d) = H_n \)} & \text{\( m \geq H_n \)}
\end{array}
\]

\[
\begin{array}{c|c|c}
\omega < H_n & \text{\( m < H_n \)}
\end{array}
\]

Table: For \( \omega \geq H_n \), play \( H_n \) from the smaller pile; for \( \omega < H_n \), play \( H_n \) from the larger pile.

(b) not as in (a)

\[
\begin{array}{c|c|c}
\omega \geq H_n & \text{\( m \geq H_n \)} & \text{\( m < H_n \)}
\end{array}
\]

\[
\begin{array}{c|c|c}
\omega < H_n & \text{\( m < H_n \)}
\end{array}
\]

(Note: s.p. = small pile; l.p. = large pile.)

As an illustration, consider two-pile Fibonacci Nim. It was first solved by Ferguson in the form of Table 4.1. For \( f(t) = 2t \), the sequence \( \langle H_k \rangle \) is the Fibonacci sequence. The first few values are

<table>
<thead>
<tr>
<th>( k )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H_k )</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>21</td>
<td>34</td>
<td>55</td>
</tr>
</tbody>
</table>

\( \ell(1) = 0 \) and \( \ell(k) = 1 \) \( \forall k \geq 2 \), since \( H_{k+1} = H_k + H_{k-1} \) \( \forall k \geq 2 \). What is the status of position \( m = 20, d = 42, \omega = 6 \)? \( d = 34 + 8 = H_8 + H_5 \). Player I is an optimist and assumes that \( L(20, 42) = 5 \), not 8. 2(b) in the table tells him to take 5 chips from the small pile.
Player II is left in position \( m = 20 - 5 = 15, d = 42 + 5 = 47, w = f(5) = 10. \) \( d = 34 + 13 = H_8 + H_6, \) \( \delta(6) = 1, \) \( r = 2, \) \( \delta(8) = \delta(7) = 1. \) \( H_8 = 8 \leq w < H_6 = 13. \) By 2(a) of the above table, this is a winning position \( (L(15, 47) = 8). \) Player II takes 8 chips from the small pile to win. We conclude that Player I has no winning strategy for the position \( (20, 42, 6). \) Consequently, \( L(20, 42) = 8, \) not 5.

Only after playing the game for a while were we able to determine who could win.

5. ELIMINATING SUSPENSE

It turns out that the suspense which can arise when \( L(m, d) \) is unknown can be eliminated. The Theorem of this section presents a simple method for computing \( L(m, d). \) If \( d = H_n + \cdots + H_s, \) then the entries in the \( d \)th column of the tableau can assume only the values \( H_n \) and \( H_{n-1}. \) We say that the \( d \)th column of the tableau makes \( k \) flips, \( 0 \leq k \leq \infty, \) if it has the form in Figure 5.1. If \( k < \infty, \) the \( k \)th flip is followed by an infinite string of

\[
\begin{align*}
H_n's & \quad \text{if } k \text{ is even} \\
H_{n-1}'s & \quad \text{if } k \text{ is odd}.
\end{align*}
\]

<table>
<thead>
<tr>
<th>( m )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>\ldots</th>
<th>( d = H_n + \cdots + H_s, )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( H_n )</td>
<td>( H_{n-1} ) entries</td>
<td>first flip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( H_n )</td>
<td>( H_{n-1} ) entries</td>
<td>second flip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( H_n )</td>
<td>( H_{n-1} ) entries</td>
<td>third flip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 5.1 The \( d \)th Column Makes \( k \) "flips"**

**Theorem:** For \( n \geq 1, \) set \( A_n = \{ x | x \geq 0, \delta(n + x) = 0 \}. \) Then:

A. **Simple Case:** \( d = H_n. \) The \( d \)th column makes \( k \) flips, where \( k = \min A_n. \)

   (Convention: \( \min \emptyset = \infty. \))

B. **Compound Case:** \( d = H_n + H_{n+r} + \cdots + H_s, s \geq 2. \) The \( d \)th column makes \( k \) flips, where

\[
k = \begin{cases} 
r - 1 & \text{if } \min A_n > r, \\
\min A_n & \text{if } \min A_n \leq r.
\end{cases}
\]
Proof:
A. Simple Case:

(1) $A_n \neq \emptyset$. We proceed by induction on $k = \min A_n$. For $k = 0$, $L(m, H_n) = H_n^m \leq 0$, since $\mathcal{L}(n) = 0$. There are zero flips.

Suppose the result holds $\forall k \leq K - 1$ for some $K \geq 1$. (That is, if $d = H_m$ and $\min A_n \leq K - 1$, then the column for $d = H_m$ makes $\min A_m$ flips.)

By the Lemma, each entry of the column $d = H_n$ is $H_n$, unless a good move can be made by taking $H_{n-1}$ from the small pile. Removing $H_{n-1}$ chips from the small pile is a winning move for position $(M, H_n)$, $M \geq H_{n-1}$, if $f(H_{n-1}) < L(M - H_{n-1}, H_n + H_{n-1})$. Since $\mathcal{L}(n) = 1, H_n + H_{n-1} = H_{n+1}$ and $L(M - H_{n-1}, H_n + H_{n-1}) = H_{n+1}$ or $H_n$. Moreover, $H_{n+1} > f(H_{n+1}) \geq H_n$. This can be a good move iff $L(M - H_{n-1}, H_{n+1}) = H_{n+1}$. The column $d = H_{n+1}$ makes $K - 1$ flips. Thus, the column $d = H_n$ makes $K$ flips. (See Fig. 5.2.) This completes the induction on $k$.

(2) $A_n = \emptyset$, $\mathcal{L}(n+k) = 1$ and $A_{n+k} = \emptyset \forall k \geq 0$. We show that each column $d = H_{n+k}$, $k \geq 0$, makes infinitely many flips. Let us proceed by induction on $m$.

<table>
<thead>
<tr>
<th>$m$</th>
<th>0</th>
<th>1</th>
<th>\ldots</th>
<th>$H_n$</th>
<th>\ldots</th>
<th>$H_{n+1} = H_n + H_{n-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{n-1}$</td>
<td>$H_n$</td>
<td>\ldots</td>
<td>$H_{n+1}$</td>
<td>$H_n$</td>
<td>$H_{n+1}$</td>
<td>$H_n$</td>
</tr>
<tr>
<td>first flip</td>
<td>good move, $f(H_{n-1}) &lt; H_{n+1}$</td>
<td>\ldots</td>
<td>$H_n$</td>
<td>$H_n$</td>
<td>$H_{n+1}$</td>
<td>$H_n$</td>
</tr>
<tr>
<td>second flip</td>
<td>bad move, $f(H_{n-1}) \geq H_n$</td>
<td>\ldots</td>
<td>$H_n$</td>
<td>$H_n$</td>
<td>$H_{n+1}$</td>
<td>$H_n$</td>
</tr>
<tr>
<td>third flip</td>
<td>good move, $f(H_{n-1}) &lt; H_{n+1}$</td>
<td>\ldots</td>
<td>$H_n$</td>
<td>$H_n$</td>
<td>$H_{n+1}$</td>
<td>$H_n$</td>
</tr>
</tbody>
</table>

Fig. 5.2 Case A(1)

By the remark to Corollary 1, $L(m, H_{n+k}) = H_{n+k} \forall m < H_{n-1}, \forall k \geq 0$. The tableau has the desired values for the first $H_{n-1}$ entries in columns $d = H_{n+k}$, $k \geq 0$.

Suppose that the tableau assumes the desired values in the entries $m = 0, 1, \ldots, M - 1$ in the columns $d = H_{n+k}$, $k \geq 0$, for some $M \geq H_{n-1}$. One can find $k_0 \geq 0$ such that

$$H_{n-1} + H_n + \ldots + H_{n+k_0} - 1 \geq M > H_{n-1} + H_n + \ldots + H_{n+k_0} - 1.$$

Equivalently, $H_n + \ldots + H_{n+k_0} - 1 \geq M - H_{n-1}$.

Table:| $k_0$ | 0 | 1 | \ldots | $H_n$ | \ldots | $H_{n+k_0} - 1$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 &amp; $H_n + \ldots + H_{n+k_0} - 1$ if $k_0 \geq 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
By the inductive assumption,
\[ L(M - H_{n-1}, H_{n+1}) = \begin{cases} 
H_{n+1} & \text{if } k_0 \text{ is even}, \\
H_n & \text{if } k_0 \text{ is odd}.
\end{cases} \]

(See Fig. 5.3.) Thus, for the position \((M, H_n)\),

(a) if \(k_0\) is even, taking \(H_{n-1}\) chips from the small pile is a good move since \(f(H_{n-1}) < H_{n+1} = L(M - H_{n-1}, H_{n+1})\);

(b) if \(k_0\) is odd, taking \(H_{n-1}\) chips from the small pile is a bad move since \(f(H_{n-1}) \geq H_n = L(M - H_{n-1}, H_{n+1})\).

As desired, we conclude
\[ L(M, H_n) = \begin{cases} 
H_n & \text{if } k_0 \text{ is odd}, \\
H_{n-1} & \text{if } k_0 \text{ is even}.
\end{cases} \]

An identical argument reveals that the entries \(L(M, H_{n+k}), k > 0\), have the desired values. Thus, the row \(m = M\) assumes the desired values in the entries corresponding to columns \(d = H_{n+k}, k \geq 0\). This completes the induction on \(m\).

<table>
<thead>
<tr>
<th>(m)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>(\ldots)</th>
<th>(H_n)</th>
<th>(\ldots)</th>
<th>(H_{n+1} = H_n + H_{n-1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H_{n-1})</td>
<td>(H_n)</td>
<td>(H_n)</td>
<td>(H_{n-1})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
</tr>
<tr>
<td>(H_n)</td>
<td>(H_n)</td>
<td>(H_n)</td>
<td>(H_{n-1})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
</tr>
<tr>
<td>(k_0) even</td>
<td>(H_n)</td>
<td>(H_n)</td>
<td>(H_{n-1})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
</tr>
<tr>
<td>(H_{n+k_0})</td>
<td>(L(M, H_n))</td>
<td>(L(M, H_n))</td>
<td>(H_{n+1})</td>
<td>(H_{n+1})</td>
<td>(H_{n+k_0})</td>
<td>(\ldots)</td>
<td></td>
</tr>
<tr>
<td>(k_0) odd</td>
<td>(H_n)</td>
<td>(H_n)</td>
<td>(H_{n-1})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
</tr>
<tr>
<td>(H_{n+k_0})</td>
<td>(L(M, H_n))</td>
<td>(L(M, H_n))</td>
<td>(H_{n+k_0})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5.3 Case A(2)
8. Compound Case:  

Suppose \( \lambda(n) = 0 \). Then \( L(m, d) = H_n \ \forall m \geq 0 \). There are no flips in the \( d \)th column. Note that \( \min A_n = 0 \).  

If \( \lambda(n) = 1 \), we consider two cases:  

(1) \( k = \min A_n \leq r \). By Corollary 1, \( L(m, d - H_n + H_{n+k}) \geq H_{n+k} \ \forall m \geq 0 \).  

The tableau from column \( d - H_n + 1 \) to column \( d - H_n + H_{n+k} - 1 \), inclusive, is a copy of the tableau from column 1 to column \( H_{n+k} - 1 \), inclusive. The \( d \)th column is identical to the \( H_n \)th column. By Part A, the latter column makes \( k \) flips, \( k = \min A_n \).  

(2) \( \min A_n > r \). Here \( \lambda(n) = \lambda(n+1) = \cdots = \lambda(n+r) = 1 \). Necessarily, \( r > 1 \). Let \( d' = d - H_n + H_{n+r-1} \). Since \( \lambda(n+r) = 1 \), \( d' \) has the form \( d' = H_{n+r+u} + \cdots + H_{n+u} \), for some \( u \geq 1 \) and \( n_u \geq n \). By Corollary 1, \( L(m, d') \geq H_{n+r+u} \). Consider the position \( (m, d'') \), \( m \geq H_{n+r-3} \), where \( d'' = d - H_n + H_{n+r-2} \). \( \lambda(n+r) = 1 \). It is a good move to take \( H_{n+r-3} \) chips from the small pile, since \( f(H_{n+r-3}) = L(m, d') \). Thus, \( L(m, d'') = H_{n+r-3} \). The column \( d'' \) makes one flip. Since the column \( d'' \) makes one flip, argue as in Part A(1) of the proof that column \( d''' = d - H_n + H_{n+r-4} \) makes two flips. Similarly, column \( d'''' = d - H_n + H_{n+r-4} \) makes three flips. Continue and argue that column \( d = d - H_n + H_n \) makes \( r-1 \) flips. (See Fig. 5.4). Q.E.D.

<table>
<thead>
<tr>
<th>( d )</th>
<th>( m )</th>
<th>( 1 )</th>
<th>( \cdots )</th>
<th>( d )</th>
<th>( \cdots )</th>
<th>( d'' = \cdots )</th>
<th>( d''' = \cdots )</th>
<th>( d'''' = \cdots )</th>
<th>( d' = )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t + H_{n+r-3} )</td>
<td>( t + H_{n+r-4} )</td>
<td>( t + H_{n+r-3} )</td>
<td>( t + H_{n+r-2} )</td>
<td>( t + H_{n+r-1} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 5.4 Case B(2)**

Notation:  

- \( \cdots \) = a good move.  
- \( \cdots \) = a bad move.
6. TWO-PILE FIBONACCI NIM REVISITED

Ferguson's solution for two-pile Fibonacci Nim was in the form of Table 4.1. His solution does not necessarily reveal which player can win at the beginning of play, because \( L(m, d) \) might not be known then. The Theorem tells us the value of \( L(m, d) \) by revealing the behavior of the columns of the tableau. Knowing \( L(m, d) \) at the start of play leaves no uncertainty as to who can win. As an illustration of the Theorem, we compute \( L(m, d) \) for two-pile Fibonacci Nim.

Suppose \( d = H_n \) for some \( n \). If \( d = H_1 \), then \( L(m, d) = H_1 = 1 \forall m \geq 0 \), since \( \lambda(1) = 0 \). If \( n \geq 2 \), the \( d \)th column makes infinitely many flips, since \( A_n = \emptyset \). For a particular value of \( m \), find the least integer \( k_0 \geq -1 \) such that \( H_{n-1} + H_{n} + \cdots + H_{n+k_0} - 1 \geq m \). Then,

\[
L(m, d) = \begin{cases} 
H_n & \text{if } k_0 \text{ is odd,} \\
H_{n-1} & \text{if } k_0 \text{ is even.}
\end{cases}
\]

Suppose \( d \) has compound form \( d = H_n + H_{n+r} + \cdots + H_s \), \( s \geq 2 \). Note that \( r > 1 \). If \( n = 1 \), the \( d \)th column of the tableau has each entry equal to 1. If \( n \geq 2 \), the \( d \)th column makes \( r - 1 \) flips. If \( k_0 \) is the least integer such that \( k_0 \geq -1 \) and \( H_{n-1} + H_{n} + \cdots + H_{n+k_0} - 1 \geq m \), then

\[
L(m, d) = \begin{cases} 
H_n & \text{if } k_0 \text{ is odd and } k_0 \leq r - 2, \text{ or}
\end{cases}
\]

\[
L(m, d) = \begin{cases} 
H_{n-1} & \text{if } k_0 \text{ is even and } k_0 \leq r - 2, \text{ or}
\end{cases}
\]

\[
L(m, d) = \begin{cases} 
r \text{ is odd and } k_0 > r - 2. 
\end{cases}
\]

\[
L(m, d) = \begin{cases} 
r \text{ is even and } k_0 > r - 2.
\end{cases}
\]

7. CONCLUSION

The function \( \lambda(k) \) was defined by (2.3). In Table 4.1, a winning strategy (provided one exists) is given for the class of two-pile take-away games in which \( \lambda(k) \in \{0, 1\} \forall k \geq 1 \). By revealing \( L(m, d) \), the Theorem enables us to determine at the beginning of play whether such a strategy exists for the player to move.

The author has considered several particular two-pile take-away games in which \( \lambda(k) \) assumes values other than 0 and 1. For example, when \( f(t) = 3t \), then \( \lambda(k) = 3 \forall k \geq 5 \). I have found no general solution for any such game. Can we find solutions for the general class of games which impose no restrictions on \( \lambda(k) \)? Can we extend to games beginning with arbitrarily many piles of chips? Let me know if you can.

REFERENCES


****