The Sprague–Grundy function for Wythoff's game: On the location of the g-values

M.Sc. Thesis

Gabriel Nivasch
gabriel.nivasch@weizmann.ac.il

Advisor: Aviezri S. Fraenkel

Weizmann Institute of Science
Rehovot 76100, Israel

August 25, 2004

Abstract

The Sprague–Grundy function of Wythoff's game has been described as "chaotic" [1]. Nevertheless, people have succeeded in proving several properties about it, shedding some light on its behavior. However, it is still unknown whether this function has a simple closed formula, or whether it can at least be calculated in time polynomial in the size of the input.

This Thesis contains two main contributions on Wythoff's Grundy function $\mathcal{G}$. The first one is a proof that for every integer $g \geq 0$, the g-values of $\mathcal{G}$ are within a bounded distance to their corresponding 0-values. Since the 0-values are located roughly along two diagonals, of slopes ϕ and ϕ^{-1}, the g-values are contained within two strips of bounded width around those diagonals. This is a generalization of a result by Blass and Fraenkel [3] regarding the 1-values.

Our second contribution is a *convergence* conjecture and an accompanying recursive algorithm. We show that for every g for which a certain conjecture is true, there exists a recursive algorithm for finding the n-th g-value in time $O(\log n)$. Our algorithm and conjecture are modifications of a similar result for the 1-values presented in [3]. We also present experimental results that seem to support our conjecture for small g.

Acknowledgements

I would primarily like to thank my advisor, Aviezri Fraenkel, for suggesting the topic for this Thesis and for many useful conversations we had together.

I also had several valuable conversations with Uriel Feige, for which I owe him thanks. I also discussed some ideas with several other people, including Nir Piterman and fellow students Dalia Krieger, Daniel Reichman, Eyal Lev, and Ronen Gradwohl.

Last but not least, I would like to thank Achim Flammenkamp for his extensive comments on a draft of this Thesis, which led to significant improvements in the presentation, and for independently checking most of the experimental data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wythoff's game

In this Thesis we consider the following two-player game, known as *Wythoff's game* [21]:

There are two piles of tokens, of sizes $x, y \geq 0$. Two players take turns. On each turn, a player either removes an arbitrary number of tokens from one pile, or the same number of tokens from both piles. The player who takes the last token wins.

For example, the game could start from position $(7, 15)$. The first player might take 2 tokens from the first pile, reaching position $(5, 15)$. Then the second player could take 5 tokens from each pile, reaching position $(0, 10)$. Then the first player can take the entire second pile, reaching position $(0, 0)$ and winning.

Wythoff's game can be represented graphically with a quarter-infinite chessboard, extending to infinity upwards and to the right (Figure 1.1). We number the rows and columns sequentially $0, 1, 2, \ldots$. A chess queen is placed in some cell of the board. On each turn, a player moves the queen to some other cell, except that the queen can only move left, down, or diagonally down-left. The player who takes the queen to the corner wins.

There is an unfortunate disagreement among mathematicians over whether the natural numbers should include 0 or not [20]. Throughout this Thesis we will let $\mathbb{N} = \{0, 1, 2, \ldots\}$.

1.2 The Sprague–Grundy theory of impartial games

We now review the Sprague–Grundy theory of acyclic impartial games [1, 5] (or *impartial games* for short, since we will not consider cyclic games).

An impartial game is a game for two players in which each player has complete information, there is no chance, and from each position the legal moves are the same for both players. (This last requirement excludes, for example, the

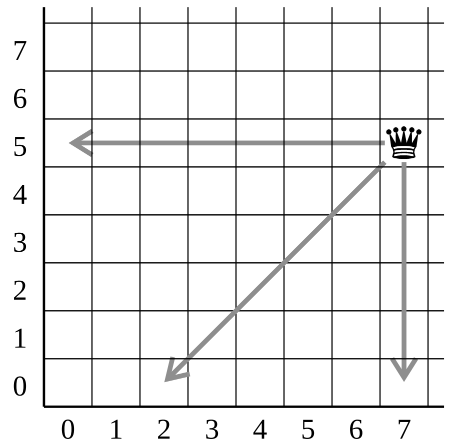


Figure 1.1: Graphic representation of Wythoff's game.

game of tic-tac-toe, in which one player plays the X's and the other plays the O's.)

Formally, an impartial game can be represented by a directed acyclic graph $G = (V, E)$. Each vertex in G represents a position in the game, and each directed edge indicates a legal move from one position to another. We call such a graph a *game-graph.*

A token is initially placed on some vertex $v \in V$. The two players take turns sliding the token from its current vertex to one of the vertex's followers. The player who moves the token into a leaf is the winner (a *leaf* is a vertex with no followers).

(These are the rules for *normal play.* In *misère play* the winning rule is the opposite—the last player to move is the loser. However, we will only consider normal play in this Thesis.)

We need to make some restrictions on the game-graph G in order to exclude pathological cases. A vertex w is said to be a *descendant* of vertex v if there exists a directed path from v to w. Then:

Definition 1.1 A game-graph G is *locally path-bounded* if each vertex in G has a finite number of descendants.

(We are using terminology from [9].)

It is easy to see that if G is locally path-bounded, then a game starting from any position must end in a finite number of moves.

Let us look, for example, at the game-graph of Wythoff's game. It is given by $G = (V, E)$ where $V = \mathbb{N} \times \mathbb{N}$ and

$$\begin{aligned} E \;=\; & \big\{ \big((x, y), (x', y)\big) \bigm| y \geq 0,\, x > x' \geq 0 \big\} \;\cup \\ & \big\{ \big((x, y), (x, y')\big) \bigm| x \geq 0,\, y > y' \geq 0 \big\} \;\cup \\ & \big\{ \big((x, y), (x - k, y - k)\big) \bigm| k > 0,\, x \geq k,\, y \geq k \big\}. \end{aligned} \tag{1.1}$$

It is easy to check that G is locally path-bounded. In fact, vertex (x, y) has exactly $xy + x + y$ descendants.

From now on, we will only consider locally path-bounded game-graphs.

1.2.1 P- and N-positions

It is intuitively clear that from any given position in a game G, if both players play as well as they can, then the game is an assured victory either for the first player or for the second player. If a position is a win for the first player, it is called an *N-position* (for *next*), and if it is a win for the second player, it is called a *P-position* (for *previous*).

Formally, we define the P- and N-positions inductively as follows:

Definition 1.2 A vertex $v \in V$ is a *P-position* if all its followers are N-positions. Otherwise (if v has a follower P-position), v is an *N-position*.

Note that this definition implies that the leaves of G are P-positions.

Lemma 1.3 *A game starting from a vertex v is a win for the first player if and only if v is an N-position. Otherwise, the game is a win for the second player.*

Proof By induction. The base cases are the leaves, which are P-positions, and the first player loses because he cannot even play. Now, suppose the claim has been verified for all of v's followers. Then, if v is an N-position, the first player can move into a P-position and win. And if v is a P-position, the first player is forced to move into an N-position, allowing the second player to win. Therefore, the claim is true also for v.

Since the game-graph is locally path-bounded, this induction will reach every vertex of the graph. ∎

Therefore, knowledge of the P- and N-positions provides the winning strategy for an impartial game: If it is our turn and the game is in an N-position, we should move into a P-position. Then our opponent will be forced to move into an N-position, and so on.

1.2.2 Sums of games; the Sprague–Grundy function

Given games G_1 and G_2, we define their *sum* $G_{\text{sum}} = G_1 + G_2$ as follows: A position in G_{sum} is a pair of positions $v_1 v_2$ where $v_1 \in G_1$ and $v_2 \in G_2$. On each turn, a player chooses between G_1 and G_2 and plays on it, leaving the other game untouched. The game ends when no moves are possible neither on G_1 nor on G_2. Formally:

Definition 1.4 Given game-graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their sum $G_{\text{sum}} = (V_{\text{sum}}, E_{\text{sum}})$ is given by

$$\begin{aligned} V_{\text{sum}} &= V_1 \times V_2, \\ E_{\text{sum}} &= \big\{(v_1 v_2, w_1 v_2) \mid (v_1, w_1) \in E_1\big\} \cup \\ &\quad \big\{(v_1 v_2, v_1 w_2) \mid (v_2, w_2) \in E_2\big\}. \end{aligned} \tag{1.2}$$

Similarly, we can define the sum of any number of games, since the sum operation on two games is associative.

Now, suppose we are given games G_1 and G_2. How can we correctly play their sum? Knowledge of the P- and N-positions in G_1 and G_2 is not enough. The sum of two P-positions is always a P-position, and the sum of a P-position and an N-position is always an N-position. But it can be easily verified that the sum of two N-positions could either be a P- or an N-position.

To play correctly sums of games we need a generalization of the notion of P- and N-positions, which is known as the *Sprague–Grundy function*. Recall that $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Definition 1.5 Given a proper subset of the natural numbers $S \subsetneq \mathbb{N}$, the *minimum excluded value* of S is

$$\operatorname{mex} S = \min\left(\mathbb{N} \setminus S\right), \tag{1.3}$$

the smallest natural number not in S.

Definition 1.6 Given a game-graph $G = (V, E)$, its *Sprague–Grundy function* (or just *Grundy function*) $\mathcal{G} : V \to \mathbb{N}$ is defined recursively by

$$\mathcal{G}(v) = \operatorname{mex} \bigl\{ g(w) \bigm| (v, w) \in E \bigr\}. \tag{1.4}$$

In words, the Grundy value of a vertex v is the smallest nonnegative integer that does not appear among the Grundy values of v's followers. In particular, the leaves of G have Grundy value 0.

The Grundy function gives us the P- and N-positions as follows:

Lemma 1.7 *Vertex v is a P-position if and only if $\mathcal{G}(v) = 0$.*

Proof By induction. The leaves are P-positions and have Grundy value 0. Suppose the claim is true for all of v's followers. If $\mathcal{G}(v) = 0$, then all of v's followers have nonzero Grundy value, so they are N-positions. Therefore, v is a P-position. And if $\mathcal{G}(v) > 0$, then v must have a follower with Grundy value 0, which is a P-position. Therefore, v is an N-position. In both cases, the claim is also true for v.

Again, the fact that the graph is locally path-bounded ensures that this induction reaches every vertex. ∎

Now we show how the Grundy function enables us to play the sum of two or more games.

Definition 1.8 Given $a, b \in \mathbb{N}$, we denote by $a \oplus b$ their *nim-sum*, which is their bitwise XOR when a and b are written in binary.

For example, $3 \oplus 5 = 011_2 \oplus 101_2 = 110_2 = 6$, and $3 \oplus 6 = 5$ and $5 \oplus 6 = 3$. The nim-sum is clearly a commutative and associative operation.

Theorem 1.9 *Let G_1 and G_2 be two game-graphs, and let $v = v_1v_2$ be a vertex in their sum $G_{\rm sum} = G_1 + G_2$. Then*

$$\mathcal{G}(v) = \mathcal{G}(v_1) \oplus \mathcal{G}(v_2). \tag{1.5}$$

Proof Let $\mathcal{H} : V_{\rm sum} \to \mathbb{N}$ be the function given by $\mathcal{H}(v) = \mathcal{G}(v_1) \oplus \mathcal{G}(v_2)$ for all $v = v_1v_2 \in V_{\rm sum}$. We will show that $\mathcal{H}$ satisfies Definition 1.6, so $\mathcal{H}$ is the Grundy function of $G_{\rm sum}$.

Let $a_1 = \mathcal{G}(v_1)$, $a_2 = \mathcal{G}(v_2)$, $a = \mathcal{H}(v) = a_1 \oplus a_2$. We need to show that v has a follower w with $\mathcal{G}(w) = b$ for every $b < a$, but v has no follower w with $\mathcal{G}(w) = a$.

Let b be an integer $< a$. Write a and b in binary, and consider the most-significant position in which a and b differ. Since $a > b$, that position must contain a 1 in a and a 0 in b. Therefore, without loss of generality, that position contains a 1 in a_1 and a 0 in a_2. Consider which digits we have to flip in a_1 in order to transform it into b_1, so that $b_1 \oplus a_2 = b$. Those flipped digits are uniquely determined, and furthermore, the most-significant flip is from a 1 to a 0. Therefore, no matter what the other flips are, we have $b_1 < a_1$. Therefore, by equation (1.4), vertex v_1 must have a follower w_1 with $\mathcal{G}(w_1) = b_1$. Thus, v has a follower w with $\mathcal{H}(w) = b$.

Now, let w be any follower of v. Without loss of generality, $w = w_1v_2$, where w_1 is a follower of v_1; so by equation (1.4), we have $\mathcal{G}(w_1) \neq \mathcal{G}(v_1)$. Therefore, $\mathcal{H}(w) \neq \mathcal{H}(v)$. ∎

Theorem 1.9 clearly generalizes to the sum of any number of games. Therefore, knowledge of the Grundy function provides a winning strategy for the sum of games: If it is our turn and the games' Grundy values have nonzero nim-sum, we should move so as to make the nim-sum zero. Then our opponent will be forced to make the nim-sum nonzero, and so on.

The following lemma gives some basic bounds on the Grundy function.

Lemma 1.10 *Given a vertex v, let n_v be the number of followers of v, and let p_v be the number of edges in the longest path from v to a leaf. Then $\mathcal{G}(v) \leq n_v$ and $\mathcal{G}(v) \leq p_v$.*

Proof The first bound follows trivially from equation (1.4). The second bound follows by induction and from equation (1.4). ∎

Finally, the following lemma shows that the Grundy function of a game-graph can be calculated up to a certain value g using the mex property.

Lemma 1.11 *Given a game-graph $G = (V, E)$ and an integer $g \geq 0$, let $\mathcal{H}$ be a function $\mathcal{H} : V \to \{0, \ldots, g, \infty\}$ such that for all $v \in V$,*

1. *if $\mathcal{H}(v) \leq g$ then*

$$\mathcal{H}(v) = \operatorname{mex}\{\mathcal{H}(w) \mid (v, w) \in E\};$$

2. *if* $\mathcal{H}(v) = \infty$ *then*

$$\text{mex}\,\{\mathcal{H}(w) \mid (v,w) \in E\} > g.$$

Then $\mathcal{G}(v) = \mathcal{H}(v)$ *whenever* $\mathcal{H}(v) \le g$*, and* $\mathcal{G}(v) > g$ *whenever* $\mathcal{H}(v) = \infty$*.*

In the function $\mathcal{H}$, the labels ∞ are placeholders that indicate values larger than g.

Proof By induction. The claim is true if v is a leaf, since $\text{mex}\,\emptyset = 0$, so we must have $\mathcal{H}(v) = 0$ to satisfy the assumptions on $\mathcal{H}$. Therefore $\mathcal{G}(v) = \mathcal{H}(v) = 0$ as required.

Now suppose the claim is true for all the followers of some vertex v. Then:

- If $\mathcal{H}(v) = h \le g$, then v must have followers with $\mathcal{H}$-values $0, \ldots, h-1$, but not with value h. Therefore, by the induction hypothesis, v has followers with $\mathcal{G}$-values $0, \ldots, h-1$, but not with value h. Therefore, $\mathcal{G}(v) = \mathcal{H}(v) = h$.

- If $\mathcal{H}(v) = \infty$, then v must have followers with $\mathcal{H}$-values $0, \ldots, g$. Therefore, by the induction hypothesis, those are also the $\mathcal{G}$ values of the followers. Therefore, $\mathcal{G}(v) > g$.

In either case, the claim is also true for v itself. ∎

1.2.3 The game of Nim

The most fundamental impartial game is the *Nim pile*. A Nim pile consists of a certain number of tokens. On each turn, a player removes from the pile any number of tokens between one token and the entire pile. The player who empties the pile wins.

This game by itself is of course trivial: The first player can win immediately by taking the entire pile. But if we add together Nim piles of various sizes, we get the well-known game of *Nim*. The Grundy value of a Nim pile of size n is n. Therefore, the Grundy value of a position in Nim is the nim-sum of its pile sizes.

1.3 Previous results on Wythoff's Grundy function

From now on, $\mathcal{G}$ will denote specifically the Grundy function of Wythoff's game. Thus, $\mathcal{G} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is given by

$$\begin{aligned} \mathcal{G}(x,y) = \text{mex}\Big(\quad & \{\mathcal{G}(x',y) \mid 0 \le x' < x\} \;\cup \\ & \{\mathcal{G}(x,y') \mid 0 \le y' < y\} \;\cup \\ & \{\mathcal{G}(x-k,y-k) \mid 1 \le k \le \min(x,y)\} \quad \Big), \end{aligned} \tag{1.6}$$

15	15	16	17	18	10	13	12	19	14	0	3	21	22	8	23	20
14	14	12	13	16	15	17	18	10	9	1	2	20	21	7	11	23
13	13	14	12	11	16	15	17	2	0	5	6	19	20	9	7	8
12	12	13	14	15	11	9	16	17	18	19	7	8	10	20	21	22
11	11	9	10	7	12	14	2	13	17	6	18	15	8	19	20	21
10	10	11	9	8	13	12	0	15	16	17	14	18	7	6	2	3
9	9	10	11	12	8	7	13	14	15	16	17	6	19	5	1	0
8	8	6	7	10	1	2	5	3	4	15	16	17	18	0	9	14
7	7	8	6	9	0	1	4	5	3	14	15	13	17	2	10	19
6	6	7	8	1	9	10	3	4	5	13	0	2	16	17	18	12
5	5	3	4	0	6	8	10	1	2	7	12	14	9	15	17	13
4	4	5	3	2	7	6	9	0	1	8	13	12	11	16	15	10
3	3	4	5	6	2	0	1	9	10	12	8	7	15	11	16	18
2	2	0	1	5	3	4	8	6	7	11	9	10	14	12	13	17
1	1	2	0	4	5	3	7	8	6	10	11	9	13	14	12	16
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 1.1: The Grundy function of Wythoff's game.

where mex is defined in Definition 1.5.

The main open problem for Wythoff's game is to compute its Grundy function in time polynomial in the size of the input. Specifically, given integers $x, y \geq 0$, we want to compute $\mathcal{G}(x, y)$ in time $O((\log x + \log y)^c)$ for some constant c. Even better would be a simple closed formula for $\mathcal{G}(x, y)$ in terms of x and y.

Table 1.1 shows the value of $\mathcal{G}$ for small x and y. This matrix looks quite chaotic at first glance, as has been pointed out before [1]. Nevertheless, as we will see, several results on $\mathcal{G}$ have been established. We proceed with an overview of these results.

1.3.1 Conventions and terminology

Let us first establish some conventions and terminology.

A pair (x, y) is also called a *point* or a *cell*. If $\mathcal{G}(x, y) = g$, we call (x, y) a *g-point* or a *g-value*.

Note that by symmetry, $\mathcal{G}(x, y) = \mathcal{G}(y, x)$ for all x, y. We refer to this property as *diagonal symmetry*.

We will consistently use the following graphical representation of $\mathcal{G}$: The first coordinate of $\mathcal{G}$ is plotted vertically, increasing upwards, and the second coordinate is plotted horizontally, increasing to the right.

Thus, we call *row* r the set of points (r, x) for all $x \geq 0$, and *column* c the set of points (x, c) for all $x \geq 0$. Also, *diagonal* e is the set of points $(x, x + e)$

for all x if $e \geq 0$, or the set of points $(x-e, x)$ for all x if $e < 0$. (Note that we only consider diagonals parallel to the movement of the queen.)

1.3.2 Rows, columns, and diagonals of $\mathcal{G}$

It follows directly from formula (1.6) that no row, column, or diagonal of $\mathcal{G}$ contains any g-value more than once. In fact, it is not hard to show that every row and column of $\mathcal{G}$ contains every g-value exactly once [3, 13]. Furthermore, every diagonal contains every g-value exactly once [3], although this is somewhat harder to show.

We will derive these results later on.

1.3.3 A closed formula for the P-positions

The P-positions are the zeros of the $\mathcal{G}$ function. Wythoff himself [21] already found a simple, closed formula for the P-positions. Let $\phi = (1+\sqrt{5})/2$ be the golden ratio; note that $\phi^2 = \phi + 1$ and $\phi^{-1} = \phi - 1$. Then:

Theorem 1.12 *The zeros of $\mathcal{G}$ are given by*

$$(\lfloor \phi n \rfloor, \lfloor \phi^2 n \rfloor) \quad \text{and} \quad (\lfloor \phi^2 n \rfloor, \lfloor \phi n \rfloor), \tag{1.7}$$

for $n = 0, 1, 2, \ldots$.

Proof First let us examine, in general, how to find iteratively the P-positions of a game-graph. Given game-graph G, we start by labelling its leaves as P-positions. Then, we remove from G all vertices that have a leaf as a follower (since they are N-positions). This creates new leaves on the graph, which are again labelled as P-positions (since all their followers are N-positions already removed), and so on.

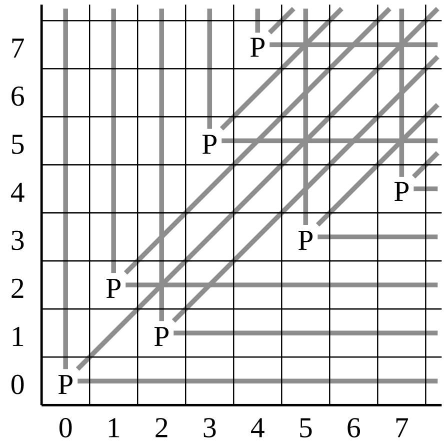
Figure 1.2 illustrates this technique applied to Wythoff's game. After doing this procedure for a while, we notice a pattern. The P-positions are given by (a_n, b_n) and (b_n, a_n) for $n = 0, 1, 2, \ldots$, where

- $a_n = \text{mex}\,\{a_m, b_m \mid m < n\}$ (a_n is the smallest integer that has not appeared so far), and
- $b_n = a_n + n$.

We can prove formally that this is indeed the case:

Lemma 1.13 *The P-positions of Wythoff are (a_n, b_n) and (b_n, a_n) for $n \geq 0$, where a_n and b_n are as given above.*

Proof Let $\mathcal{P}' = \big\{(a_n, b_n), (b_n, a_n) \bigm| n \geq 0\big\}$. We have to show that no position in $\mathcal{P}'$ has a follower in $\mathcal{P}'$, and that every position *not* in $\mathcal{P}'$ has a follower in $\mathcal{P}'$. This implies that $\mathcal{P}'$ is the unique set of P-positions.



a_n	b_n
0	0
1	2
3	5
4	7
6	10
8	13
9	15
11	18
12	20
14	23
16	26
17	28

Figure 1.2: The P-positions of Wythoff's game.

Note that $\{a_n\}$ and $\{b_n\}$ are increasing sequences and that $b_n > a_n$ for all $n > 0$. Suppose that we are in position $(a_n, b_n) \in \mathcal{P}'$, $n > 0$. Then, subtracting from the first coordinate clearly does not lead to another position in $\mathcal{P}'$. And subtracting from the second coordinate can only help if the second coordinate becomes smaller than the first, reaching some position (a_n, a_m) where $a_n = b_m$. But there is no case of $a_n = b_m$ for $n > 0$, so this cannot happen. And subtracting from both coordinates simultaneously cannot help either, because subtracting from both coordinates preserves the difference between the coordinates, and every position in $\mathcal{P}'$ has a unique difference between coordinates.

Now, suppose we are in position $(x, y) \notin \mathcal{P}'$, $x \leq y$. If $x = b_n$ for some n, then we can move to $(x, a_n) \in \mathcal{P}'$. And if $x = a_n$ for some n, then:

- if $y > b_n$, then we can move to $(x, b_n) \in \mathcal{P}'$;
- if $x \leq y \leq x + (n-1) = b_n - 1$, then let $m = y - x < n$; we can subtract from both coordinates and move to $(a_m, b_m) \in \mathcal{P}'$. ∎

Now we use the following well-known theorem:

Theorem 1.14 (Beatty [2]) *Let $\alpha, \beta < 1$ be irrational numbers such that $\alpha^{-1} + \beta^{-1} = 1$. Then the sequences $\{\lfloor \alpha n \rfloor\}_{n=1}^{\infty}$ and $\{\lfloor \beta n \rfloor\}_{n=1}^{\infty}$ together contain every positive integer exactly once.*

We prove this theorem in Appendix A.

Note that the irrationals ϕ and ϕ^2 satisfy the condition of Theorem 1.14. Let $a'_n = \lfloor \phi n \rfloor$ and $b'_n = \lfloor \phi^2 n \rfloor$ for $n \geq 0$. Then we must have

$$a'_n = \text{mex}\,\{a'_m, b'_m \mid 0 \leq m < n\}$$

for every $n \geq 0$, since otherwise an integer would either be repeated or missing in the Beatty sequences $\{a'_n\}$ and $\{b'_n\}$. And furthermore,

$$\begin{aligned} b'_n - a'_n = \lfloor \phi^2 n \rfloor - \lfloor \phi n \rfloor &= \lfloor \phi n + n \rfloor - \lfloor \phi n \rfloor \\ &= \lfloor \phi n \rfloor + n - \lfloor \phi n \rfloor \\ &= n; \end{aligned}$$

so $a'_n = a_n$ and $b'_n = b_n$. ∎

We end this topic by pointing out that there is a technique for finding a winning move given position (x, y), which involves writing x and y in so-called *Fibonacci representation*; see [18].

1.3.4 Additive periodicity

Another important result on Wythoff's game concerns the additive periodicity of $\mathcal{G}$.

Definition 1.15 A sequence $\{a_i\}$ is *additively periodic* if there exist integers $n \geq 0$ and $p \geq 1$ such that $a_{i+p} = a_i + p$ for all $i \geq n$.

Theorem 1.16 *Every row in $\mathcal{G}$ is additively periodic.*

This result was first proven by Norbert Pink in his doctoral thesis [17] (published in [8]). Landman [13] later found a simpler proof. He let $\mathcal{H}_x(y) = \mathcal{G}(x, y) - y$ and showed that for every row x, $|\mathcal{H}_x(y)|$ is bounded independently of y. Furthermore, for every row x the sequence $\mathcal{H}_y(x)$ can be computed by a finite-state machine. Therefore, $\mathcal{H}_x(y)$ is periodic, so row x is additively periodic.

Both [8] and [13] derive an upper bound of $2^{O(x^2)}$ for the pre-period and the period of row x.

Theorem 1.16 implies that for fixed x, we can compute $\mathcal{G}(x, y)$ in time linear in $\log y$, as follows: Given x, we first find the pre-period length n and the period length p of row x, knowing that both n and p are bounded by some $M_x = 2^{O(x^2)}$. This can be done in $O(M_x)$ steps of the finite-state machine, each of which takes time polynomial in x. Therefore, n and p can be found in time $O(\text{poly}(x) M_x) = 2^{O(x^2)}$.

By additive periodicity, for $y \geq n$ we have $\mathcal{G}(x, y) = \mathcal{G}(x, y') + kp$, where

$$y' = n + (y - n) \bmod p \qquad \text{and} \qquad k = \left\lfloor \frac{y - n}{p} \right\rfloor. \tag{1.8}$$

The multiplication or division of an s-bit number by a t-bit number can be done in time $O(st)$. Therefore, $\mathcal{G}(x, y)$ can be computed in time

$$2^{O(x^2)} + O(\log y \log M_x) = 2^{O(x^2)} + O(x^2 \log y), \tag{1.9}$$

which is linear in $\log y$ for constant x.

1.3.5 The 1-values of $\mathcal{G}$

Blass and Fraenkel [3] obtained several results on the 1-values of $\mathcal{G}$. Let us order the 1-values that lie on or to the right of the main diagonal by increasing row number. Formally, let

$$T_1 = \big\langle (a_0^1, b_0^1), (a_1^1, b_1^1), (a_2^1, b_2^1), \ldots \big\rangle \tag{1.10}$$

be the sequence of 1-values having $a_n^1 \le b_n^1$, ordered by increasing a_n^1. Similarly, let

$$T_0 = \big\langle (a_0^0, b_0^0), (a_1^0, b_1^0), (a_2^0, b_2^0), \ldots \big\rangle, \tag{1.11}$$

where $(a_n^0, b_n^0) = (\lfloor \phi n \rfloor, \lfloor \phi^2 n \rfloor)$, be the sequence of 0-values, as in Subsection 1.3.3.

The authors show in [3] that the n-th 1-value is within a bounded distance to the n-th 0-value. Specifically,

$$\begin{aligned} 8 - 6\phi \ &< \ a_n^1 - \phi n \ < \ 6 - 3\phi\,, \\ -3\phi \ &< \ b_n^1 - \phi^2 n \ < \ 8 - 3\phi \end{aligned} \tag{1.12}$$

(Theorem 5.6, Corollary 5.13 in [3]).

The authors also present a recursive algorithm for computing the n-th 1-value given n. We will not get into the details of the algorithm, but suffice it to say that the recursion is carried out to a logarithmic number of levels. Further, if the computation done at each level were shown to be constant, then the algorithm would have a running time of $O(\log n)$. For the computation at each level to be constant, certain arrangements in the sequence of 0- and 1-values must occur infinitely many times with at least constant regularity. The authors do not manage to prove this latter property, so they leave the polynomiality of their algorithm as a conjecture.

1.4 Our results

In this Thesis we make two main contributions on the function $\mathcal{G}$. The first one is a generalization of the result for the 1-values described above. Let

$$T_g = \big\langle (a_0^g, b_0^g), (a_1^g, b_1^g), (a_2^g, b_2^g), \ldots \big\rangle \tag{1.13}$$

be the sequence of g-values having $a_n^g \le b_n^g$, ordered by increasing a_n^g. Let $p_n^g = (a_n^g, b_n^g)$.

For example, Figure 1.3 plots the first terms of T_0 and T_{20}. A pattern is immediately evident: The 20-values seem to lie within a strip of constant width around the 0-values.

In this Thesis we will prove that, in fact, for all g, p_n^g is within a bounded distance to p_n^0, where the bound depends only on g, not on n. Our theoretical bound turns out to be much worse than the actual distances seen in practice. We present experimental data and compare them to our theoretical result.

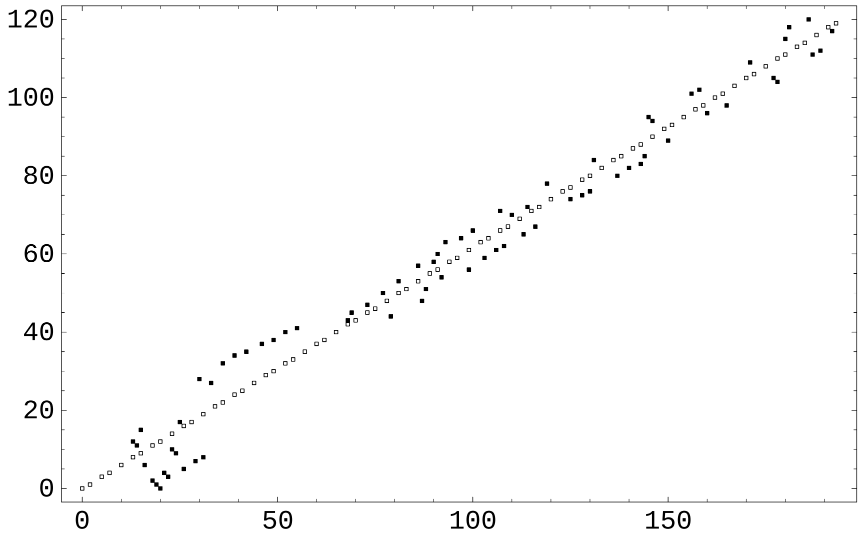


Figure 1.3: First terms of the sequences T_0 (framed squares) and T_{20} (filled squares).

Our second contribution is a modification and generalization of Blass and Fraenkel's recursive algorithm. We present a conjecture, called the *Convergence Conjecture*, which claims a certain property about the sequences T_0 through T_g. We show that for every g for which the conjecture is true, there exists an algorithm that computes p_n^g in time $O(\log n)$, where the factor implicit in the O notation depends on g.

We present experimental results that seem to support the conjecture for small g. We finally use our recursive algorithm to predict the value of several points p_n^g for small g and very large n.

1.4.1 Significance of our results

Suppose we are playing the sum of Wythoff's game with some other game, like a Nim pile. Our winning strategy, then, is to make the Grundy values of the two games equal. Suppose that the position in Wythoff has Grundy value m, and the Nim pile is of size n. Then, if $m < n$, we should reduce the Nim pile to size m, and if $m > n$, we should move in Wythoff to a position with Grundy value n.

How do the results in this Thesis help us in this scenario? The first result, regarding the location of the g-values, is of no practical help: It gives us only the approximate location of the g-values, not their precise position.

The recursive algorithm, on the other hand, has much more practical significance. If the conjecture is true for small values of g, then we can play on sums where the Nim pile is of size $\leq g$. And even if there are sporadic counterexamples to the conjecture, the recursive algorithm will probably give the correct answer in most cases, so it is a good heuristic.

1.5 Historical notes

We end this Introduction with some historical notes on the field.

The game of Nim was formally introduced by Charles L. Bouton in 1901 [4], who gave its winning strategy based on the nim-sum, or bitwise XOR, of the pile sizes. Apparently, the name of the game comes from the German *nimm*, which means "to take", or from the archaic English *nim*, which means "steal" [10].

Wythoff introduced his game in 1907 and gave formula (1.7) for its P-positions [21]. According to some sources [7, 11, 22], the game already existed in China under the name of *tsyan-shidzi* (or perhaps *jian-shi-zi* [15]), which means "choosing stones". The game was reinvented independently at least once more, by Rufus P. Isaacs in the 1960's [11]. The winning strategy for Wythoff's game is described in several places, for example [6, 7, 11, 22].

The theory of impartial games was presented by P. M. Grundy in 1939 [12]. Later, it was realized that Roland P. Sprague had published the theory independently three years earlier [19]. Therefore, now it is known as the Sprague–Grundy Theory. It is presented, for example, in [1, 5].

Chapter 2

Closeness of the g-values to the 0-values

2.1 Basic results and definitions

We begin with some basic results on $\mathcal{G}$.

Lemma 2.1 (Landman [13]) *Given x and g, there exists a unique y such that $\mathcal{G}(x, y) = g$. Moreover,*

$$g - x \leq y \leq g + 2x. \tag{2.1}$$

Proof Uniqueness follows trivially from the definition of $\mathcal{G}$.

As for existence, for any x, y, the longest path from cell (x, y) to the corner $(0, 0)$ has length $x + y$, so by Lemma 1.10, $g \leq x + y$, implying the lower bound in (2.1).

Next, let y be an integer such that $G(x, y') \neq g$ for all $y' < y$. At most g such points (x, y') have a value smaller than g, so at least $y - g$ of them have a value larger than g. Each of the latter points must be "supported" by a g-point in a lower row, either vertically of diagonally down (see Figure 2.1). No two such supporters can share a row, so there are at most x of them. On the other hand,

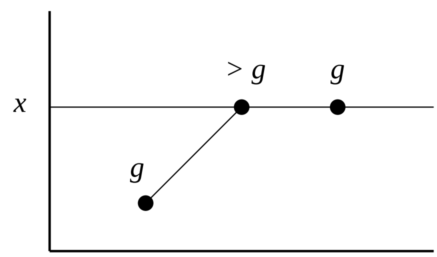


Figure 2.1: A supporter from a row lower than x.

each supporter can support at most two points in row x. Therefore, $y - g \leq 2x$, yielding the upper bound of (2.1). ∎

The following result was also known already to Landman [14]:

Lemma 2.2 *Given $g \geq 0$, there exists a unique integer x such that $G(x, x) = g$. Moreover,*

$$g/2 \leq x \leq 2g. \tag{2.2}$$

Proof As before, uniqueness follows from the definition of G. And the lower bound follows from equation (2.1).

For the upper bound, suppose x is such that $G(y, y) \neq g$ for all $y < x$. At most g of such points (y, y) have value smaller than g, so at least $x - g$ of them have value larger than g. By diagonal symmetry, for each of the latter points there exist two g-points, one to the left of (y, y) and the other one below it.

Therefore, there are at least $2(x - g)$ g-points below and to the left of point (x, x). No two of them can share a column, so $2(x - g) \leq x$, or $x \leq 2g$. ∎

Recall the terminology introduced in Section 1.3.1. Let us introduce some additional terminology (some of which we already mentioned in the Introduction). For every $g \geq 0$ we let

$$T_g = \left\langle (a_0^g, b_0^g), (a_1^g, b_1^g), (a_2^g, b_2^g), \ldots \right\rangle \tag{2.3}$$

be the sequence of g-values having $b_i^g \geq a_i^g$, ordered by increasing a_i^g. We denote the i-th g-value by $p_i^g = (a_i^g, b_i^g)$. Also, we let

$$d_i^g = b_i^g - a_i^g.$$

Note that for any fixed g, the sequence a_i^g does not contain any repeated value, and the same holds for the sequences b_i^g and d_i^g.

We define the sets

$$A_g = \{a_i^g \mid i \geq 0\}, \quad B_g = \{b_i^g \mid i \geq 0\}, \quad D_g = \{d_i^g \mid i \geq 0\}.$$

Lemmas 2.1 and 2.2, together with diagonal symmetry, imply that $|A_g \cap B_g| = 1$ and $A_g \cup B_g = \mathbb{N}$ for all g. Therefore, we could say that the sequences $\{a_i^g\}$ and $\{b_i^g\}$ are "almost complementary". We will show later on that $D_g = \mathbb{N}$ for all g.

2.2 Algorithm WSG for computing T_g

The following algorithm (Table 2.1), first formulated in [3], takes as input an integer $g \geq 0$, and computes the sequences T_h and the sets A_h, B_h, D_h for $0 \leq h \leq g$. For simplicity, we do not specify when the algorithm halts, although we could make it halt after, say, computing the first n terms of T_g.

We will rely heavily on this algorithm later on, in our analysis of T_g.

Algorithm WSG (Wythoff Sprague–Grundy)

Initialize the sets A_h, B_h, D_h, and the sequences T_h, to $\emptyset$, for $0 \le h \le g$.
For $r = 0, 1, 2, \ldots$ do:
    For $h = 0, \ldots, g$ do:
        If $r \notin B_h$ then:
          • find the smallest $d = 0, 1, 2, \ldots$ for which:
              ◦ $(r, r+d) \notin T_k$ for all $0 \le k < h$,
              ◦ $r + d \notin B_h$, and
              ◦ $d \notin D_h$;
          • append $(r, r+d)$ to T_h;
          • insert r into A_h;
          • insert $r + d$ into B_h;
          • insert d into D_h.

Table 2.1: Algorithm WSG.

2.2.1 Correctness of the algorithm

To prove the correctness of Algorithm WSG, note that it finds the purported h-values sequentially row by row, and that on each row, it only examines cells to the right of the main diagonal. Once it finds a value's location, it updates the sets A, B, and D.

The symmetric image of a cell to the right of the main diagonal always lies on a higher row, to the left of the main diagonal. Therefore, there is no need to examine cells to the left of the main diagonal, since we can assume by induction that all the lower rows were computed correctly.

The algorithm never places an h-value on the same row, column, or diagonal as a previous h-value: The test $r \notin B_h$ precludes placing an h-value on row r if there is already a reflected h-value on that row, to the left of the main diagonal. The tests $r + d \notin B_h$ and $d \notin D_h$ prevent placing an h-value on a column or diagonal that already contains an h-value.

The test $(r, r+d) \notin T_k$ prevents placing an h-value on a cell that was already assigned a lower value k.

Further, we claim that an h-value always has a k-value further down on the same row, column, or diagonal, for every $k < h$. Suppose an h-value was placed on cell $(r, r+d)$. For any $k < h$, if $r \in B_k$ then there is a k-value on row r to the left of the main diagonal. Otherwise, suppose the algorithm placed a k-value on cell $(r, r+d')$. If $d' < d$, then again the h-value has a k-value to its left. And if $d' > d$, then the k-value skipped cell $(r, r+d)$; therefore this cell must have a k-value either diagonally or vertically down.

Finally, we argue that any cell (x, y) to which no $\mathcal{G}$ value was assigned has an h-value further down on the same row, column, or diagonal, for every $h \leq g$. If $x \leq y$ then the cell is either on or after the main diagonal, and the claim holds by an argument as in the previous paragraph. And if $x > y$, then (x, y) is the symmetric image of (y, x) after the main diagonal, so again the claim holds.

Therefore, the purported Grundy values calculated by the algorithm satisfy the conditions of Lemma 1.11 (if we consider unassigned cells as having the dummy value "∞"). Therefore, the algorithm computes correctly the Grundy values $0, \ldots, g$.

2.3 Statement of the main Theorem

Recall from Theorem 1.12 that the 0-values of Wythoff's game are given by

$$(a_n^0, b_n^0) = (\lfloor \phi n \rfloor, \lfloor \phi^2 n \rfloor).$$

Graphically, the 0-values lie close to a straight line of slope ϕ^{-1} that starts at the origin.

Our main result for this Chapter is the following:

Theorem 2.3 *For every Grundy value $g \geq 0$ and every diagonal $e \geq 0$, there exists an n such that*

$$d_n^g = e$$

(i.e., every diagonal e contains a g-value).

Further, for every $g \geq 0$ there exist constants α_g, β_g, such that

$$|a_n^g - a_n^0| \leq \alpha_g, \quad |b_n^g - b_n^0| \leq \beta_g, \quad \textit{for all } n$$

(i.e., the n-th g-value is close to the n-th 0-value).

Our strategy for proving Theorem 2.3 is as follows. We first show that for every g there is a g-value in every diagonal, and furthermore, for every g there exists a constant δ_g such that

$$|d_n^g - n| \leq \delta_g \quad \text{for all } n. \tag{2.4}$$

In other words, the g-values occupy the diagonals in roughly sequential order.

Then we show how equation (2.4), together with the "almost-complementarity" of the sequences $\{a_n^g\}$ and $\{b_n^g\}$, implies that $|a_n^g - \phi n|$ and $|b_n^g - \phi^2 n|$ are bounded.

Note that for $g = 0$, Theorem 1.12 gives us

$$d_n^0 = b_n^0 - a_n^0 = (\lfloor \phi n \rfloor + n) - \lfloor \phi n \rfloor = n;$$

in other words, the 0-values occupy the diagonals in sequential order. This can be confirmed easily by following Algorithm WSG with $g = 0$.

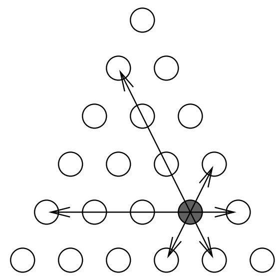


Figure 2.2: Queen in a triangular lattice.

2.4 Non-attacking queens on a triangle

The following is a variation on the well-known “eight queens problem”. We will use its solution in proving bound (2.4).

We are given a triangular lattice of side n, as shown in Figure 2.2. A queen on the lattice can move along a straight line parallel to any of the sides of the triangle. How many queens can be placed on the lattice, without any two queens attacking each other?

Theorem 2.4 ([16]) *The maximum number of non-attacking queens that can be placed on a triangular lattice of side n is exactly*

$$q(n) = \left\lfloor \frac{2n+1}{3} \right\rfloor.$$

We give the proof in Appendix A.

2.5 d_n^g is close to n

For convenience, in this Section we fix g, and we write $a_n = a_n^g$, $b_n = b_n^g$, $d_n = d_n^g$, $p_n = p_n^g$. Whenever we refer to h-points, $h < g$, we will say so explicitly.

Recall that the points p_n are ordered by increasing row a_n, so that $a_n > a_m$ if and only if $n > m$.

In this Section we will make extensive use of Algorithm WSG.

Observe that Algorithm WSG does not place a g-point on certain rows r, because it skips row r on line 4. Then such an r is not added to A_g (line 10). We call such an r a *skipped row*.

Similarly, sometimes a certain column c is never inserted into B_g (line 11), because no point p_m falls on that column. In that case, we call column c a *skipped column*.

Let us define the notion of a g-point skipping a diagonal. Intuitively, point p_m skips diagonal e if Algorithm WSG places point p_m after diagonal e, while diagonal e does not yet contain a g-point (see Figure 2.3). Formally:

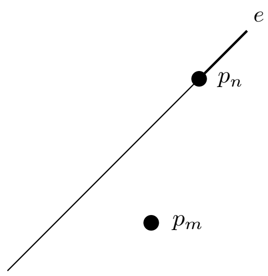


Figure 2.3: Point p_m skips diagonal e.

Definition 2.5 Diagonal e is *empty up to row* r if there is no point p_n with $d_n = e$ and $a_n \leq r$.

Definition 2.6 Point p_m is said to *skip diagonal* e if $d_m > e$ and e is empty up to row a_m.

Our goal in this Section is to derive a bound on $|d_n - n|$. We will derive separately upper bounds for $d_n - n$ and $n - d_n$. We do this by bounding the number of diagonals that a given point can skip, and the number of points that can skip a given diagonal:

Lemma 2.7 *If no point p_n skips more than k diagonals, then $d_n - n \leq k$ for all n. If no diagonal is skipped by more than k points, then $n - d_n \leq k$ for all n.*

Proof For the first claim, suppose by contradiction that $d_n - n > k$ for some n. Then, of the d_n diagonals $0, \ldots, d_n - 1$, only n can be occupied by points $p_0, \ldots, p_{n-1}$. Therefore, point p_n skips at least $k + 1$ diagonals.

For the second claim, suppose by contradiction that $n - d_n > k$ for some n. Then, of the n points $p_0, \ldots, p_{n-1}$, only d_n can fall on diagonals $0, \ldots, d_n - 1$. Therefore, diagonal d_n is skipped by at least $k + 1$ points. ∎

Let us inspect why a g-point skips a diagonal according to Algorithm WSG. Suppose point p_m skips diagonal e, and let $C = (a_m, a_m + e)$ be the cell on diagonal e on the row in which p_m was inserted. Then, point p_m skipped diagonal e, either because cell C was already assigned some value $k < g$ (WSG line 6), or because there was already a point $p_{m'}$ directly below cell C (WSG line 7).

We need a further definition: Let e be a diagonal and r be a row, such that diagonal e is empty up to row $r - 1$. Draw a line from the intersection of row r and diagonal e vertically down. If a point p_m is strictly below row r, and on or to the right of the said vertical line, then we say that p_m is *active with respect to diagonal e and row r* (see Figure 2.4). In other words:

Definition 2.8 If diagonal e is empty up to row $r - 1$, then point p_m is *active with respect to diagonal e and row r* if $a_m < r$ and $b_m \geq r + e$.

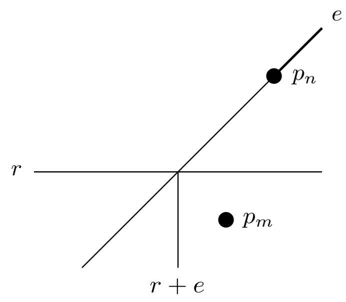


Figure 2.4: Point p_m is active with respect to diagonal e and row r.

We can bound the number of active g-points:

Lemma 2.9 *The number of g-points active with respect to any diagonal e and any row r is at most g.*

Proof By assumption, diagonal e is empty up to row $r-1$.

We will show that for every $r' \leq r$, if diagonal e contains k h-points, $h < g$, below row r', then there can be at most k active g-points with respect to diagonal e and row r'. This implies our Lemma, since there are at most g h-points, $h < g$, on diagonal e.

We prove the above claim by induction on r'. If $r' = 0$ then clearly $k = 0$ and there are no active g-points with respect to e and r'.

Suppose our claim is true up to row r', and let us examine Algorithm WSG on row r' itself. If no point p_m is inserted on row r', then the number of active g-points does not increase when we go from row r' to row $r'+1$. And if point p_m is inserted on row r' and it skips diagonal e, it must be for one of the two reasons mentioned above. If there is an h-point, $h < g$, on the intersection of row r' with diagonal e, then the number k of our claim increases by 1 when we go to row $r'+1$. And if there is no such h-point, then there must be an earlier point $p_{m'}$ directly below the intersection of e and r'. But then $p_{m'}$ is active with respect to e and r', but not with respect to e and $r'+1$, so the number of active g-points stays the same when we go from row r' to row $r'+1$.

So in either case, the inductive claim is also true for row $r'+1$. ∎

We can now bound the number of diagonals a given g-point can skip:

Lemma 2.10 *A point p_m can skip at most $2g$ diagonals.*

Proof Let e_0 be the first diagonal skipped by point p_m. For every diagonal e skipped by p_m, there must be either an active g-point with respect to diagonal e_0 and row a_m, or an h-point, $h < g$, on cell $(a_m, a_m + e)$. There can be at most g of the latter, and by Lemma 2.9, at most g of the former. ∎

We proceed to bound the number of g-points that can skip a given diagonal. For this we need a lower bound on the number of skipped columns in an interval of consecutive columns:

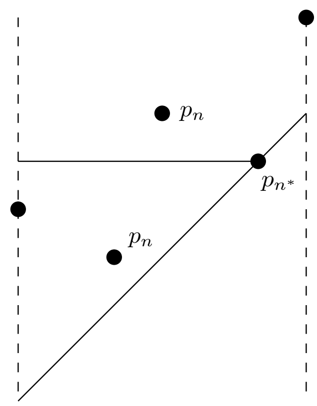


Figure 2.5: p_{n^*} is the point p_n with maximum d_n.

Lemma 2.11 *An interval of k consecutive columns contains at least*

$$\frac{k-4}{3} - 2g$$

skipped columns.

Proof Consider the points p_n that lie within the given interval of columns. Let p_{n^*} be the point p_n with maximum d_n (see Figure 2.5). The number of points p_n with $n > n^*$ is at most $2g$ by Lemma 2.10. And the points p_n with $n < n^*$ are confined to a triangular lattice of size $\leq k$; but this situation is isomorphic to the non-attacking queens of Section 2.4!

Therefore, the number of points p_n with $n < n^*$ is at most

$$\left\lfloor \frac{2k+1}{3} \right\rfloor \leq \frac{2k+1}{3}.$$

Thus, the total number of points p_n is at most $(2k+1)/3+2g+1$, so the number of skipped columns is at least

$$\frac{k-4}{3} - 2g. \quad \blacksquare$$

Corollary 2.12 *An interval of k consecutive rows contains at least $(k-4)/3-2g$ points p_n.*

Proof By diagonal symmetry: If column c is a skipped column, then row c contains a point p_n. $\blacksquare$

Lemma 2.13 *Suppose point p_m skips diagonal e, and let $\Delta = d_m - e$. Suppose diagonal e is empty up to row $a_m + \Delta$ (see Figure 2.6). Then $\Delta \leq 15g + 4$.*

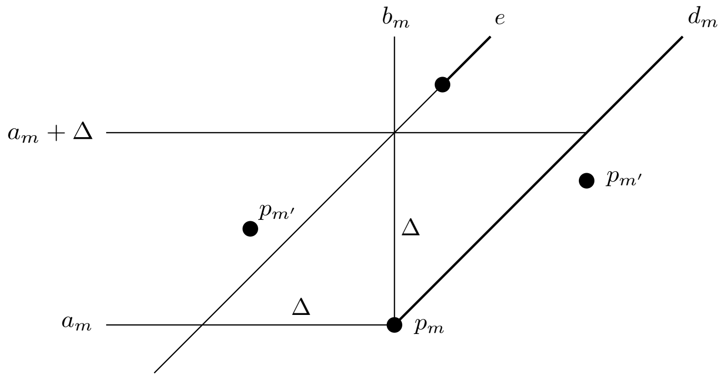


Figure 2.6: Points $p_{m'}$ between rows a_m and $a_m + \Delta$.

Proof Let us bound the number of points $p_{m'}$ in the interval between row $a_m + 1$ and row $a_m + \Delta$. For every such $p_{m'}$, either $d_{m'} < d_m$, or $b_{m'} > b_m$, or both (see Figure 2.6). In the former case, p_m skips diagonal $d_{m'}$, and in the latter case, $p_{m'}$ is active with respect to diagonal e and row $a_m + \Delta + 1$. So by Lemmas 2.9 and 2.10, there are at most $3g$ such points $p_{m'}$.

Therefore, by Corollary 2.12, we must have

$$\frac{\Delta - 4}{3} - 2g \leq 3g,$$

so $\Delta \leq 15g + 4$. ∎

Corollary 2.14 *The number of points p_n that can skip a given diagonal e is at most $16g + 4$.*

Proof Every point that skips diagonal e must lie on a different diagonal. Therefore, Lemma 2.13 already implies that diagonal e must eventually be occupied by a point p_n.

Now, consider the points p_m, $m < n$, that skip diagonal $e = d_n$. Partition these points into two sets: those having $b_m > b_n$ and those having $b_m < b_n$.

By Lemma 2.9, there are at most g points in the first set, since each such point is active with respect to the diagonal and row of p_n. And every point in the second set satisfies the assumptions of Lemma 2.13, so there are at most $15g + 4$ such points. Therefore, diagonal e is skipped by no more than $16g + 4$ points. ∎

We used somewhat messy arguments, but we have finally proven:

Theorem 2.15 *For every Grundy value g and every diagonal e, there exists a g-point with $d_n^g = e$. Furthermore,*

$$-16g - 4 \leq d_n^g - n \leq 2g. \quad ∎$$

2.6 The g-values are close to the 0-values

We proceed to show the existence of the constants α_g and β_g of Theorem 2.3. In order to understand the idea behind our proof, it is helpful to look first at the following proof that the ratio between consecutive Fibonacci numbers tends to ϕ:

Claim 2.16 *Let F_n be the n-th Fibonacci number. Then $F_n/F_{n-1} \to \phi$.*

Proof Let $x_n = F_n - \phi F_{n-1}$. Then,

$$\begin{aligned} x_{n+1} = F_{n+1} - \phi F_n &= (F_n + F_{n-1}) - \phi F_n \\ &= -\phi^{-1}(F_n - \phi F_{n-1}) \\ &= -\phi^{-1} x_n. \end{aligned} \tag{2.5}$$

Therefore, $x_n \to 0$, since $|-\phi^{-1}| < 1$. Therefore,

$$\frac{F_n}{F_{n-1}} = \frac{x_n}{F_{n-1}} + \phi \to \phi. \quad \blacksquare$$

Next, we introduce the following notation, which will help make our arguments clearer:

Definition 2.17 Given sequences $\{f_n\}$ and $\{g_n\}$, we write

$$f_n \sim g_n$$

if, for some k, $|f_n - g_n| \leq k$ for all n.

Note that the relation $\sim$ is transitive: If $f_n \sim g_n$ and $g_n \sim h_n$, then $f_n \sim h_n$.

In this Section we make a few claims that are intuitively obvious. We therefore decided to defer their proofs to Appendix A, in order not to interrupt the main flow of the arguments. Our first intuitive claim is the following:

Lemma 2.18 *Let $\{x_n\}$ be a sequence that satisfies $x_{n+1} \sim cx_n$ for some $|c| < 1$. Then $\{x_n\}$ is bounded as a sequence.*

The main result of this Section is the following somewhat general theorem:

Theorem 2.19 *Let $a_0 < a_1 < a_2 < \cdots$ be a sequence of increasing natural numbers, and let $b_0, b_1, b_2, \ldots$ be a sequence of distinct natural numbers. Let $A = \{a_n \mid n \geq 0\}$, $B = \{b_n \mid n \geq 0\}$. Suppose the following conditions hold:*

1. *$|A \cap B|$ is finite;*
2. *$A \cup B = \mathbb{N}$;*
3. *$b_n - a_n \sim n$.*

Then $a_n \sim \phi n$ and $b_n \sim \phi^2 n$.

Note that, in particular, our Wythoff sequences $\{a_n^g\}$ and $\{b_n^g\}$ satisfy all of the above requirements, so the above theorem yields Theorem 2.3, as desired.

Proof of Theorem 2.19 We start with the following claim, which we prove in Appendix A:

Lemma 2.20 *Regarding the sequences $\{a_n\}$ and $\{b_n\}$:*

(a) There is a constant k such that for all n, the number of $b_m > b_n$, $m < n$, is at most k.

(b) There is a constant k' such that for all n, the number of $b_m < b_n$, $m > n$, is at most k'.

(c) $a_n \sim a_{n-1}$ and $b_n \sim b_{n-1}$.

(Note that, for our sequences $\{a_n^g\}$ and $\{b_n^g\}$, the lemmas of Section 2.5 already give bounds on the number of m's in Lemma 2.20(a,b). But we still want to prove Theorem 2.19 in general.)

Now, for $n \geq 0$, define

$$x_n = \phi n - a_n,$$

and let

$$f(n) = \big|A \cap \{0, \ldots, b_n - 1\}\big|$$

be the number of a's smaller than b_n.

By Lemma 2.20(a,b), the number of b's smaller than b_n is $\sim n$, so by conditions 1 and 2 of our Theorem, the number of a's smaller than b_n is $\sim b_n - n$. And by condition 3 we have $b_n - n \sim a_n$. Therefore,

$$f(n) \sim a_n. \tag{2.6}$$

Further, $a_{f(n)}$ is the first a that is $\geq b_n$ (by the definition of $f(n)$), so $a_{f(n)} \sim b_n$ by Lemma 2.20(c). Therefore (compare with (2.5)),

$$\begin{aligned} x_{f(n)} = \phi f(n) - a_{f(n)} &\sim \phi a_n - b_n \\ &\sim \phi a_n - a_n - n \\ &= \phi^{-1} a_n - n \\ &= -\phi^{-1} x_n. \end{aligned} \tag{2.7}$$

The following lemma is proven in Appendix A:

Lemma 2.21 *There exists an integer n_1 such that $f(n) > n$ for all $n \geq n_1$.*

Now, choose n_1 as in Lemma 2.21, and define the sequence $n_1, n_2, n_3, \ldots$, by $n_{i+1} = f(n_i)$. This sequence, therefore, tends to infinity. Also let $n_0 = 0$.

Define the sequence $\{y_j\}_{j=0}^{\infty}$ by

$$y_j = \max_{n_j \leq i \leq n_{j+1}} |x_i|, \quad \text{for } j \geq 0. \tag{2.8}$$

Lemma 2.22 *The sequence* $\{y_j\}$ *satisfies* $y_{j+1} \sim \phi^{-1} y_j$.

This follows from equation (2.7) and the fact that $x_n \sim x_{n+k}$ for every constant k, which is a consequence of Lemma 2.20(c). The full proof of Lemma 2.22 is given in Appendix A.

Therefore, by Lemma 2.18, $\{y_j\}$ is bounded as a sequence, so $\{|x_n|\}$ is bounded as a sequence. Therefore, $a_n \sim \phi n$, and by condition 3 of our Theorem, $b_n \sim \phi^2 n$. ∎

This completes the proof of the existence of the constants α_g and β_g of Theorem 2.3.

2.7 Experimental results

In this Section we present experimental results on a few aspects of the function $\mathcal{G}$.

2.7.1 Experimental bounds on $d_n^g - n$

We now compare the rigorous bound of $-16g - 4 \leq d_n^g - n \leq 2g$ given by Theorem 2.15 with data obtained experimentally.

Table 2.2 shows the extreme values of $d_n^g - n$ achieved for different g by points lying in rows up to $5 \cdot 10^6$. In each case we show the earliest appearance of the extremal value.

We notice an interesting phenomenon: For $g \geq 7$ the maximum is achieved by the zeroth point $p_0^g = (0, g)$. This phenomenon is due to the fact that the sequence T_g tends to start with an anomalous behavior that "smooths out" over time.

Table 2.3 shows the maximum $d_n^g - n$ achieved by points having $n \geq 100$, for $g \geq 7$. We see that as g grows, the maximum achieved decreases substantially from Table 2.2.

Finally, Figure 2.7 shows a histogram of $d_n^g - n$ for $g = 30$, counting points up to row $5 \cdot 10^6$.

The conclusion from these observations is the following: The bounds observed experimentally for $d_n^g - n$ are much tighter than those given by Theorem 2.15. Therefore, either the theoretical bound is much looser than necessary, or it is a worst-case bound that is achieved very rarely in practice.

2.7.2 The converse of Theorem 2.3

Theorem 2.3 implies that if $\mathcal{G}(a, b)$ is bounded with $a \leq b$, then $|b - \phi a|$ is bounded. Is the converse also true? Namely, does a bound on $|b - \phi a|$ imply a bound on $\mathcal{G}(a, b)$? Or, on the contrary, are there arbitrarily large values very close to 0-values? We do not know the answer, but we explored this question experimentally.

g	$\min d_n^g - n$	n	$\max d_n^g - n$	n
0	0	0	0	0
1	−4	57	2	282
2	−6	35745	3	38814
3	−8	149804	4	2335
4	−10	569350	5	15486
5	−11	1245820	6	2638
6	−11	30165	7	1974933
7	−11	75459	7	0
8	−12	701260	8	0
9	−13	17972	9	0
10	−13	516328	10	0
11	−14	722842	11	0
12	−16	2853838	12	0
13	−17	2860809	13	0
14	−18	2814039	14	0
15	−18	2597774	15	0
16	−18	1027151	16	0
17	−18	2979529	17	0
18	−19	789978	18	0
19	−20	22347	19	0
20	−21	2548028	20	0
21	−19	277362	21	0
22	−20	30200	22	0
23	−23	1412268	23	0
24	−22	684205	24	0
25	−23	349878	25	0
26	−24	2087092	26	0
27	−24	617166	27	0
28	−24	2343474	28	0
29	−26	27	29	0
30	−27	1872274	30	0

Table 2.2: Extreme values of $d_n^g - n$ for given g, for points p_n^g having $a_n^g \leq 5 \cdot 10^6$.

g	$\max d_n^g - n$	n	g	$\max d_n^g - n$	n
7	7	131307	19	14	594141
8	8	20735	20	14	2482469
9	9	1056831	21	14	90130
10	9	258676	22	15	347510
11	10	987102	23	15	323425
12	10	1295870	24	16	129240
13	10	90426	25	17	1880006
14	11	453415	26	17	36662
15	11	61780	27	18	332552
16	12	509772	28	18	370321
17	12	86093	29	19	2425182
18	13	32439	30	18	444272

Table 2.3: Maximum value of $d_n^g - n$ for given g, for points p_n^g having $n \geq 100$ and $a_n^g \leq 5 \cdot 10^6$.

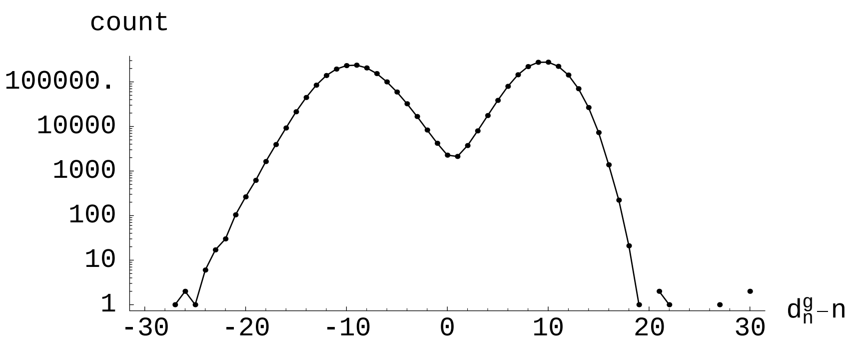


Figure 2.7: Histogram of $d_n^g - n$ for $g = 30$, counting points up to row $5 \cdot 10^6$.

distance	cell	$\mathcal{G}$ value	cell (row ≥ 600)	$\mathcal{G}$ value (row ≥ 600)
1	$(283432, 458601)$	82		
2	$(944634, 1528447)$	96		
3	$(44, 67)$	89	$(82399, 133320)$	81
4	$(49, 86)$	115	$(665224, 1076349)$	82
5	$(58, 86)$	116	$(402997, 652071)$	103
6	$(62, 110)$	147	$(538568, 871413)$	99
7	$(97, 168)$	≥ 200	$(839162, 1357804)$	108
8	$(95, 167)$	≥ 200	$(182922, 295987)$	115
9	$(87, 155)$	≥ 200	$(319656, 517229)$	122
10	$(85, 154)$	≥ 200	$(927492, 1500730)$	125

Table 2.4: Large Grundy values at a given Manhattan distance from the closest 0-value.

We looked for the cells with the largest Grundy value lying at a given Manhattan distance from the closest 0-value. (The *Manhattan distance* between (a_1, b_1) and (a_2, b_2) is defined as $|a_2 - a_1| + |b_2 - b_1|$.) We looked up to row 10^6, calculating points up to $g = 199$. Our results are shown in Table 2.4.

The first column in Table 2.4 lists the cell with the largest Grundy value at a given Manhattan distance from the closest 0-value, for cells in rows $\leq 10^6$. Some cells are labelled "≥ 200" because the were not assigned any Grundy value ≤ 199.

The second column in Table 2.4 lists the cell with the largest Grundy value at a given Manhattan distance from the closest 0-value, restricted to cells between rows 600 and 10^6. (Only entries differing from the first column are shown.)

We see a very significant difference in the Grundy values between the first and second columns. This phenomenon is due to the fact that the sequences T_g tend to start by passing very close to the 0-points, before "smoothing out". This can be seen in Figure 2.8, which plots the beginning of T_{200}.

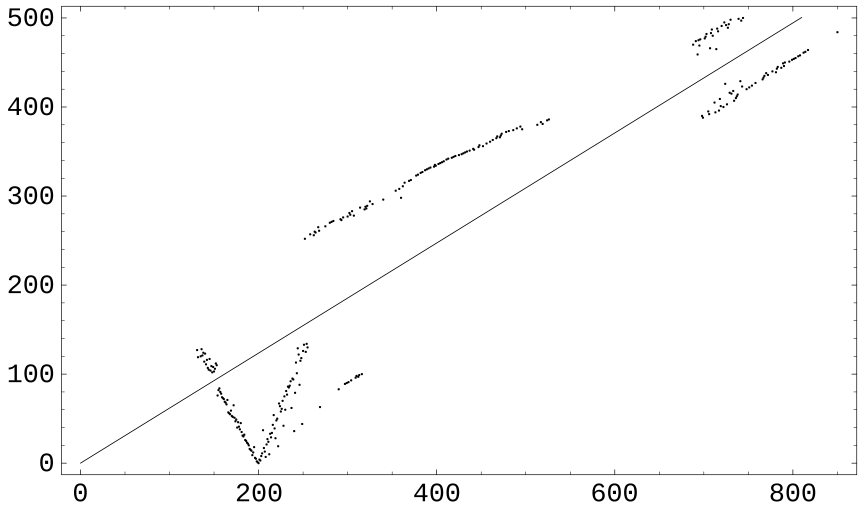


Figure 2.8: First terms of the sequence T_{200}.

Chapter 3

A recursive algorithm for the n-th g-value

Consider Algorithm WSG as it is about to start inserting points in row r. For each h, $0 \leq h \leq g$, there are cells on row r that cannot take an h-point: some because they have an h-point below them, some because they have an h-point diagonally below, and others because they have already been assigned a value $k < h$.

We will show how we can represent this information about row r, which we will call its *state*, in such a way that there is a finite number of states among all rows. We will further consider the set of h-points to be inserted in row r as a single *symbol* out of a finite alphabet of symbols. Then, with the state of row r and the said symbol, we can correctly place along row r the relevant h-points, and compute the state of row $r + 1$.

Therefore, we can reformulate Algorithm WSG as a *finite-state automaton* that jumps from state to state as it reads symbols from some external source. (Here we ignore the fact that the points to be inserted at row r are determined by the points inserted in lower rows.)

This finite-state formulation forms the basis for a conjecture which, if true, leads to a recursive algorithm for finding the n-th g-point in time $O(f(g) \log n)$, where f is some function on g.

Let us now develop these ideas formally.

3.1 A finite-state algorithm

For the rest of this Chapter, we fix g. The variable h will always take the values $0 \leq h \leq g$.

In this Chapter, whenever we refer to an h-point, we mean a point (x, y) with $\mathcal{G}(x, y) = h$ and $x \leq y$. Points with $x \geq y$ will be referred to as *reflected* h-points. A point with $x = y$ is both reflected and unreflected.

Definition 3.1

1. $\mathit{index}_h(r) = \big|A_h \cap \{0, \ldots, r-1\}\big|$ is the number of h-points strictly below row r. It is also the index of the first h-point on a row $\geq r$.

2. $D_h(r) = \{d_n^h \mid 0 \leq n < \mathit{index}_h(r)\}$ is the set of diagonals occupied by h-points on rows $< r$.

3. $\mathit{firstd}_h(r) = \operatorname{mex} D_h(r)$ denotes the first empty diagonal on row r.

4. $\mathit{ocdiag}_h(r) = \{d \in D_h(r) \mid d > \mathit{firstd}_h(r)\}$ is the set of occupied diagonals after the first empty diagonal on row r.

   Note that
   $$\mathit{index}_h(r) = \mathit{firstd}_h(r) + |\mathit{ocdiag}_h(r)|, \tag{3.1}$$
   because there are $\mathit{index}_h(r)$ occupied diagonals on row r: diagonals 0 through $\mathit{firstd}_h(r) - 1$, and $|\mathit{ocdiag}_h(r)|$ additional ones.

5. Similarly, $B_h(r) = \{b_n^h \mid 0 \leq n < \mathit{index}_h(r)\}$ is the set of columns occupied by h-points on rows $< r$.

6. We let $\mathit{occol}_h(r) = \{b - r \mid b \in B_h(r),\ b - r \geq \mathit{firstd}_h(r)\}$. This set has the following interpretation: Identify each cell on row r by the diagonal it lies on, i.e., cell (r, b) is identified by $b - r$. Then $\mathit{occol}_h(r)$ represents the set of cells on row r on a diagonal $\geq \mathit{firstd}_h(r)$ that lie on a column occupied by a lower h-point.

Lemma 3.2 *The expression*
$$\big(\max \mathit{ocdiag}_h(r)\big) - \mathit{firstd}_h(r) \tag{3.2}$$
is bounded for all r, and so is $|\mathit{ocdiag}_h(r)|$.

Proof Expression (3.2) corresponds to the maximum distance between a free diagonal and a subsequent occupied diagonal on row r; and this is bounded by Theorem 2.3. And since $\mathit{ocdiag}_h(r)$ only contains integers $> \mathit{firstd}_h(r)$, its size is also bounded. ∎

Lemma 3.3 $\max \mathit{occol}_h(r) < \max \mathit{ocdiag}_h(r)$ *for all h, r.*

Proof Suppose a cell on row r lies on a column occupied by a lower h-point. Then a cell on row r further to the right lies on the diagonal occupied by that h-point. ∎

Lemma 3.4 *For $g = 0$ we can compute explicitly the quantities of Definition 3.1. In particular,*
$$\mathit{index}_0(r) = \mathit{firstd}_0(r) = \lceil \phi^{-1} r \rceil, \tag{3.3}$$
$$\mathit{ocdiag}_0(r) = \mathit{occol}_0(r) = \emptyset. \tag{3.4}$$

Proof $index_0(r)$ is the index of the first 0-point on a row $\geq r$. Since $a_n^0 = \lfloor \phi n \rfloor$, the first n that gives $a_n^0 \geq r$ is $n = \lceil \phi^{-1} r \rceil$. (3.4) follows from the fact that the 0-points fill the diagonals in sequential order. And the second part of (3.3) follows from (3.1). ∎

Definition 3.5 We define the following "normalized" quantities by subtracting $index_0(r)$:

$$\begin{aligned} n\_index_h(r) &= index_h(r) - index_0(r), \\ n\_firstd_h(r) &= firstd_h(r) - index_0(r), \\ n\_ocdiag_h(r) &= \{d - index_0(r) \mid d \in ocdiag_h(r)\}, \\ n\_occol_h(r) &= \{c - index_0(r) \mid c \in occol_h(r)\}. \end{aligned}$$

Lemma 3.6 *The quantities $n\_index_h(r)$ and $n\_firstd_h(r)$, as well as the elements of $n\_ocdiag_h(r)$ and $n\_occol_h(r)$, are bounded in absolute value for all r.*

Proof This follows from Theorem 2.3. ∎

Definition 3.7 The *state* of a given row r consists of

$$n\_index_h(r),\ n\_firstd_h(r),\ n\_ocdiag_h(r),\ n\_occol_h(r),$$

for $0 \leq h \leq g$.

By Lemma 3.6 we have:

Corollary 3.8 *There is a finite number of distinct states among all rows $r \geq 0$.* ∎

Note that the state of a row r always satisfies

$$n\_index_h(r) = n\_firstd_h(r) + |n\_ocdiag_h(r)|. \tag{3.5}$$

Note also that if $g = 0$ there is a single state for all rows:

$$n\_index_0(r) = n\_firstd_0(r) = 0, \quad n\_ocdiag_0(r) = n\_occol_0(r) = \emptyset.$$

Definition 3.9 Given row r, we denote by

$$insert(r) = \{h \mid r \in A_h, 0 \leq h \leq g\}$$

the set of h-points that Algorithm WSG must insert in this row.

Definitions 3.7 and 3.9 enable us to reformulate Algorithm WSG as a finite-state automaton that jumps from state to state as it reads symbols from a finite alphabet Σ. The automaton is in the state of row r when it begins to calculate row r, and after reading the symbol $insert(r)$, it goes to the state of row $r + 1$. The input alphabet is $\Sigma = 2^{\{0,\ldots,g\}}$, the set of all possible values of $insert(r)$.

Algorithm FSW (Table 3.1) spells out in detail this finite-state formulation. (This algorithm is not equivalent to Algorithm WSG because it does not calculate the sets $insert(r)$ from previous rows, but only receives them as input.)

Algorithm FSW (Finite-State Wythoff)

Input: Integer g; integers r_1, r_2 (initial and final rows); state at row r_1, given by the variables

$$n\_index_h, n\_firstd_h, n\_ocdiag_h, n\_occol_h, \quad \text{for } 0 \le h \le g; \tag{3.6}$$

sets $insert(r_1), \ldots, insert(r_2)$ (points to insert in rows $r_1, \ldots, r_2$).

Output: h-points in rows r_1 through r_2, given as 4-tuples

$$(h, n, a_n^h, b_n^h) \quad \text{for } 0 \le h \le g,\ r_1 \le a_n^h \le r_2.$$

1. For $r = r_1, \ldots, r_2$ do:
2. • Let $S \leftarrow \emptyset$ [location of points inserted in this row].
3. • For $h = 0, \ldots, g$ do:
4. ◦ If $h \in insert(r)$ then:
5. ∗ find the smallest $d \ge n\_firstd_h$ which is in none of the sets n_ocdiag_h, n_occol_h, and S;
6. ∗ output the 4-tuple

$$\big(h,\ n\_index_h + index_0(r),\ r,\ r + d + index_0(r)\big)$$

[note that $index_0(r)$ can be calculated by Lemma 3.4];

7. ∗ let $n\_index_h \leftarrow n\_index_h + 1$;
8. ∗ insert d into n_ocdiag_h, n_occol_h, and S;
9. ∗ while $n\_firstd_h \in n\_ocdiag_h$ do $n\_firstd_h \leftarrow n\_firstd_h + 1$.
10. ◦ Subtract 1 from each element of n_occol_h [since r increases by 1—see Definition 3.1–6].
11. ◦ Remove from n_ocdiag_h and n_occol_h all elements $< n\_firstd_h$.
12. • If $n\_index_0 = 1$ then [renormalize]:
13. ◦ subtract 1 from n_index_h and n_firstd_h for all h;
14. ◦ subtract 1 from each element of n_ocdiag_h and n_occol_h for all h.

Table 3.1: Algorithm FSW.

3.2 Convergence of states

Suppose we run Algorithm FSW starting from some row r_1, giving it as input the correct values of $insert(r_1), insert(r_1 + 1), \ldots$, but with a different initial state

$$n\_index'_h, n\_firstd'_h, n\_ocdiag'_h, n\_occol'_h, \quad 0 \le h \le g, \tag{3.7}$$

instead of (3.6). Then the algorithm will output 4-tuples $(h, n, a_n^h, {b'}_n^h)$, where the b-coordinates of the points will not necessarily be correct.

Could it happen that at some row $r > r_1$ the algorithm reaches the correct state for row r? If that happens, then for all subsequent rows the algorithm will be in the correct state, since the state of a row depends only on the state of the previous row. Therefore, for all rows $\geq r$ the algorithm will output the correct 4-tuples (h, n, a_n^h, b_n^h).

Denote by

$$\mathit{n\_index}'_h(r), \mathit{n\_firstd}'_h(r), \mathit{n\_ocdiag}'_h(r), \mathit{n\_occol}'_h(r), \quad 0 \leq h \leq g, \quad r \geq r_1,$$

the state of the algorithm at row r when run with the initial state (3.7). We assume that the initial state (3.7) is consistent with property (3.5).

Observe that if $\mathit{n\_index}'_h(r_1) \neq \mathit{n\_index}_h(r_1)$, this difference will persist in all subsequent rows, since changes to $\mathit{n\_index}_h$ at Lines 7 and 13 depend only on the input symbol $\mathit{insert}(r)$. Therefore, convergence can only occur if the initial state contains the correct values of $\mathit{n\_index}_h(r_1)$ for all h.

Now, we make the following conjecture:

Conjecture 3.10 (Convergence Conjecture) *For every g there exists a constant R_g such that for every row r_1, if Algorithm FSW is run starting from row r_1 with the initial "dummy" state*

$$\begin{gathered} \mathit{n\_index}'_h = \mathit{n\_firstd}'_h = \mathit{n\_index}_h(r_1), \\ \mathit{n\_ocdiag}'_h = \mathit{n\_occol}'_h = \emptyset, \end{gathered} \qquad \text{for } 0 \leq h \leq g, \tag{3.8}$$

and with the correct values of $\mathit{insert}(r_1), \mathit{insert}(r_1 + 1), \ldots$, then the algorithm will converge to the correct state within at most R_g rows.

3.2.1 Experimental evidence for convergence

We tested Conjecture 3.10 experimentally as follows: For some constant $r_{\max}$ we precalculated the state of row r and the value of $\mathit{insert}(r)$ for all r between 0 and $r_{\max}$. We then ran Algorithm FSW starting from each row r, $0 \leq r \leq r_{\max}$, with the dummy initial state (3.8), comparing at each step whether the algorithm's internal state converged to the correct state of the current row. We carried out this experiment for different values of g.

Our results are as follows: For $g = 0$ convergence always occurs after 0 rows; i.e., convergence is immediate.

For $g = 1$ the maximum time to convergence found was 45 rows. In fact, up to row 10^7 there are 3019 instances of convergence taking 45 rows.

For $g = 2$ the maximum found was 72 rows. Below row 10^7 there are 91 instances of convergence taking 72 rows.

For $g = 3$ the maximum of 140 rows to convergence is achieved only once below row 10^7.

For larger values of g we ran our experiment until row 10^6. Table 3.2 shows our findings. In each case we indicate the largest number of rows to convergence, and the starting row that achieves the maximum (or the first such starting row in case there are several).

g	rows to convergence	starting row
0	0	0
1	45	2201
2	72	72058
3	140	804421
4	180	862429
5	235	732494
6	395	685531
7	395	685531
8	461	827469
9	630	59948
10	909	443109
$\vdots$		
15	2041	8662
$\vdots$		
20	4136	896721

Table 3.2: Maximum number of rows to convergence for different g up to row 10^6, and first starting row that achieves the maximum.

Finally, Figure 3.1 shows a histogram of the time to convergence for $g = 10$ up to row 10^6. The shape of the curve suggests that there might be instances of higher convergence times that occur very rarely. However, we still find it plausible that a theoretical maximum R_g exists.

Note that Conjecture 3.10 could also be true only up to a certain value of g.

3.3 The recursive algorithm

We now show how Conjecture 3.10 leads to an algorithm for computing point p_n^g in time $O(f(g) \log n)$, where f is some function on the constants R_g, α_g, and β_g.

Algorithm RW (Table 3.3) is a recursive algorithm that receives as input an integer g and an interval $[r_1, r_2]$ of rows, and calculates all h-points, $0 \le h \le g$, in that interval.

The idea behind Algorithm RW is the following: To calculate the h-points between rows r_1 and r_2, we run Algorithm FSW starting from row $r_0 = r_1 - R_g$ and the dummy initial state (3.8). Then, by Conjecture 3.10, the 4-tuples obtained from row r_1 on will be the correct ones (h, n, a_n^h, b_n^h).

But to run Algorithm FSW we also need to know $insert(r_0), \ldots, insert(r_2)$, i.e., which h-points to insert between rows r_0 and r_2. We obtain this information by calculating all the *reflected* h-points that lie between rows r_0 and r_2, to the

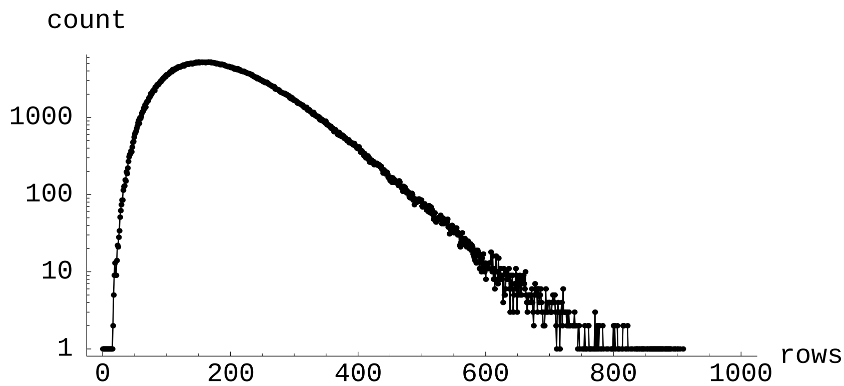


Figure 3.1: Histogram of the number of rows to convergence for $g = 10$, up to row 10^6.

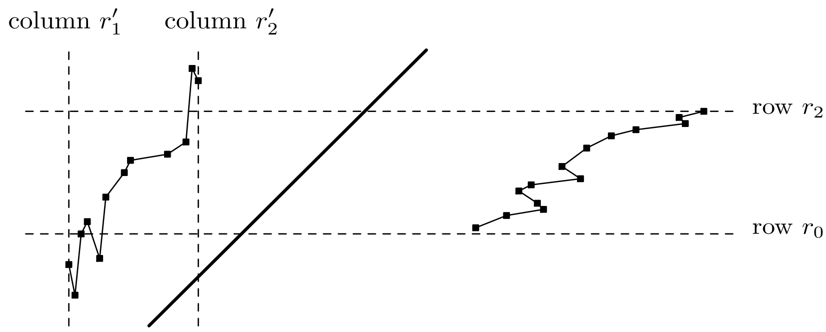


Figure 3.2: Points and reflected points in rows r_0 through r_2.

Algorithm RW (Recursive Wythoff)

Input: Integer g; integers r_1, r_2 (initial and final rows).

Output: $index_h(r_1)$ for $0 \le h \le g$; set S of h-points in rows r_1 through r_2, given as 4-tuples (h, n, a_n^h, b_n^h) for $0 \le h \le g,\ r_1 \le a_n^h \le r_2$.

1. Let $r_0 \leftarrow r_1 - R_g$.
2. Let L and H be lower and upper bounds for $a_n^h - \phi^{-1} b_n^h$ for $0 \le h \le g$ and all n, according to Theorem 2.3. [Note that $a_n^h < \phi^{-1} r + L$ implies $b_n^h < r$, and $a_n^h > \phi^{-1} r + H$ implies $b_n^h > r$.]
3. Let $r_1' \leftarrow \lceil \phi^{-1} r_0 + L \rceil$, $r_2' \leftarrow \lfloor \phi^{-1} r_2 + H \rfloor$.
4. If $r_2' \ge r_0$ or $r_1' \le 2g$ then:
5. • calculate and return the desired points by starting from row 0 using Algorithm WSG;
6. else:
7. • call Algorithm RW recursively and get $index_h(r_1')$ and the set S' of h-points in rows r_1' through r_2' for $0 \le h \le g$;
8. • calculate $insert(r_0), \ldots, insert(r_2)$ as

$$insert(r) = \left\{ h \;\middle|\; \nexists n \text{ for which } (h, n, a_n^h, r) \in S' \right\}$$

   for $r_0 \le r \le r_2$;
9. • let t_h be the number of h-points in S' with $b_n^h < r_0$, for $0 \le h \le g$;
10. • calculate $index_h(r_0)$ as $index_h(r_0) = r_0 + 1 - index_h(r_1') - t_h$, for $0 \le h \le g$ [see explanation];
11. • calculate $n\_index_h(r_0)$ as $n\_index_h(r_0) = index_h(r_0) - index_0(r_0)$, for $0 \le h \le g$;
12. • run Algorithm FSW from rows r_0 to r_2 starting from the dummy state

$$\begin{array}{c} n\_index_h' = n\_firstd_h' = n\_index_h(r_0), \\ n\_ocdiag_h' = n\_occol_h' = \emptyset, \end{array} \qquad 0 \le h \le g,$$

   using $insert(r_0), \ldots, insert(r_2)$; get set T of 4-tuples $(h, n, a_n^h, {b'}_n^h)$ for $r_0 \le a_n^h \le r_2$;
13. • return $index_h(r_1)$ for $0 \le h \le g$, and the 4-tuples in T with $r_1 \le a_n^h \le r_2$.

Table 3.3: Algorithm RW.

left of the main diagonal (see Figure 3.2). By the definition of L and H (line 2), all these reflected h-points lie between columns r'_1 and r'_2 (line 3).

Of course, finding these reflected h-points is equivalent to finding the unreflected originals, which we do by calling Algorithm RW recursively.

Once we have the reflected h-points, constructing $insert(r_0), \dots, insert(r_2)$ is simple, since every row r, $r_0 \le r \le r_2$ that does not contain a reflected h-point must contain an h-point, and vice versa.

The formula for $index_h(r_0)$ at line 10 requires an explanation. Recall that $index_h(r_0)$ is the number of h-points on rows $0, \dots, r_0 - 1$. Let k_h be the number of *reflected* h-points on rows $0, \dots, r_0 - 1$. Then

$$index_h(r_0) + k_h = r_0 + 1,$$

since there is one h-point on the main diagonal, which is counted twice.

To calculate k_h, note that all reflected h-points before column r'_1 lie below row r_0, and there are $index_h(r'_1)$ such reflected h-points. Therefore,

$$k_h = index_h(r'_1) + t_h,$$

where t_h is the number of reflected h-points below row r_0 that lie on or after column r'_1, as in line 9. Putting all this together, we get

$$index_h(r_0) = r_0 + 1 - index_h(r'_1) - t_h,$$

as in line 10.

The above calculation is only valid if the h-point on the main diagonal lies before column r'_1. That is why we check for the case $r'_1 \le 2g$ at line 4 (recall Lemma 2.2).

Finally, the check $r'_2 \ge r_0$ at line 4 prevents making a recursive call if the new interval $[r'_1, r'_2]$ is not strictly below the old interval $[r_0, r_2]$.

If we cannot make a recursive call (for either of the two possible reasons), we calculate the h-points in the standard way, using Algorithm WSG starting from row 0.

3.3.1 Algorithm RW's running time

If we want to use Algorithm RW to calculate a single point p_n^g, we must first estimate its row number a_n^g. By Theorem 2.3, we can bound it between $r_1 = \lceil \phi n + L' \rceil$ and $r_2 = \lfloor \phi n + H' \rfloor$ for some constants L', H' that depend on g.

Whenever Algorithm RW makes a recursive call, it goes from an interval of length $\Delta r = r_2 - r_1$ to an interval of length $\Delta r' = r'_2 - r'_1$, where

$$\Delta r' = \phi^{-1} \Delta r + (H - L + \phi^{-1} R_g)$$

(ignoring the rounding to integers). After repeated application of this transformation, the interval length converges to the constant

$$\Delta r^* = \phi^2 (H - L) + \phi R_g.$$

The number of recursive calls is $O(\log n)$, since each interval is ϕ times closer to the origin than its predecessor. And at the last recursive call, Algorithm WSG needs to be run for at most a bounded number of rows, which takes constant time.

Therefore, the total running time of Algorithm RW is $O(f(g) \log n)$ for some function f that depends on R_g, α_g, and β_g, as claimed.

3.4 Application of Algorithm RW

Let us discuss how to apply Algorithm RW to the problem raised in the Introduction, namely playing the sum of Wythoff's game with a Nim pile.

Suppose we are given the sum of a game of Wythoff in position (a, b), $a \leq b$, with a Nim pile of size g, where a and b are very large and g is relatively small. Suppose Conjecture 3.10 is true for this value of g, and we know the value of R_g.

We have to determine whether $\mathcal{G}(a, b)$ is larger than, smaller than, or equal to g. By Theorem 2.3, we can only have $\mathcal{G}(a, b) \leq g$ if $|b - \phi a| \leq k_g$ for some constant k_g that depends on α_g and β_g.

Therefore, if $|b - \phi a| > k_g$, we know right away that $\mathcal{G}(a, b) > g$. If, on the other hand, $|b - \phi a| \leq k_g$, then we use Algorithm RW to find all the h-points, $h \leq g$, in the vicinity of (a, b), and we check whether (a, b) is one of them.

If we find that $\mathcal{G}(a, b) = g$, then the overall game is in a P-position, so there is no winning move. If $\mathcal{G}(a, b) = h < g$, then our winning move is to reduce the Nim pile to size h. And if $\mathcal{G}(a, b) > g$, then our winning move consists of moving in Wythoff's game to a position with Grundy value g. There are at most three alternatives to check—moving horizontally, vertically, or diagonally. Therefore, the winning move can be found by making at most three calls to Algorithm RW with bounded-size intervals, as shown schematically in Figure 3.3.

3.5 Algorithm RW in practice

We wrote a C++ implementation of Algorithm RW. For the constants L and H we used experimental lower and upper bounds for $a_n^g - \phi^{-1} b_n^g$, to which we added safety margins. For the constant R_g we added a safety margin to the values shown in Table 3.2.

We checked our program's results against those produced by the non-recursive Algorithm WSG. The results were in complete agreement as far as we tested.

We also used our recursive program to predict the trillionth g-values for g between 0 and 20. We used $L = -15.0$, $H = 15.0$ (which are a safe distance away from the experimental bounds of -12.4 and 11.3), and $R_{20} = 8000$ (almost twice the value in Table 3.2).

Our predictions are shown in Table 3.4. The program actually performed this calculation in just twenty seconds. These predictions might be verified one day with a powerful computer.

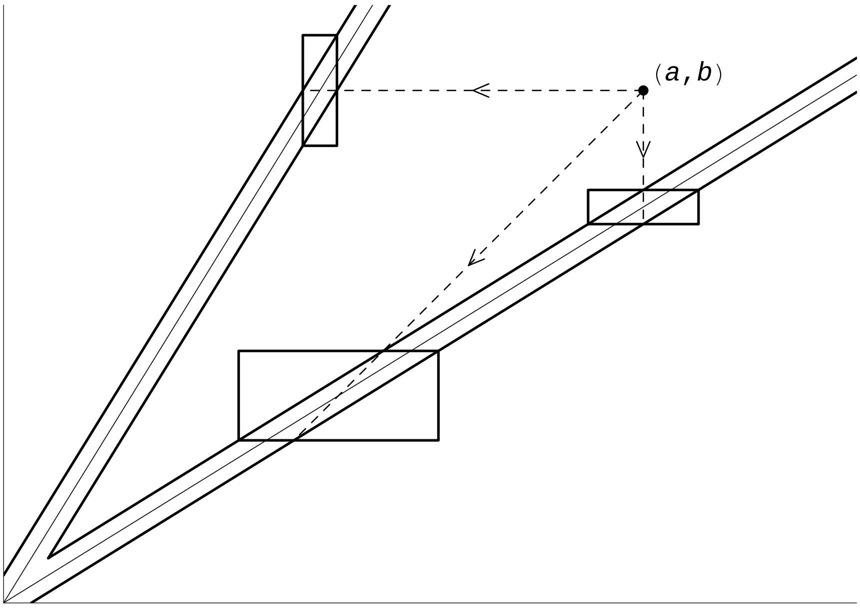


Figure 3.3: Intervals in which to look for a successor with Grundy value g to position (a, b).

$a = 1\,618\,033\,988\,700, \quad b = 2\,618\,033\,988\,700$

g	$p^g_{10^{12}}$	g	$p^g_{10^{12}}$
0	$(a+49, b+49)$	11	$(a+51, b+49)$
1	$(a+50, b+50)$	12	$(a+49, b+53)$
2	$(a+49, b+50)$	13	$(a+50, b+55)$
3	$(a+50, b+49)$	14	$(a+49, b+54)$
4	$(a+50, b+51)$	15	$(a+47, b+51)$
5	$(a+50, b+52)$	16	$(a+49, b+43)$
6	$(a+49, b+51)$	17	$(a+53, b+51)$
7	$(a+50, b+46)$	18	$(a+48, b+52)$
8	$(a+51, b+51)$	19	$(a+52, b+61)$
9	$(a+51, b+56)$	20	$(a+49, b+39)$
10	$(a+49, b+52)$		

Table 3.4: Predicted value of $p^g_{10^{12}}$ for $0 \leq g \leq 20$.

To conclude, note that if there are only sporadic counterexamples to Conjecture 3.10 for a certain value of R_g, then Algorithm RW is still likely to give correct results in most cases. Algorithm RW will only fail if one of the rows r_0 in the different recursion levels constitutes the initial row of a counterexample to Conjecture 3.10. But, as we showed earlier, the number of recursion levels is logarithmic in the magnitude of the initial parameters.

Appendix A

Proofs

We relegated some proofs in the Thesis to this Appendix.

A.1 Beatty's Theorem

Theorem 1.14 (Beatty [2]) *Let $\alpha, \beta < 1$ be irrational numbers such that $\alpha^{-1} + \beta^{-1} = 1$. Then the sequences $\{\lfloor \alpha n \rfloor\}_{n=1}^{\infty}$ and $\{\lfloor \beta n \rfloor\}_{n=1}^{\infty}$ together contain every positive integer exactly once.*

Proof Consider the reals $S = \{\alpha n, \beta n \mid n \geq 1\}$. Note that none of them are integers. Let $k \geq 1$ be an integer. The number of reals in S that are $< k$ is exactly

$$N(k) = \lfloor k/\alpha \rfloor + \lfloor k/\beta \rfloor.$$

But $k/\alpha + k/\beta = k$. Since $x - \lfloor x \rfloor < 1$ for any x, and $N(k)$ is an integer, we must have $N(k) = k - 1$. Therefore, $N(k+1) = k$, so there is exactly one real in S between k and $k+1$ for every $k \geq 1$. The claim follows. ∎

A.2 Non-attacking queens on a triangle

Theorem 2.4 ([16]) *The maximum number of non-attacking queens that can be placed on a triangular lattice of side n is exactly*

$$q(n) = \left\lfloor \frac{2n+1}{3} \right\rfloor.$$

Proof We count in two different ways the total number of attacks by queens on cells. Let s be the number of times a cell is attacked by (collinear with) a queen, summed over all the cells of the board. By this definition, a queen attacks its own cell three times—once for each direction of movement.

It is easy to check that each queen contributes exactly $2n+1$ to s, no matter where it is placed. Therefore, if there are q non-attacking queens on the board,

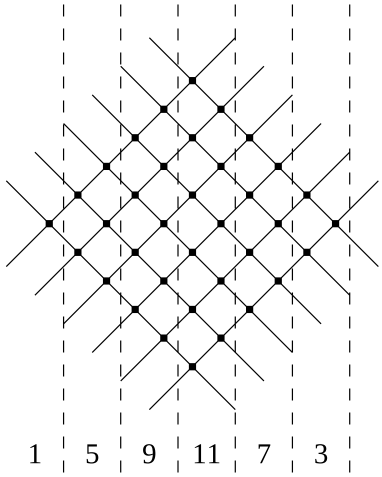


Figure A.1: Intersections between two sets of lines, partitioned into layers.

then

$$s = (2n+1)q. \tag{A.1}$$

On the other hand, each cell can be attacked at most three times. And since the board contains $n(n+1)/2$ cells, we have

$$s \leq \frac{3n(n+1)}{2}. \tag{A.2}$$

Combining (A.1) and (A.2), we get $q \leq 3(n+1)/4$, which is not tight enough.

We can get a tighter bound by bounding the number of cells that can be actually attacked three times. Suppose q non-attacking queens are placed on an arbitrarily large board. Trace a line through each queen along each of the three directions of movement. Then, the cells attacked three times are exactly those on which three lines intersect.

Consider two of the sets of q lines. The distance between adjacent lines may vary, but the lines will always produce a rectangle of $q \times q$ intersections, as shown in Figure A.1. Partition the intersections into "layers" as shown by the dashed lines. The numbers of intersections in the layers will always be $1, 3, 5, \ldots, 2q-1$, the first q odd numbers (as can be easily shown by induction).

Now, consider the third set of lines (which would be horizontal in the figure). Each such line can cross at most one intersection per layer, since no layer contains two horizontally-aligned intersections.

Therefore, the most we can do is cross q intersections with the first horizontal line, $q-1$ intersections with the second, again $q-1$ intersections with the third, and so on, until we use up all the q horizontal lines. Thus, the number of triple intersections is, for q even, no more than

$$q + 2\Big((q-1) + (q-2) + \cdots + (\frac{q}{2}+1)\Big) + \frac{q}{2} = \frac{3q^2}{4};$$

and for q odd, no more than

$$q + 2\Big((q-1) + (q-2) + \cdots + \frac{q+1}{2}\Big) = \frac{3q^2+1}{4}.$$

Thus, in either case, the number of triple intersections is no more than $(3q^2 + 1)/4$.

Therefore, s, the total number of times a cell is attacked by a queen, is bounded by

$$s \leq \frac{2n(n+1)}{2} + \frac{3q^2+1}{4}. \tag{A.3}$$

(We add 2 for each cell of the board, and then 1 for each cell that can be attacked a third time.)

Combining (A.1) and (A.3), and solving this quadratic inequality for q, we get

$$q \leq \frac{2n+1}{3} \quad \text{or} \quad q \geq 2n+1.$$

Obviously, q cannot be larger than n, and q must be an integer. Therefore,

$$q \leq \Big\lfloor \frac{2n+1}{3} \Big\rfloor. \tag{A.4}$$

We end by showing that the bound given by equation (A.4) is in fact tight. Figure A.2 illustrates how to place $(2n+1)/3$ queens on a board of side n, when $n \equiv 1 \pmod 3$. If we number the board's columns 1 through $2n-1$ as shown, then the queens are placed on all cells in columns number $(2n+1)/3$ and $(4n+2)/3$.

For the cases $n \equiv 0$ and $n \equiv 2 \pmod 3$, we can use the same configuration and remove the board's bottom one or two rows, respectively, along with their queens. Therefore, we can always place $\lfloor (2n+1)/3 \rfloor$ queens on a board of side n. ∎

A.3 Lemmas of Section 2.6

Lemma 2.18 *Let $\{x_n\}$ be a sequence that satisfies $x_{n+1} \sim cx_n$ for some $|c| < 1$. Then $\{x_n\}$ is bounded as a sequence.*

Proof Let k be the constant such that $|x_{n+1} - cx_n| \leq k$, as in Definition 2.17; and let $d = k/(1-|c|)$. Let I be the real interval $I = [-d, d]$. It can be verified that if $x_n \in I$, then $x_{n+1} \in I$, and if $x_n \notin I$, then

$$|x_{n+1}| - d \leq |c| \, (|x_n| - d);$$

in other words, the distance between x_n and I is multiplied by a factor of at most $|c|$. Therefore, since $|c| < 1$, the sequence $\{x_n\}$ is "attracted" towards I. ∎

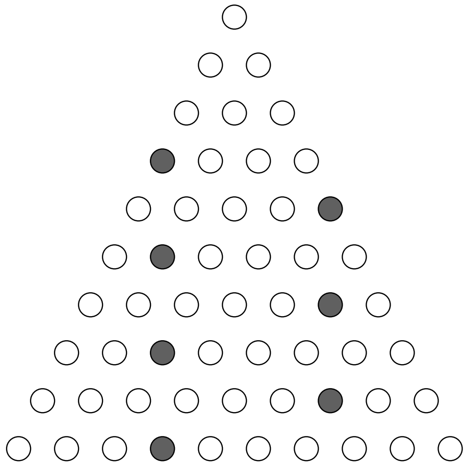


Figure A.2: $(2n+1)/3 = 7$ non-attacking queens on a board of side $n = 10$.

Lemma 2.20 *Regarding the sequences* $\{a_n\}$ *and* $\{b_n\}$ *in the statement of Theorem 2.19:*

(a) There is a constant k *such that for all* n*, the number of* $b_m > b_n$, $m < n$, *is at most* k.

(b) There is a constant k' *such that for all* n*, the number of* $b_m < b_n$, $m > n$, *is at most* k'.

(c) $a_n \sim a_{n-1}$ *and* $b_n \sim b_{n-1}$.

Proof According to condition 3 in the Theorem, let L and H be such that

$$n + L \leq b_n - a_n \leq n + H \quad \text{for all } n.$$

Suppose $b_m > b_n$, with $m < n$. Then,

$$\begin{aligned} b_n &\geq a_n + n + L, \\ b_m &\leq a_m + m + H. \end{aligned}$$

But $a_n - a_m \geq n - m$, so

$$0 < b_m - b_n \leq 2(m-n) + H - L,$$

so

$$m > n - \frac{H-L}{2}.$$

Therefore, for every n there are at most $(H-L)/2$ possible values for m. This proves claim (a). Claim (b) follows analogously.

For claim (c), let $k = a_n - a_{n-1}$. Then, by condition 2 in the Theorem, the interval

$$I = \{a_{n-1} + 1, \ldots, a_n - 1\},$$

whose size is $k-1$, is a subset of B. Let i be the smallest index and j the largest index such that both b_i and b_j are in I. Then $b_j - b_i \leq k - 2$ and $j - i \geq k - 2$. But

$$\begin{aligned} b_j &\geq a_j + j + L, \\ b_i &\leq a_i + i + H, \quad \text{and} \end{aligned}$$

$$a_j - a_i \geq j - i,$$

so

$$k - 2 \geq b_j - b_i \geq 2(j - i) + L - H \geq 2k - 4 + L - H,$$

so

$$k \leq H - L + 2,$$

proving that $a_n \sim a_{n-1}$. And

$$b_n - b_{n-1} \leq (a_n + n + H) - (a_{n-1} + n - 1 + L) \leq 2(H - L) + 3;$$

$$b_n - b_{n-1} \geq (a_n + n + L) - (a_{n-1} + n - 1 + H) \geq L - H + 2,$$

since $a_n - a_{n-1} \geq 1$. Therefore, $b_n \sim b_{n-1}$. ∎

Lemma 2.21 *There exists an integer n_1 such that $f(n) > n$ for all $n \geq n_1$.*

Proof The number of a's smaller than a_n is exactly n, so the number of b's smaller than a_n is $\sim a_n - n$. On the other hand, $b_n \sim b_{n-1}$, so the number of b's smaller than a_n goes to infinity as $n \to \infty$. Therefore, $a_n - n \to \infty$. But $a_n \sim f(n)$ (equation (2.6) in Section 2.6). Therefore, $f(n) - n \to \infty$, which is even stronger than our claim. ∎

Lemma 2.22 *The sequence $\{y_j\}$ defined in equation (2.8) satisfies $y_{j+1} \sim \phi^{-1} y_j$.*

Proof For each $j \geq 0$, let $m(j)$, $n_j \leq m(j) \leq n_{j+1}$, be the index for which the maximum $y_j = |x_{m(j)}|$ is achieved.

For each $j \geq 1$, let $p(j)$ be the largest integer in the range $n_j \leq p(j) \leq n_{j+1}$ for which

$$f(p(j)) \leq m(j+1). \tag{A.5}$$

(Note that $p(j)$ always exists, since $m(j+1) \geq n_{j+1} = f(n_j)$. Further, if $m(j+1) = n_{j+2}$ then $p(j) = n_{j+1}$ and we have equality in (A.5).)

By Lemma 2.20(c) and equation (2.6) we have $f(p(j)) \sim f(p(j)+1)$, so by the definition of $p(j)$,

$$f(p(j)) \sim m(j+1).$$

Similarly, we have $x_n \sim x_{n+1}$, so $x_n \sim x_{n+k}$ for every constant k.

Therefore, using equation (2.7),

$$\begin{aligned} y_{j+1} = |x_{m(j+1)}| &\sim |x_{f(p(j))}| \\ &\sim \phi^{-1}|x_{p(j)}| \\ &\leq \phi^{-1}|x_{m(j)}| \\ &= \phi^{-1}y_j. \end{aligned}$$

So

$$y_{j+1} \sim h_j \leq \phi^{-1}y_j \tag{A.6}$$

for some sequence $\{h_j\}$.

Now, it is not hard to show that if $f(n) > f(n')$, $n < n'$, then $f(n) - f(n')$ is bounded. Therefore, define f', a slight modification of f, as:

$$f'(n) = \left\{ \begin{array}{ll} n_{j+1}, & \text{if } f(n) < n_{j+1}, \\ f(n), & \text{if } n_{j+1} \leq f(n) \leq n_{j+2}, \\ n_{j+2}, & \text{if } n_{j+2} < f(n), \end{array} \right\} \quad \text{for } n_j \leq n \leq n_{j+1}. \tag{A.7}$$

Then $f'(n) \sim f(n)$, and $f'(n)$ is always in the interval $n_{j+1} \leq f'(n) \leq n_{j+2}$ for $n_j \leq n \leq n_{j+1}$.

Therefore,

$$\begin{aligned} y_{j+1} \geq |x_{f'(m(j))}| &\sim |x_{f(m(j))}| \\ &\sim \phi^{-1}|x_{m(j)}| \\ &= \phi^{-1}y_j; \end{aligned}$$

so

$$y_{j+1} \geq h'_j \sim \phi^{-1}y_j \tag{A.8}$$

for some sequence $\{h'_j\}$.

Equations (A.6) and (A.8) together imply that $y_{j+1} \sim \phi^{-1}y_j$. ∎

Bibliography

[1] E. R. Berlekamp, J. H. Conway and R. K. Guy, *Winning Ways for your Mathematical Plays*, vol. 1, second edition, A K Peters, Natick, MA, 2001. (First edition: Academic Press, New York, 1982.)

[2] S. Beatty, Problem 3173, *Amer. Math. Monthly*, vol. 33 (1926), p. 159; vol. 34 (1927), pp. 159–160.

[3] U. Blass and A. S. Fraenkel, The Sprague–Grundy function for Wythoff's game, *Theoret. Comput. Sci.*, vol. 75 (1990), pp. 311–333.

[4] C. L. Bouton, Nim, A game with a complete mathematical theory, *Ann. of Math.*, 2nd Ser., vol. 3 (1901–1902), pp. 35–39.

[5] J. H. Conway, *On Numbers and Games*, second edition, A K Peters, Natick, MA, 2000.

[6] H. S. M. Coxeter, The golden section, phyllotaxis and Wythoff's game, *Scripta Math.*, vol. 19 (1953), pp. 135–143.

[7] A. P. Domoryad, *Mathematical Games and Pastimes*, Pergamon Press, Oxford, 1964.

[8] A. Dress, A. Flammenkamp and N. Pink, Additive periodicity of the Sprague–Grundy function of certain Nim games, *Adv. in Appl. Math.*, vol. 22 (1999), pp. 249–270.

[9] A. S. Fraenkel, *Adventures in Games and Computational Complexity*, to be published.

[10] M. Gardner, Nim and Hackenbush, in: *Wheels, Life, and other Mathematical Amusements*, Freeman, New York, 1983.

[11] M. Gardner, Wythoff's Nim, in: *Penrose Tiles to Trapdoor Ciphers*, The Mathematical Association of America, Washington, 1997.

[12] P. M. Grundy, Mathematics and games, *Eureka*, vol. 2 (1939), pp. 6–8.

[13] H. Landman, A simple FSM-based proof of the additive periodicity of the Sprague–Grundy function of Wythoff's game, in: *More Games of No Chance*, MSRI Publications, vol. 42, Cambridge University Press, Cambridge, 2002, pp. 383–386.

[14] H. Landman, Personal communication.

[15] mathtalk-ga, Chinese origins of mathematical game, Google Answers, `http://answers.google.com/answers/threadview?id=378106` .

[16] G. Nivasch and E. Lev, Non-attacking queens on a triangle, submitted for publication.

[17] N. Pink, *Über die Grundyfunktionen des Wythoffspiels und verwandter Spiele*, dissertation, Universität Heidelberg, 1993.

[18] R. Silber, Wythoff's Nim and Fibonacci representations, *Fibonacci Quart.*, vol. 15 (1977), pp. 85–88.

[19] R. Sprague, Über mathematische Kampfspiele, *Tôhoku Math. J.*, vol. 41 (1936), pp. 438–444.

[20] E. W. Weisstein, Natural Number, MathWorld, `http://mathworld.wolfram.com/NaturalNumber.html` .

[21] W. A. Wythoff, A modification of the game of Nim, *Niew Archief voor Wiskunde*, vol. 7 (1907), pp. 199–202.

[22] A. M. Yaglom and I. M. Yaglom, *Challenging Mathematical Problems with Elementary Solutions*, vol. 2, Holden-Day, San Francisco, 1967.