Harnessing the Unwieldy MEX Function

Aviezri S. Fraenkel

aviezri.fraenkel@weizmann.ac.il

Udi Peled

udi@udipeled.co.il

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 76100, Israel

Abstract

A pair of integer sequences that split $\mathbb{Z}_{>0}$ is often—especially in the context of combinatorial game theory—defined recursively by $A_n = \operatorname{mex}\{A_i, B_i : 0 \le i < n\}$, $B_n = A_n + C_n$ ($n \ge 0$), where mex (**M**inimum **EX**cludant) of a subset S of nonnegative integers is the smallest nonnegative integer not in S, and $C : \mathbb{Z}_{\ge 0} \to \mathbb{Z}_{\ge 0}$. Given $x, y \in \mathbb{Z}_{>0}$, a typical problem is to decide whether $x = A_n$, $y = B_n$. For general functions C_n, the best algorithm for this decision problem was until now exponential in the input size $\Omega(\log x + \log y)$. We prove constructively that the problem is actually polynomial for the wide class of *approximately linear* functions C_n. This solves constructively and efficiently the complexity question of a number of previously analyzed *take-away* games of various authors.

Keywords: combinatorial games, mex, efficient algorithm, complexity.

Mathematics Subject Classification: 11B39, 11B75, 68Q25, 91A46

1 Prologue

This paper is about the complexity of combinatorial games. Its main contribution is showing constructively that a large class of games whose complexity was hitherto unknown and its best winning strategy was exponential, is actually solvable in polynomial time.

For many *take-away* games on 2 piles, the set of *losing positions* $\{(a_n, b_n)\}$ $(n \geq 0)$, also called *P-positions*, is given by recursive formulas of the form:

$$\begin{cases} A_n = \operatorname{mex}\{A_i, B_i : 0 \leq i < n\} \\ B_n = A_n + C_n, \end{cases} \tag{1}$$

where mex (**M**inimum **EX**cludant) of a subset S of nonnegative integers is the smallest nonnegative integer not in S, $\{A_n\}$, $\{B_n\}$ $(n \geq 1)$ are *complementary* sets and $C : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. (Later on we define precisely notions alluded to here informally.)

A case in point is *Wythoff's* game, played on two piles of tokens of sizes (x, y), where we may assume $0 \leq x \leq y$. The two players play alternately. There are two types of moves: (i) taking any positive number of tokens from a *single* pile (Nim rule), or (ii) removing the *same* number of tokens from both piles (Wythoff rule). In this and all games considered here, the first player unable to move (because the position is $(0,0)$) loses and the opponent wins. Its P-positions are given by (1) with $C_n = n$. Given a game position (x, y), to decide whether $(x, y) = (A_n, B_n)$ requires the computation of $A_0, B_0, A_1, \ldots, B_{k-1}, A_k$, where k is the largest index such that $A_k \leq x$. This is an exponential computation, since the input length is $O(\log x + \log y)$.

For Wythoff's game, however, there is an explicit formula for the P-positions, namely, $A_n = \lfloor n\varphi \rfloor$, $B_n = \lfloor n\varphi^2 \rfloor$, where $\varphi = \left(1 + \sqrt{5}\right)/2$ is the golden section. Using this formula it is possible to efficiently compute A_n, B_n for any n, and more importantly, to decide efficiently (in polynomial time) whether a given position (x, y) of Wythoff's game is a P-position. Another efficient winning strategy is based on the Fibonacci numeration system. There is a generalization of Wythoff's game where the Wythoff move is relaxed: one can take $k > 0$ from one pile and $\ell > 0$ from the other, provided $|k - \ell| < p$, where p is a fixed integer parameter ($p = 1$ is Wythoff's

game). The P-positions of this *p-Wythoff* game are given by (1) for $C_n = pn$. Also for this case there is an explicit formula of its P-positions: $A_n = \lfloor n\alpha \rfloor$, $B_n = \lfloor n\beta \rfloor$, where $\alpha = 1 - p/2 + \sqrt{1 + (p/2)^2}$, $\beta = \alpha + p$, leading to an efficient computation of A_n, B_n [Fra82]. For a recent and comprehensive survey on the complexity of combinatorial games see [DemHea09]. Previous surveys are in [Dem01], [Fra00], [Fra04-1]. There is an extensive literature on Wythoff's game; here we just cite two of them: [Wyt1907], [Fra82].

For other functions C_n, the present state of affairs is that ad hoc methods have to be devised for each C in order to try and decide whether an efficient winning strategy exists. For example, for $C_n = (s-1)A_n + pn$ where $s, p \in \mathbb{Z}_{>0}$, it was shown in [Fra98] that for $s > 1$, there are no α, β, γ, δ such that $A_n = \lfloor n\alpha + \gamma \rfloor$, $B_n = \lfloor n\beta + \delta \rfloor$. However, in this case a polynomial winning strategy can still be recovered by means of an exotic numeration system. If C_n is not an integer, no efficient strategy is known in general. A case in point is a game considered in [DucGra–], where $C_n = 2\lfloor n/4 \rfloor$. For the case where C is a special algebraic number, it was recently shown in [DucRig–] that an efficient winning strategy exists. But for most cases, nothing is known about the efficiency of winning.

In conclusion, the present state of affairs is that for each function C_n, the corresponding game has to be analyzed and ad hoc methods have to be devised which might or might not produce an efficient strategy.

The main impact of the present paper is the formulation of a constructive recursive algorithm showing that the strategy is efficient for every *approximately linear* function C_n, in particular functions of the form $C_n = k\lfloor \theta n \rfloor$, where $k \in \mathbb{Z}_{>0}$, $\theta \in \mathbb{R}_{>0}$. This solves take-away games considered previously where the complexity analysis was left open, such as in [Fra04-2], [HegLar06] and [DucGra–]. We also prove several approximation results for certain sequences, of independent interest, which we use for formulating our recursive algorithm.

Many complementary sequences are listed in Sloane's on-line encyclopedia of integer sequences. They play a prominent role not only in the theory of combinatorial games, but also in combinatorics of words and spectra of numbers investigations. See [Kim07] for a survey and also [Kim08].

2 Formalizations and Overview

Definition 1 (P- and N-positions)**.** (i) A position from which the player moving first has a winning strategy is called an *N-position* (**N**ext player to move wins). A position from which the player moving second has a winning strategy is called a *P-position* (**P**revious player to move wins).
(ii) A *take-away* game is played on a collection of piles of finitely many tokens. A move consists of selecting a pile and removing from it any positive number of tokens.

Note that player I, starting from a P-position, is doomed to lose the game, assuming player II plays optimally. Hence to find optimal strategies, it suffices to characterize the P-positions, and this characterization should be as simple as possible computationally. One might go about characterizing the N-positions instead, but it's more economical to characterize the P-positions, since they are rare. Intuitively, this follows from the fact that a position is a P-position if and only if *all* its direct followers are N-positions, whereas a position is an N-position if and only if it *has* a direct P-position follower. Formally, this has been proved by Singmaster [Sin81]; see also [Sin82].

Definition 2 (Minimal EXcludant)**.** Let $S \subsetneq Z_{\geq 0}$. Then $\mathrm{mex}(S) := \min(Z_{\geq 0} \setminus S)$.

Notice that by the mex definition, the sets $\{A_n\}_{n\geq 1}$, $\{B_n\}_{n\geq 1}$ in (1) partition the positive integers into two disjoint sets whenever $C_n \geq 1$ ($n \geq 1$). The mex function appears frequently in the study of combinatorial games, hence it is important to understand its behavior and influence on the sequences which it defines.

The game of q-Blocking p-Wythoff's Nim [HegLar06] ($p, q \geq 1$ integer constants) was defined by Hegarty and Larsson. It is p-Wythoff mentioned earlier, with an additional "Muller twist" (see [SmSt02]), namely, if the current position is (k, ℓ), then, before the next move is made, the previous player is allowed to choose up to $(q-1)$ distinct positive integers $t_1, \ldots, t_{q-1} \leq \min\{k, l\}$ and prevent the next player from moving to any position $(k - t_i, \ell - t_i)$. They proved that its P-positions are given by (1) with $C_n = p\,\lfloor n/q \rfloor$ ($n \geq 0$).

Alas, by a result in [BosFra81], there is no representation of the type $\lfloor n\alpha+\beta \rfloor$ for either of the sequences A_n, B_n in the general case. No numeration system seems to help, so there appears to be no efficient method for computing the sequences.

Duchêne and Gravier analyzed the following 2-pile extension of Wythoff's game [DucGra–]: remove any number of tokens from a *single* pile, or remove the same positive *even* number of tokens from both piles. They proved that The P-positions (A_n,B_n) are given by $A_n = F_n$, $B_{4n} = G_{4n}$, $B_{4n+1} = G_{4n+1}$, $B_{4n+2} = G_{4n+3}$, $B_{4n+3} = G_{4n+2}$ for all $n \geq 0$, where $F_n = mex\{P_i, Q_i : 0 \leq i < n\}$, $G_n = F_n + 2\lfloor n/4 \rfloor$ for all $n \geq 0$. So F_n, G_n have the same values as A_n, B_n respectively of q-Blocking p-Wythoff for the special case $p = 2$, $q = 4$. In fact, the Duchêne and Gravier game was one of our original motivations for investigating sequences of this type.

Definition 3 (Approximate Linearity)**.** We call the sequence A *approximately linear* if there exist constants α, u_1, $u_2 \in \mathbb{R}$ such that

$$u_1 \leq a_n - n\alpha \leq u_2 \quad \forall n \geq 0.$$

We say that that u_1, u_2 are *approximation bounds*, and α is the *linearity rate*.

Remark. Notice that α is unique since $\lim_{n\to\infty} a_n/n = \alpha$, but u_1, u_2 are not.

Though there is no known explicit formula for general sequences $\{A_n\}$, $\{B_n\}$ of (1), it was conjectured that for the special case $C_n = p\lfloor n/q \rfloor$ they can be approximated by linear sequences, namely that $A_n - n\alpha$ is bounded for some constant α, and similarly for B_n. Our first result was proving that indeed this approximation holds, by finding bounds depending on p, q. Hegarty [Lar07] independently proved the same approximation for the special case $p = 1$. We then generalized our result to the case where C_n is any approximately linear function.

After approximating A_n, B_n, there still remained the problem of efficiently computing the sequences, which, without an explicit formula, seemed impossible. We should mention that more often than not, for a general function C_n, there is no efficient algorithm for computing the elements of the sequences defined by (1) and deciding whether an arbitrary number belongs to it. The best known algorithms are exponential.

But quite surprisingly, after deepening our understanding of partitions created by the mex function, we indeed found an efficient algorithm. This is our most important result. The algorithm computes sequences for any approximately linear C_n.

In particular, the algorithm solves efficiently the q-Blocking p-Wythoff's game, and also Duchêne's game. Moreover, since most of the results presented here are for general sequences defined by the mex function, and the idea behind the algorithm is basic, we believe that it can be generalized and implemented with some modifications for other sequences. For example, Fraenkel and Krieger [FraKri04] and Sun and Zeilberger [SunZei04] [Sun05] investigated the P-positions of *Fraenkel's N-Heap Wythoff's game* [Fra04-1]. These P-positions are defined similarly to the sequences $\{A_n\}$, $\{B_n\}$, and they share the same essential properties as them. Hence the results presented here can possibly offer some new insights to understanding Fraenkel's N-Heap Wythoff's game.

In §3 below we prove Theorem 1, which gives a basic yet important inequality about partitions. This is followed by Theorem 2, which establishes an important combinatorial equivalence concerning partitions. Both of these results are then used to prove the fundamental Bounded Additivity Theorem 3. Using it and Fekete's Lemma [Fek23] we prove in §4 our main result, the Approximate Linearity Theorem 4, which leads, in §5, to the efficient algorithm, together with an example and its complexity analysis. In §6 we wrap-up with an epilogue.

3 Natural Partitions

Unless otherwise specified, any interval of the form $[a,b]$ denotes the set of integers $\{k \in \mathbb{Z} : a \le k \le b\}$. Analogously for open/half-open intervals.

Definition 4 (Natural Partition)**.** We say that two integer sequences $A := \{a_n\}_{n\ge 0}$, $B := \{b_n\}_{n\ge 0}$ form a *natural partition* if they are strictly increasing sequences which satisfy $A \cup B = \mathbb{Z}_{\ge 0}$, $A \cap B = \{a_0 = b_0 = 0\}$. We also define $C := B - A$.

We begin with a property of natural partitions, which will be used both in the proof of Theorem 3, and in the efficient algorithm for computing sequences. The idea

is that given an element in a sequence, we know its position between two consecutive elements of the complementary sequence.

Theorem 1 (Partition Inequality)**.** *Suppose that A, B form a natural partition. Then*

$$b_{a_k-k} < a_k < b_{a_k-k+1}, \quad a_{b_k-k} < b_k < a_{b_k-k+1} \quad \forall k \geq 1.$$

Proof. How is the interval $I := [1, a_k]$ split between A and B? Since A is increasing, I contains precisely the first k terms of $\{a_n\}$. Hence the remaining $a_k - k$ terms of I must be the leading terms of the increasing sequence B. Therefore $b_{a_k-k} \leq a_k < b_{a_k-k+1}$. By complementarity, actually $b_{a_k-k} < a_k$. The second inequality is derived the same way by splitting $[1, b_k]$ between the two sequences. □

Now for simple bounds on the growth of the smaller sequence of a natural partition.

Corollary 1. *Assume that A, B form a natural partition. If $a_n < b_n$ for all $n \geq 1$, then $n \leq a_n \leq 2n-1$ for all $n \geq 1$.*

Proof. Since A is increasing, obviously $n \leq a_n$. From Theorem 1 we know that $b_{a_n-n} < a_n$, and we assumed $a_n < b_n$, so $b_{a_n-n} < b_n$. Since B is strictly increasing we have $a_n - n < n$ as claimed. □

The following very intuitive theorem gives an equivalence between the number of elements of an increasing sequence in an arbitrary interval of length u, and an inequality about the increments of this interval. We need it for proving Theorem 3.

Theorem 2 (Min-Max-Equivalence)**.** *Assume that $B = \{b_n\}_{n\geq 0}$ is an increasing sequence, satisfying $b_0 = 0$.*

Then the following equivalence holds for any two integers x, u.

$$\min_{L\geq 0} |[L+1, L+u] \cap B| \geq x \quad \Longleftrightarrow \quad \forall n \geq 0, b_{n+x} \leq b_n + u.$$

And similarly also the following equivalence holds

$$\max_{L\geq 0} |[L+1, L+u] \cap B| \leq x \quad \Longleftrightarrow \quad \forall n \geq 0, b_n + u \leq b_{n+x}.$$

Proof. The implication $\min_{L\geq 0}|[L+1,L+u]\cap B|\geq x \quad\Longrightarrow\quad \forall n\geq 0, b_{n+x}\leq b_n+u$ is obvious, since we can take $L=a_n$.

We'll prove the other direction: given $L\geq 0$, take the maximal n such that $b_n\leq L$. So since $b_{n+x}\leq b_n+u\leq L+u$, we get $b_{n+1},b_{n+2},\ldots,b_{n+x}\in[L+1,L+u]$ as claimed.

The proof of the second equivalence is similar: for the first implication take $L=b_n-1$. And for the second direction take minimal $n\geq 0$ such that $L<b_n$. □

Remark. Notice that the equivalence also holds when either x or u are negative integers.

The following result enables us to prove Theorem 4. The proof is by induction, and in order to better understand it, we recommended following the first steps of $m=1,2,3,\ldots$ with specific sequences $\{a_n\}$, $\{b_n\}$ (e.g., $b_n=a_n+\lfloor\sqrt{2}n\rfloor$).

Theorem 3 (Bounded Additivity)**.** *Let A, B be a natural partition. Suppose that $C=\{c_n\}=\{b_n-a_n\}_{n\geq 0}$ is a nonnegative sequence which satisfies, for all $n,m\geq 0$,*

$$c_{n+m-r}\leq c_n+c_m\leq c_{n+m+s},$$

for some integer constants $r,s\geq 0$ satisfying $r-1\leq s$. Then we have for all $n,m\geq 0$,

$$a_{n+m-r}-1\leq a_n+a_m+s-r\leq a_{n+m+s}.$$

Proof. We shall prove the inequality by induction on m. For $m=0$, the inequality is $a_{n-r}-1\leq a_n+s-r\leq a_{n+s}$ for $n\geq 0$, which follows from $r-1\leq s$ and $a_n+s\leq a_{n+s}$ for all $n\geq 0$ (since a_n is increasing).

We assume that the following holds for all $0\leq t\leq m$,

$$a_{n+t-r}-1\leq a_n+a_t+s-r\leq a_{n+t+s}\quad\forall n\geq 0,$$

and we shall prove that it holds for all $0\leq t\leq m+1$.

By the induction assumption we know

$$a_{n+t-r}-1\leq a_n+a_t+s-r\quad\forall n\geq 0, 0\leq t\leq m,$$

and by adding $c_{n+t-r} \leq c_n + c_t$ we get

$$b_{n+t-r} - 1 \leq b_n + b_t + s - r \quad \forall n \geq 0, 0 \leq t \leq m.$$

Now, by Corollary 1 we have $t+1 \leq a_{t+1} \leq 2(t+1)-1$, hence $0 \leq a_{t+1}-(t+1) \leq t \leq m$, and we can switch t with $a_{t+1} - (t+1)$ in the above inequality to get

$$b_{n+a_{t+1}-(t+1)-r} \leq b_n + b_{a_{t+1}-(t+1)} + s - r + 1 \quad \forall n \geq 0, 0 \leq t \leq m.$$

From Theorem 1 we know that $b_{a_{t+1}-(t+1)} < a_{t+1}$, meaning $b_{a_{t+1}-(t+1)} \leq a_{t+1} - 1$, which, concatenated to the right of the above inequality, gives

$$b_{n+a_{t+1}-(t+1)-r} \leq b_n + a_{t+1} + s - r \quad \forall n \geq 0, 0 \leq t \leq m.$$

Since this holds for all $n \geq 0$, Theorem 2 implies that for every fixed $t \leq m$,

$$\min_{L \geq 0} |I_L \cap B| \leq a_{t+1} - (t+1) - r,$$

where $I_L = [L+1, L+a_{t+1}+s-r]$. Now, since A, B form a partition, and $|I_L| = a_{t+1} + s - r$, we get

$$\max_{L \geq 0} |I_L \cap A| \leq (a_{t+1} + s - r) - (a_{t+1} - (t+1) - r) = t + 1 + s.$$

Again by Theorem 2 we get

$$a_n + a_{t+1} + s - r \leq a_{n+t+1+s} \quad \forall n \geq, 0 \leq t \leq m.$$

So we established the right inequality of our goal inequality, namely

$$a_n + a_t + s - r \leq a_{n+t+s} \quad \forall n \geq, 0 \leq t \leq m+1.$$

Now, for the left hand side we use similar arguments. We add $c_n + c_t \leq c_{n+t+s}$ to

the last inequality, to get

$$b_n + b_t + s - r \leq b_{n+t+s} \quad \forall n \geq 0, 0 \leq t \leq m+1.$$

By Corollary 1 we have $t \leq a_t \leq 2t-1$, hence $0 \leq a_t - t + 1 \leq t \leq m+1$, and we can switch t with $a_t - t + 1$ in the above inequality to get

$$b_n + b_{a_t - t + 1} + s - r \leq b_{n + a_t - t + 1 + s} \quad \forall n \geq 0, 0 \leq t \leq m+1.$$

From Theorem 1 we know that $a_t < b_{a_t - t + 1}$, meaning $a_t + 1 \leq b_{a_t - t + 1}$, which, concatenated to the left of the above inequality, gives

$$b_n + a_t + 1 + s - r \leq b_{n + a_t - t + 1 + s} \quad \forall n \geq 0, 0 \leq t \leq m+1.$$

Since this holds for all $n \geq 0$, by the Theorem 2 we get for every fixed $t \leq m+1$,

$$\max_{L \geq 0} |I_L \cap B| \leq a_t - t + 1 + s,$$

where $I_L = [L+1, L + a_t + 1 + s - r]$. Now, since A, B form a partition, and $|I_L| = a_t + 1 + s - r$, we get

$$\min_{L \geq 0} |I_L \cap A| \geq (a_t + 1 + s - r) - (a_t - t + 1 + s) = t - r.$$

Again by Theorem 2 we get

$$a_{n+t-r} \leq a_n + a_t + 1 + s - r \quad \forall n \geq, 0 \leq t \leq m+1.$$

So we also established the left inequality of our goal inequality, hence the induction is complete. □

Remark. The theorem's conclusion also holds for the sequence B, i.e., for all $n, m \geq 0$ $b_{n+m-r} - 1 \leq b_n + b_m + s - r \leq b_{n+m+s}$. To see this, just add the inequality assumption of the sequence C to the inequality conclusion of a_n.

We can change the indices in the theorem's conclusion to get $a_{n+m} \le a_n + a_{m+r} + s - r + 1 \le a_{n+m+a+s} + 1$ for all $n, m \ge 0$. But we do not want to be concerned with the exact indices and constants of such inequalities (though these were essential for the theorem's proof), so we state a very simple lemma.

Lemma 1. *Assume that $\{a_n\}_{n\ge 0}$ is a real sequence, which for some constants $r, s \in \mathbb{Z}_{\ge 0}$ and $u, v \in \mathbb{R}$, satisfies*

$$a_{n+m} \le a_n + a_{m+r} + u \le a_{n+m+s} + v \quad \forall n, m \ge 0.$$

Then there exist constants $D, E \in \mathbb{R}$, such that

$$a_{n+m} + D \le a_n + a_m \le a_{n+m} + E \quad \forall n, m \ge 0.$$

Proof. By the left inequality of the assumption we know that $a_{t+r} \le a_t + a_{2r} + u$ for all $t \ge 0$, so by taking $t = m$ together with the assumption we get $a_{n+m} \le a_n + a_{m+r} + u \le a_n + a_m + a_{2r} + 2u$. We can take $D = -a_{2r} - 2u$ to fit our goal.

Now, again by the left inequality of the assumption we know that $a_{t+s} \le a_t + a_{s+r} + u$ for all $t \ge 0$. By taking $t = n + m$ we can continue the right inequality of the assumption as $a_n + a_{m+r} + u \le a_{n+m+s} + v \le a_{n+m} + a_{r+s} + u + v$.

By taking $n = 0$ in the assumption we get $a_m \le a_0 + a_{m+r} + u$, and together with the previous inequality we get $a_n + a_m - a_0 \le a_n + a_{m+r} + u \le a_{n+m} + a_{r+s} + u + v$. So by both ends of this inequality we can take $E = a_0 + a_{r+s} + u + v$, and we're done. □

4 Approximate Linearity

We use a well known lemma of Fekete [Fek23].

Lemma 2 (Fekete's Lemma)**.** *Let $\{s_n\}_{n\ge 0}$ be a super-additive sequence of real numbers, i.e., $s_n + s_m \le s_{n+m}$ for all $n, m \ge 0$. Then $\lim_{n\to\infty} s_n/n$ exists, and is equal to $\sup_{n\ge 1} s_n/n \in (-\infty, \infty]$.*

From this lemma we derive the following corollary, which can be applied directly to the results of Theorem 3.

Corollary 2 (Fekete's Corollary)**.** *Assume that A is a sequence of real numbers which satisfies, for some constants $u, v \in \mathbb{R}$,*

$$u \leq a_n + a_m - a_{n+m} \leq v \quad \forall n, m \geq 0.$$

Then A is approximately linear, with approximation bounds u, v.

Proof. We have $(a_n - v) + (a_m - v) \leq a_{n+m} - v$ for $n, m \geq 0$, so by applying Fekete's Lemma to the sequence $a_n - v$ we get

$$\sup_{n \geq 1} \frac{a_n - v}{n} = \lim \frac{a_n - v}{n} = \alpha \in (-\infty, \infty].$$

Hence for $n \geq 1 \quad a_n - v \leq n\alpha$, i.e., $a_n - n\alpha \leq v$.

Similarly, $(u - a_n) + (u - a_m) \leq u - a_{n+m}$ for $n, m \geq 0$, so by Fekete's Lemma

$$\sup_{n \geq 1} \frac{u - a_n}{n} = \lim \frac{u - a_n}{n} = \alpha' \in (-\infty, \infty].$$

Hence for $n \geq 1 \quad u - a_n \leq n\alpha'$, i.e., $u \leq a_n + n\alpha'$.

Lastly, notice that $\alpha' = \lim(u - a_n)/n = -\lim a_n/n = -\lim(a_n - v)/n = -\alpha$, hence $\alpha \in \mathbb{R}$. And in total we get $u \leq a_n - n\alpha \leq v$, as claimed. □

Using Theorem 3 together with Fekete's Lemma and their corollaries, we get Theorem 4, which is the main theorem of this paper.

Theorem 4 (Approximate Linearity)**.** *Let A, B be a natural partition. Then A and B are both approximately linear if and only if $C = B - A$ is approximately linear.*

Proof. Assume that A, B are approximately linear, i.e., $u_1 \leq a_n - n\alpha \leq u_2$ and $v_1 \leq b_n - n\beta \leq v_2$ for suitable constants u_1, u_2, v_1, v_2. Then we easily get $v_1 - u_2 \leq (b_n - a_n) - n(\beta - \alpha) \leq v_2 - u_1$, which means that C is approximately linear with linearity rate $\beta - \alpha$.

In the other direction, assume that C is approximately linear, i.e., $K_1 \leq c_n - n\gamma \leq K_2$ for constants K_1, K_2. We would like to use Theorem 3, so we first look for integer constants $r, s \geq 0$. We take an integer $r \geq 0$, which satisfies $r \geq (K_2 - 2K_1)/c$. Thus,

$c_{n+m-r} \le (n+m-r)c+K_2 \le nc+K_1+mc+K_1 \le c_n+c_m$. We also take some integer $s \ge 0$ which satisfies $s \ge (2K_2-K_1)/c$. Then $c_n+c_m \le nc+K_2+mc+K_2 \le (n+m+s)c+K_1 \le c_{n+m+s}$. But recall that the theorem also requires $s \ge r-1$, so we choose s that also satisfies this.

In conclusion, $c_{n+m-r} \le c_n+c_m \le c_{n+m+s}$, so by Theorem 3 and Lemma 1 we get $u_1 \le a_n+a_n-a_{n+m} \le u_2$ for some constants u_1, u_2, and similarly for B. Hence Fekete's Corollary implies that A, B are approximately linear. □

Remark. If we know the approximation bounds K_1, K_2 and the linearity rate $\gamma = \lim c_n/n$, then by Theorem 3's conclusion and the proof of Lemma 1, we can compute approximation bounds for the sequences A and B after choosing — preferably minimal — r,s. Thus we regard them as known constants.

Also the linearity rates α, β are known. Since A, B form a natural partition, and since $\lim a_n/n = \alpha, \lim b_n/n = \beta$, density considerations imply $\alpha^{-1}+\beta^{-1} = 1$. Also notice that $c_n = b_n - a_n$ implies $\beta-\alpha = \gamma$, so we find α,β by solving the quadratic equation $\alpha^{-1}+(\alpha+\gamma)^{-1} = 1$. For $c_n = k\lfloor \theta n \rfloor$ we have $c = k\theta$, so doing so gives $a = (2-k\theta+\sqrt{k^2\theta^2+4})/2$, $b = (2+k\theta+\sqrt{k^2\theta^2+4})/2$.

If $0 < \theta < 1$, then $b_i = a_i = i$ for $0 \le i < 1/\theta$, thus $a_{i_0} = i_0$, $b_{i_0} > i_0$ for $i_0 = \lceil 1/\theta \rceil$. It follows that the sequences $a'_n = a_{n+i_0-1}-i_0+1$, $b'_n = b_{n+i_0-1}-i_0+1$ split the integers $\ge i_0$. The difference $c'_n = b'_n - a'_n = b_{n+i_0-1} - a_{n+i_0-1} = k\lfloor \theta(n+i_0-1) \rfloor$ is approximately linear, so by Theorem 4, $\{a'_n\}$, $\{b'_n\}$ are approximately linear. Definition 3 then implies that also A, B are approximately linear.

5 Efficient Algorithm for Computing Sequences

5.1 Preparation

We deal with sequences A, B which form a natural partition, and assume that C is approximately linear with known approximating bounds K_1, K_2, and known linearity rate $\gamma = \lim c_n/n$. Then A, B are also approximately linear by Theorem 4, and as we demonstrate below, we can compute some integer approximating bounds

u_1, u_2, v_1, v_2, and compute $\alpha = \lim a_n/n$, $\beta = \lim b_n/n$ by the formula $1 \pm \gamma/2 + \sqrt{1 + (\gamma/2)^2}$.

Now, given a pair of integers (x, y), we would like to check whether $(x, y) = (a_n, b_n)$ for some n. If indeed $x = a_n$, then $(x - u_2)/\alpha \le n \le (x - u_1)/\alpha$. So we need to check whether $x = a_n$ for only a fixed number of n's (at most $(u_2 - u_1)/\alpha$ such n's). If $x = a_n$ for some n, we are just left to check if $y = b_n$ in order solve the decision problem. Notice that if we know $x = a_n$, then checking whether $y = b_n$ can be done by checking whether $y - x = c_n$, thus it depends on the complexity of computing c_n.

So, we need to find an efficient method for computing the elements of a_n given n. The naïve recursive method for computing the elements of a_n, b_n is very slow and inefficient: its complexity is $O(n)$, whereas the input size is of complexity $O(\log n)$ in succinct representation. Hence the naïve method is exponential in the input size.

5.2 Idea

The algorithm computes $a_m, a_{m+1}, \ldots, a_n$ for given m, n, by operating as follows. If for some r, s we can compute $b_r, b_{r+1} \ldots, b_s$, where $b_r < a_m$ and $a_n < b_s$. Then since A, B form a natural partition, we can compute $a_m, \ldots, a_n$ by complementing the interval $[b_r, b_s]$ with elements of $\{a_i\}$. Notice, however, that the first element of A in the interval $[b_r, b_s]$ will not be necessarily a_m, but rather $a_{b_r - r + 1}$ by Theorem 1. The last element of A in the interval will be $a_{b_s - s}$. So since we know the start and end indices, namely $b_r - r + 1$ and $b_s - s$, we can immediately locate the desired $a_m, \ldots, a_n$ inside $a_{b_r - r + 1}, \ldots, a_{b_s - s}$.

The problem that remains is to find r, s, which are easy to compute, and that satisfy $b_r < a_m, a_n < b_s$. Moreover, $b_r, \ldots, b_s$ should be efficiently computable. First we deal with finding such r. By Theorem 1 we can take any $r \le a_m - m$, and it will satisfy $b_r < a_m$. But we have not yet computed a_m, so how can we find $r \le a_m - m$?

By the approximate linearity of $\{a_i\}$ we know that $m(\alpha - 1) + u_1 \le a_m - m$, so we can take $r = \lfloor m(\alpha - 1) + u_1 \rfloor$ to satisfy $r \le m(\alpha - 1) + u_1 \le a_m - m$, hence $b_r < a_m$. By similar reasoning, for $s = \lceil n(\alpha - 1) + u_2 \rceil + 1$ we have $a_n - n + 1 \le n(\alpha - 1) + u_2 + 1 \le s$, hence $a_n < b_s$ (Theorem 1).

So we got easy formulas for r, s, which satisfy $b_r < a_m, a_n < b_s$. But how can we

easily compute $b_r, \ldots, b_s$?

Notice that it's enough to know $a_r, a_{r+1}, \ldots, a_s$, and then we can compute directly $b_i = a_i + c_i$. In order to compute $a_r, \ldots, a_s$, we simply recursively call the main part of the algorithm, with r, s instead of m, n, and we're done.

We now turn to formulate the algorithm, afterwards we show how it works on a specific example, and lastly we analyze its complexity.

Remark. Though the algorithm is written here recursively, it is easy to implement it only with loops, and without recursion.

5.3 The Algorithm

Algorithm 1 ComputeSequenceA(m, n)

Input: Two indices m, n (integers).

Output: The elements $a_m, a_{m+1}, \ldots, a_n$.

1: If $m \leq 0$ then
2: For $j = 0, \ldots, n$ do
3: $a_j \leftarrow \operatorname{mex}\{a_i, b_i : 0 \leq i < j\}$.
4: $b_j \leftarrow a_j + c_j$.
5: End for
6: Else $\{m > 0\}$
7: Calculate $r \leftarrow \lfloor m(\alpha - 1) + u_1 \rfloor$ and $s \leftarrow \lceil n(\alpha - 1) + u_2 \rceil + 1$.
8: $a_r, \ldots, a_s \leftarrow$ ComputeSequenceA(r, s).
9: For $j = r, \ldots, s$ do
10: $b_j \leftarrow a_j + c_j$.
11: End for
12: Calculate $a_{b_r - r + 1}, \ldots, a_{b_s - s}$ as the complemented elements of $b_r, \ldots, b_s$ in the interval $[b_r, b_s]$.
13: End if
14: Return $a_m, \ldots, a_n$.

5.4 Example

We demonstrate the following case: A, B form a natural partition, and $b_n = a_n + \lfloor 4n/3 \rfloor$, i.e., $c_n = \lfloor 4n/3 \rfloor$. Then $K_1 \le c_n - n\gamma \le K_2$ holds for $K_1 = -1, K_2 = 0, \gamma = 4/3$. In the proof of Theorem 4 we take $r = 2$, $s = 1$ to satisfy $r \ge (K_2 - 2K_1)/\gamma = 3/2$ and $s \ge (2K_2 - K_1)/\gamma = 3/4$ (also $s \ge r - 1$). Then we have $c_{n+m-2} \le c_n + c_m \le c_{n+m+1}$. Note that usually this general method for finding r, s does not give the best (smallest) r, s possible, so one might try to look for better r, s if one wanted to; we don't.

Now, by Theorem 3, $a_{n+m-2} - 1 \le a_n + a_m + 2 - 1 \le a_{n+m+1}$, equivalently $a_{n+m} \le a_n + a_{m+2} + 0 \le a_{n+m+3} + 1$. Then in the proof of Lemma 1 we get that $a_{n+m} + u_1 \le a_n + a_m \le a_{n+m} + u_2$ holds for $u_1 = -a_4$, $u_2 = a_0 + a_5 + 1$. So we compute the first element of A, B by the definition.

n	0	1	2	3	4	5	6	7	…
c_n	0	1	2	4	5	6	8	9	…
a_n	0	1	3	4	6	7	9	10	…
b_n	0	2	5	8	11	13	17	19	…

Thus $u_1 = -6$, $u_2 = 0 + 7 + 1 = 8$, and by Fekete's Corollary we have $-6 \le a_n - n\alpha \le 8$, where $\alpha = 1 - \gamma/2 + \sqrt{1 + (\gamma/2)^2} = \left(1 + \sqrt{13}\right)/3$.

We begin with $m = n = 1000$. By lines 7, 8, the parameters m, n called in the recursion steps are

Recursive step #	0	1	2	3	4	5	6	7
m	1000	529	277	142	69	30	10	-1
n	1000	545	301	171	101	64	44	33

The computation of the first elements a_n, b_n begins at the recursion stop (line 2).

a_0	a_1	a_2	$\cdots$	a_{31}	a_{32}	a_{33}
0	1	3	$\cdots$	48	49	50

Computing $b_i = a_i + c_i$ (line 9) gives

b_0	a_1	b_1	a_2	a_3	$\cdots$	b_{32}	a_{60}	a_{61}	b_{33}
0		2			$\cdots$	91			94

By complementing the interval $[b_0, b_{33}]$ (line 12) we get $a_1, a_2, \ldots, a_{61}$. From these we extract the desired elements

a_{10}	a_{11}	b_6	a_{12}	$\cdots$	b_{23}	a_{43}	a_{44}
15	16		18	$\cdots$		66	67

Computing $b_i = a_i + c_i$

$\vdots$

b_{277}	a_{518}	b_{278}	a_{519}	a_{520}	$\cdots$	b_{300}	a_{561}	a_{562}	b_{301}
794		796			$\cdots$	860			863

Complementing the interval $[b_{277}, b_{301}]$, and extracting desired elements

a_{529}	a_{530}	b_{284}	a_{531}	$\cdots$	b_{291}	a_{544}	a_{545}
812	813		815	$\cdots$		835	836

Computing $b_i = a_i + c_i$:

b_{529}	a_{989}	b_{530}	$\cdots$	a_{1016}	b_{544}	a_{1017}	b_{545}
1517		1519	$\cdots$		1560		1562

Complementing the interval $[b_{529}, b_{545}]$, and extracting the wanted element

a_{1000}
1535

5.5 Complexity Analysis and Validity Proof

Due to the variety of multiplication algorithms, we denote by $M(k)$ the complexity of multiplication of two k-digits numbers. We also denote by $T(k)$ the complexity of computing c_k, and by $Y(k)$ the complexity of computing the multiplication αk to integer precision.

For example, we can trivially choose $M(k) = k^2$. Then, if we take $c_k = \lfloor 4k/3 \rfloor$, we get $T(k) = O(\log k)$. Since $\gamma = \lim c_n/n = 4/3$, we have $\alpha = \left(1+\sqrt{13}\right)/3$. Hence computing αk forces a computation of the square root of a $\log k$-digits integer (plus some computations with lower complexity). Thus $Y(k) = M(\log k) = O(\log^2 k)$ (complexity of square root computation is the same as of multiplication).

Now, for the complexity analysis of the algorithm. Denote by m_i, n_i the parameters m, n called on the i^{th} step of the recursion. So we have $m_0 = n_0 = n$ (to compute a_n we call ComputeSequenceA(n,n)), and line 7 gives the recursive relation $m_{i+1} = \lfloor m_i(\alpha-1)+u_1 \rfloor, n_{i+1} = \lceil n_i(\alpha-1)+u_2+1 \rceil$. From this it's easy to see that $m_{i+1} \le m_i(\alpha-1)+u_1$ and $n_{i+1}-m_{i+1} \le (n_i-m_i)(\alpha-1)+u_2-u_1+3$, which implies

$$\begin{aligned} m_i &\le (\alpha-1)^i \left(n - \tfrac{u_1}{2-\alpha}\right) + \tfrac{u_1}{2-\alpha} \\ n_i - m_i &\le -(\alpha-1)^i \tfrac{u_2-u_1+3}{2-\alpha} + \tfrac{u_2-u_1+3}{2-\alpha}. \end{aligned} \tag{2}$$

Notice from this that m_i decreases at least exponentially ($1 < \alpha < 2$, since $\alpha = 1 - \gamma/2 + \sqrt{1+\left(\gamma/2\right)^2}$), and that though the difference $n_i - m_i$ is increasing, it is bounded by a constant (namely $(u_2-u_1+3)/(2-\alpha)$).

Since $0 < \alpha - 1 < 1$, and $u_1 \le a_0 - 0 = 0$, from (2) we see that the recursion stopping condition of the algorithm $m \le 0$ (line 1), is fulfilled after at most $O(\log n)$ steps of the recursion. In this case, we compute the first elements of a_i directly by definition (line 2). At the stopping condition we know that $m_i \le 0$, and since also the difference $n_i - m_i$ is bounded by a constant, we see that the number of initial n_i elements computed is bounded by a constant. Hence the complexity of the stopping part of the algorithm (line 2) is constant $O(1)$.

On line 7 we have complexity $Y(n)$ (m_i and n_i are at most n). The complexity of line 9 is just $T(n)$, since the loop index runs from m_i to n_i, and we saw that $n_i - m_i$ is bounded by a constant. We also get from this that in line 12 the size of the interval $\left[b_{m_i}, b_{n_i}\right]$ is bounded by a constant (since b_i is approximately linear), hence this line's complexity is $O(1)$.

In conclusion, the algorithm has complexity $O((Y(n)+T(n))\log n)$, which, depending on this complexity, is generally much more efficient then $O(n)$. For the

example in §5.4, it is $O(\log^3 n)$. The arguments in subsections 5.1, 5.2 and 5.5 jointly also establish the validity of the algorithm.

5.6 Epilogue

It is interesting that the most time consuming part of the algorithm is not the recursion (with complexity $O(\log n)$), but the calculations on lines 7 and 9-10, of complexity $O(Y(n)+T(n))$.

Another surprising aspect is that even though it seems at first that the m_i, n_i are growing apart from each other as the recursion progresses, which would affect both the time complexity and the space complexity, this effect vanishes eventually, since the difference $n_i - m_i$ is bounded by a constant, as shown by (2).

On a fundamental level, we attribute the efficiency of the algorithm to three properties of the sequences A, B.

- The two sequences partition the positive integers. This enabled use of Theorem 1 to compute the $\{a_n\}$ elements as the complement of a $\{b_n\}$-interval.
- The sequence $\{a_n\}$ has a simple approximation by another efficiently computable sequence, namely $s_n = n\alpha$. The sequence $S(n) = s_n - n$, is decreasing fast when composed onto itself (in our case $S(n) \approx (\alpha-1)n$, so $S^{(k)}(n) \approx (\alpha-1)^k n$ decreases exponentially).
- The recursive relation between a_n and b_n is simple. That is, knowing a_n, it is easy to compute b_n by the formula $b_n = a_n + c_n$.

One direction of further research is to extend the results to sequences to which the idea of the algorithm can be applied. For example, for $C_n = \zeta \lfloor n\theta \rfloor$, where $\zeta, \theta \in \mathbb{R}_{>0}$. This requires a generalization of the mex function, since the sequences are not integers anymore. It should not be hard to extend our results to multiple sequences that split $\mathbb{Z}_{\geq 1}$.

6 Acknowledgments

We wish to thank Eric Duchêne for being one of the triggers of this research by a question in his 2006 Ph.D. thesis. We owe special thanks to Peter Hegarty and Urban Larsson for a few short but very useful discussions. We also thank Dorit Ron for running high-precision numerical experiments on sequences during the initial phase of this work.

References

[BosFra81] M. Boshernitzan and A.S. Fraenkel, Nonhomogeneous Spectra of Numbers, *Discrete Math 34 (1981)*, pp. 325–327.

[Dem01] E. D. Demaine, Playing games with algorithms: algorithmic combinatorial game theory, in: *Mathematical Foundations of Computer Science* (Mariánské Lázně), Lecture Notes in Comput. Sci. 2136, Springer, Berlin (2001), pp. 18–32.

[DemHea09] E. D. Demaine and R. A. Hearn, Playing games with algorithms: algorithmic combinatorial game theory, in: *Games of No Chance 3*, *Proc. BIRS Workshop on Combinatorial Games, July, 2005, Banff, Alberta, Canada, MSRI Publ.* (M. H. Albert and R. J. Nowakowski, eds.), Cambridge University Press, Cambridge (2009), pp. 3–56.

[DucGra–] E. Duchêne and S. Gravier, Geometrical Extension of Wythoff's Game, To appear in *Discrete Math.*.

[DucRig–] E. Duchêne and M. Rigo, Invariant games, Preprint.

[Fek23] M. Fekete, Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten, *Mathematische Zeitschrift 17 (1923)*, pp. 228–249.

[Fra69] A. S. Fraenkel, The bracket function and complementary sets of integers, *Canad. J. Math. 21 (1969)*, pp. 6–27.

[Fra82] A. S. Fraenkel, How to beat your Wythoff games' opponent on three fronts, *Amer. Math. Monthly 89 (1982)*, pp. 353–361.

[Fra98] A. S. Fraenkel, Heap games, numeration systems and sequences, *Ann. Comb. 2 (1998)*, pp. 197–210.

[Fra00] A. S. Fraenkel, Recent results and questions in combinatorial games complexities, *Theoretical Computer Science 249 #2 (2000)*, pp. 265–288.

[Fra04-1] A. SḞraenkel, Complexity, appeal and challenges of combinatorial games, *Theoretical Computer Science 313 (2004)*, pp. 393–415.

[Fra04-2] A. S. Fraenkel [2004], New games related to old and new sequences, *Integers, Electr. J. of Combinat. Number Theory 4 (2004)*, #G06, Comb. Games Sect.

[FraKri04] A. S. Fraenkel and D. Krieger, The structure of complementary sets of integers: a 3-shift theorem, *Internat. J. Pure and Appl. Math. 10 (2004)*, pp. 1–49.

[HegLar06] P. Hegarty and U. Larsson, Permutation of the Natural Numbers with Prescribed Difference Multisets, *Integers, Electr. J. Combinat. Number Theory 6 (2006)*, #A03.

[Kim07] C. Kimberling, Complementary Equations, *Journal of Integer Seq. 10 (2007)* Article 07.1.4.

[Kim08] C. Kimberling, Complementary equations and Wythoff sequences, *Integer Seq. 11 (2008)* Article 08.3.3.

[Lar07] U. Larsson, 2-Pile Nim with a Restricted Number of Move-Size Imitations, with an appendix by P. Hegarty, *arXiv:0710.3632*.

[Obr03] K. O'Bryant, Fraenkel's partition and Brown's decomposition, *Integers, Electr. J. Combinat. Number Theory 3 (2003)*, #A11.

[Sin81] D. Singmaster [1981], Almost all games are first person games, *Eureka 41 (1981)* pp. 33–37.

[Sin82] D. Singmaster, Almost all partizan games are first person and almost all impartial games are maximal, *J. Combin. Inform. System Sci. 7 (1982)*, pp. 270–274.

[SmSt02] F. Smith and P. Stânicâ, Comply/constrain games or games with a Muller twist, *Integers, Electr. J. of Combinat. Number Theory 2 (2002)*, #G3.

[Sun05] X. Sun, Wythoff's sequence and N-Heap Wythoff's conjectures, *Discrete Mathematics 300 1-3 (2005)*, pp. 180–195.

[SunZei04] X. Sun and D. Zeilberger,On Fraenkel's N-Heap Wythoff's Conjectures, *Annals of Combinatorics 8 (2004)*, pp. 225–238.

[Wyt1907] W. Wythoff, A Modification of the Game of Nim, *Nieuw Arch. Wisk. 7 (1907)* pp. 199–202.