

# Harnessing the unwieldy MEX function

AVIEZRI S. FRAENKEL AND UDI PELED

A pair of integer sequences that split  $\mathbb{Z}_{>0}$  is often — especially in the context of combinatorial game theory — defined recursively by

$$a_n = \text{mex}\{a_i, b_i : 0 \leq i < n\}, \quad b_n = a_n + c_n \quad (n \geq 0),$$

where **mex** (**M**inimum **E**Xcludant) of a subset  $S$  of nonnegative integers is the smallest nonnegative integer not in  $S$ , and  $c : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ . Given  $x, y \in \mathbb{Z}_{>0}$ , a typical problem is to decide whether  $x = a_n, y = b_n$ . For general functions  $c$ , 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.

## 1. Introduction

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 two 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:

$$\begin{cases} a_n = \text{mex}\{a_i, b_i : 0 \leq i < n\}, \\ b_n = a_n + c_n, \quad n \geq 0, \end{cases} \quad (1)$$

where **mex** (**M**inimum **E**Xcludant) of a subset  $S$  of nonnegative integers is the smallest nonnegative integer not in  $S$ . The mex definition implies that sequences  $\{a_n\}, \{b_n\}$  ( $n \geq 1$ ) of the form (1), for  $c : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ , are *complementary sequences* [Fraenkel 1982]. Later on we define precisely the 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

*MSC2010:* 11B39, 11B75, 68Q25, 91A46.

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

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  $\{(a_n, b_n)\}$ ,  $n \geq 0$ , where  $a_n, b_n$  are given by (1) with  $c_n = n$  [Fraenkel 1982]. Given a game position  $(x, y)$ , to decide whether  $(x, y) = (a_n, b_n)$  requires the computation of  $a_0, b_0, a_1, \dots, b_{k-1}, a_k$ , where  $k$  is the largest index such that  $a_k \leq x$ . This is an exponential computation, since the input size 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  $\varphi = (1 + \sqrt{5})/2$  is the golden section [Fraenkel 1982]. 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 positive integer parameter ( $p = 1$  is Wythoff's game). The  $P$ -positions of this  $p$ -Wythoff game are given by (1) for  $c_n = pn$  [Fraenkel 1982]. 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  $a_n, b_n$  [Fraenkel 1982]. For a recent and comprehensive survey on the complexity of combinatorial games see [Demaine and Hearn 2009]. Previous surveys are in [Demaine 2001; Fraenkel 2000; 2004a]. There is an extensive literature on Wythoff's game; here we just cite two of them: [Wythoff 1906; Fraenkel 1982].

For other functions  $c$ , 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 [Fraenkel 1998] that for  $s > 1$ , there are no  $\alpha, \beta, \gamma, \delta$  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  $c$  is not an integer, no efficient strategy is known in general. A case in point is a game considered in [Duchêne and Gravier 2009], where  $c_n = 2\lfloor n/4 \rfloor$ . For the case where  $c$  is a special algebraic number, it was recently shown in [Duchêne and Rigo 2010] 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  $c$ , 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 [Fraenkel 2004b; Hegarty and Larsson 2006; Duchêne and Gravier 2009]. 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 online 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 [Kimberling 2007] for a survey and also [Kimberling 2008].

In the next section we formalize some of the notions and tools which will be used in the sequel. In particular, we define the notion of approximate linearity. In Section 3, which is the main technical section, we explore complementary sequences by proving Theorem 3.2, which gives a basic yet important inequality about complementary sequences. This is followed by Theorem 3.4, which establishes an important combinatorial equivalence concerning increasing sequences. Both of these results are then used to prove the fundamental Bounded Additivity Theorem 3.5. It enables one to wed together complementarity and approximate linearity in Section 4, with the aid of Fekete's lemma [1923], leading to our main result, the Approximate Linearity Theorem 4.3. This result then leads, in Section 5, to the efficient algorithm, together with an example and its complexity analysis. In Section 6 we wrap-up with a conclusion.

## 2. Formalizations

**Definition 2.1** (*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 in [Singmaster 1981]; see also [Singmaster 1982].

**Definition 2.2** (Minimal EXcludant). Let  $S \subsetneq Z_{\geq 0}$ . Then

$$\text{mex}(S) := \min(Z_{\geq 0} \setminus S).$$

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  $c_n \geq 1$  ( $n \geq 1$ ) [Fraenkel 1982]. 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 ( $p, q \geq 1$  integer constants) was defined by Hegarty and Larsson [2006]. It is  $p$ -Wythoff mentioned earlier, with an additional "Muller twist" (see [Smith and Stănică 2002]), namely, if the current position is  $(k, \ell)$ , then, before the next move is made, the previous player is allowed to choose up to  $(q-1)$  distinct positive integers  $t_1, \dots, t_{q-1} \leq \min\{k, \ell\}$  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 [Boshernitzan and Fraenkel 1981], 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 of positive integers  $p, q$ . No numeration system seems to help, so there appears to be no efficient method for computing the sequences. This game was one of the motivations for the current work.

Duchêne and Gravier [2009] analyzed the following 2-pile extension of Wythoff's game: 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  $F_n = \text{mex}\{P_i, Q_i : 0 \leq i < n\}$ ,  $G_n = F_n + 2 \lfloor n/4 \rfloor$  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. Both of the last mentioned games can be solved polynomially by our method.

**Definition 2.3** (approximate linearity). We call the sequence  $A$  *approximately linear* if there exist constants  $\alpha, u_1, u_2 \in \mathbb{R}$  such that

$$u_1 \leq a_n - n\alpha \leq u_2 \quad \text{for all } n \geq 0.$$

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

**Remark.** Notice that  $\alpha$  is unique since  $\lim_{n \rightarrow \infty} a_n/n = \alpha$ , 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  $\alpha$ , and similarly for  $b_n$ . Our first result of this work was proving that indeed this approximation holds, by finding bounds depending on  $p, q$ . Hegarty [2009] independently proved the same approximation for the special case  $p = 1$ . We then generalized our result to the case where  $c$  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 function  $c$ .

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 [2004] and Sun and Zeilberger [2004; Sun 2005] investigated the  $P$ -positions of *Fraenkel's  $N$ -Heap Wythoff's game* [Fraenkel 2004a]. 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.

### 3. Complementary sequences

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

**Definition 3.1** (complementary sequences). We say that two integer sequences  $A := \{a_n\}_{n \geq 0}, B := \{b_n\}_{n \geq 0}$  are *complementary sequences* if they are strictly increasing sequences which satisfy  $A \cup B = \mathbb{Z}_{\geq 0}, A \cap B = \{a_0 = b_0 = 0\}$ . We also define  $c := B - A$ .

We begin with a property of complementary sequences, which will be used both in the proof of Theorem 3.5, 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 3.2** (complementary sequences inequality). *Suppose that  $A, B$  are complementary sequences. Then*

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

*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.  $\square$

**Remark.** Denote  $I = [b_r, b_s]$ , by the theorem we get that the first element of  $A \cap I$  is  $a_{b_r-r+1}$ , and the last element of  $A \cap I$  is  $a_{b_s-s}$ , and a similar result by switching the roles of  $A, B$ .

Now for simple bounds on the growth of the smaller sequence of complementary sequences.

**Corollary 3.3.** *Assume that  $A, B$  are complementary sequences. 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 3.2 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, the indices of the last inequality must satisfy  $a_n - n < n$  as claimed.  $\square$

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 3.5.

**Theorem 3.4** (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  $x, u$ .*

$$\min_{L \geq 0} |[L+1, L+u] \cap B| \geq x \iff \text{for all } n \geq 0, b_{n+x} \leq b_n + u.$$

*And similarly also the following equivalence holds*

$$\max_{L \geq 0} |[L+1, L+u] \cap B| \leq x \iff \text{for all } n \geq 0, b_n + u \leq b_{n+x}.$$

*Proof.* The implication

$$\min_{L \geq 0} |[L+1, L+u] \cap B| \geq x \implies \text{for all } n \geq 0, b_{n+x} \leq b_n + u$$

is obvious, since we can take  $L = b_n$ .

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  $b_{n+1}, b_{n+2}, \dots, b_{n+x} \in [L + 1, L + u]$  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  $L < b_n$ .  $\square$

**Remark.** Equivalence also holds when either  $x$  or  $u$  are negative integers.

The following result enables us to prove Theorem 4.3. The proof is by induction, and in order to better understand it, we recommend following the first steps of  $m = 1, 2, 3, \dots$  with specific sequences  $\{a_n\}, \{b_n\}$  (e.g.,  $b_n = a_n + \lfloor \sqrt{2n} \rfloor$ ).

**Theorem 3.5** (bounded additivity). *Let  $A, B$  be complementary sequences. 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  $0 \leq t \leq m$ ,

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

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

By the induction assumption we know

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

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

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

Now, by Corollary 3.3 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

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

From Theorem 3.2 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

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

Since this holds for all  $n \geq 0$ , Theorem 3.4 implies that for every fixed  $t \leq m$ ,

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

where the closed interval  $I_L$  is given by  $I_L = [L+1, L+a_{t+1}+s-r]$ . Now, since  $A, B$  are complementary, 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 3.4 we get

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

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

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

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

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

By Corollary 3.3 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

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

From Theorem 3.2 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

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

Since this holds for all  $n \geq 0$ , by Theorem 3.4 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  $A, B$  are complementary, 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 3.4 we get

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

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



**Remark.** The theorem's conclusion also holds for the sequence  $B$ , that is, 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

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

for all  $n, m \geq 0$ . 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 3.6.** *Assume that  $\{a_n\}_{n \geq 0}$  is a real sequence, which for some constants  $r, s \in \mathbb{Z}_{\geq 0}$  and  $u, v \in \mathbb{R}$ , satisfies*

$$a_{n+m} \leq a_n + a_{m+r} + u \leq a_{n+m+s} + v \quad \text{for all } n, m \geq 0.$$

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

$$a_{n+m} + D \leq a_n + a_m \leq a_{n+m} + E \quad \text{for all } n, m \geq 0.$$

*Proof.* By the left inequality of the assumption we know that  $a_{t+r} \leq a_t + a_{2r} + u$  for all  $t \geq 0$ , so by taking  $t = m$  together with the assumption we get

$$a_{n+m} \leq a_n + a_{m+r} + u \leq a_n + a_m + a_{2r} + 2u.$$

We can take  $D = -a_{2r} - 2u$  to fit our goal.

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

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

#### 4. Approximate linearity

We use a well known lemma from [Fekete 1923].

**Lemma 4.1** (Fekete's lemma). *Let  $\{s_n\}_{n \geq 0}$  be a superadditive sequence of real numbers, that is,  $s_n + s_m \leq s_{n+m}$  for all  $n, m \geq 0$ . Then  $\lim_{n \rightarrow \infty} s_n/n$  exists, and is equal to  $\sup_{n \geq 1} s_n/n \in (-\infty, \infty]$ .*

From this lemma we derive the following corollary, which can be applied directly to the results of Theorem 3.5.

**Corollary 4.2** (Fekete's corollary). *Assume that  $A$  is a sequence of real numbers which satisfies, for some constants  $u, v \in \mathbb{R}$ ,*

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

*Then  $A$  is approximately linear, with approximation bounds  $u, v$ , and linearity rate  $\alpha = \lim a_n/n$ .*

*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

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

Hence for  $n \geq 1$  we have  $a_n - v \leq n\alpha$ , 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

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

Hence for  $n \geq 1$   $u - a_n \leq n\alpha'$ , 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  $\alpha \in \mathbb{R}$ . Altogether we get  $u \leq a_n - n\alpha \leq v$ , as claimed.  $\square$

Using Theorem 3.5 together with Fekete's lemma and their corollaries, we get Theorem 4.3, which is the main theorem of this paper.

**Theorem 4.3** (approximate linearity). *Let  $A, B$  be complementary sequences. Then both  $A$  and  $B$  are 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  $c$  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 3.5, so we first look for integer constants  $r, s \geq 0$ . We take an integer  $r \geq 0$ , which satisfies  $r \geq (K_2 - 2K_1)/\gamma$ . Thus,

$$c_{n+m-r} \leq (n+m-r)\gamma + K_2 \leq n\gamma + K_1 + m\gamma + K_1 \leq c_n + c_m$$

(the middle inequality is by our choice of  $r$ , and rightmost/leftmost inequalities are the approximate linearity of  $c$ ).

We also take some integer  $s \geq 0$  which satisfies  $s \geq (2K_2 - K_1)/\gamma$ . Then

$$c_n + c_m \leq n\gamma + K_2 + m\gamma + K_2 \leq (n+m+s)\gamma + K_1 \leq c_{n+m+s}.$$

But recall that the theorem also requires  $s \geq r - 1$ , so we choose  $s$  that also satisfies this.

In conclusion,  $c_{n+m-r} \leq c_n + c_m \leq c_{n+m+s}$ , so by Theorem 3.5 and Lemma 3.6 we get  $u_1 \leq a_n + a_n - a_{n+m} \leq u_2$  for some constants  $u_1, u_2$ , and similarly for  $B$ . Hence Fekete's Corollary implies that  $A, B$  are approximately linear.  $\square$

**Remark.** If we know the approximation bounds  $K_1, K_2$  and the linearity rate  $\gamma = \lim c_n/n$ , then by Theorem 3.5's conclusion and the proof of Lemma 3.6, 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  $\alpha, \beta$  are known. Since  $A, B$  are complementary sequences, and since  $\lim a_n/n = \alpha, \lim b_n/n = \beta$ , density considerations imply  $\alpha^{-1} + \beta^{-1} = 1$  [Fraenkel 1982]. Also notice that  $c_n = b_n - a_n$  implies  $\beta - \alpha = \gamma$ , so we find  $\alpha, \beta$  by solving the quadratic equation  $\alpha^{-1} + (\alpha + \gamma)^{-1} = 1$ . For any  $\theta \in \mathbb{R}_{>0}$ , setting  $c_n = k \lfloor \theta n \rfloor$  we have  $c = k\theta$ , so we obtain  $a = \frac{1}{2}(2 - k\theta + \sqrt{k^2\theta^2 + 4})$ ,  $b = \frac{1}{2}(2 + k\theta + \sqrt{k^2\theta^2 + 4})$ .

If  $0 < \theta < 1$  then for  $0 \leq i < 1/\theta$  we get  $c_i = k \lfloor \theta i \rfloor = 0$ . And for  $i_0 = \lceil 1/\theta \rceil$  we get  $c_{i_0} = k \lfloor \theta i_0 \rfloor \geq 1$  and  $b_i = a_i = i$ ; thus  $a_{i_0} = i_0, b_{i_0} > i_0$ . By shifting the indices, we get the complementary sequences  $A', B'$  defined by  $a'_n = a_{n+i_0-1} - i_0 + 1$ ,  $b'_n = b_{n+i_0-1} - i_0 + 1$  for  $n \geq 0$ . 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 4.3,  $A', B'$  are approximately linear. Definition 2.3 then implies that also the original  $A, B$  are approximately linear.

## 5. Efficient algorithm for computing sequences

**5.1. Preparation.** We deal with complementary sequences  $A, B$ , 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 4.3, 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 by the approximate linearity we must have  $(x - u_2)/\alpha \leq n \leq (x - u_1)/\alpha$ . 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  $a_m, a_{m+1}, \dots, a_n$  for given  $m, n$ , by operating as follows. If for some  $r, s$  we can compute  $b_r, b_{r+1}, \dots, b_s$ , where  $b_r < a_m$  and  $a_n < b_s$ . Then since  $A, B$  are complementary sequences, we can compute  $a_m, \dots, a_n$  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 necessarily be  $a_m$ , but rather  $a_{b_r-r+1}$  by Theorem 3.2. 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  $a_m, \dots, a_n$  inside  $a_{b_r-r+1}, \dots, a_{b_s-s}$ .

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,  $b_r, \dots, b_s$  should be efficiently computable. First we deal with finding such  $r$ . By Theorem 3.2 we can take any  $r \leq a_m - m$ , and it will satisfy  $b_r < a_m$ . But we have not yet computed  $a_m$ , so how can we find  $r \leq a_m - m$ ?

By the approximate linearity of  $\{a_i\}$  we know that  $m(\alpha - 1) + u_1 \leq a_m - m$ , so we can take  $r = \lfloor m(\alpha - 1) + u_1 \rfloor$  to satisfy  $r \leq m(\alpha - 1) + u_1 \leq a_m - m$ , hence  $b_r < a_m$ . By similar reasoning, for  $s = \lceil n(\alpha - 1) + u_2 \rceil + 1$  we have  $a_n - n + 1 \leq n(\alpha - 1) + u_2 + 1 \leq s$ , hence  $a_n < b_s$  (Theorem 3.2).

So we have easy formulas for  $r, s$ , which satisfy  $b_r < a_m, a_n < b_s$ . But how can we easily compute  $b_r, \dots, b_s$ ?

Notice that it's enough to know  $a_r, a_{r+1}, \dots, a_s$ , and then we can compute directly  $b_i = a_i + c_i$ . In order to compute  $a_r, \dots, a_s$ , 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  $a_m, a_{m+1}, \dots, a_n$ .

- 1: If  $m \leq 0$  then
- 2:     For  $j = 0, \dots, n$  do

```

3:    $a_j \leftarrow \text{mex}\{a_i, b_i : 0 \leq i < j\}$ .
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:    $a_r, \dots, a_s \leftarrow \text{ComputeSequenceA}(r, s)$ .
9:   For  $j = r, \dots, s$  do
10:     $b_j \leftarrow a_j + c_j$ .
11:   End for
12:   Calculate  $a_{b_r-r+1}, \dots, a_{b_s-s}$  as the complemented elements of  $b_r, \dots, b_s$ 
      in the interval  $[b_r, b_s]$ .
13: End if
14: Return  $a_m, \dots, a_n$ .

```

**5.4. Example.** We demonstrate the following case:  $A, B$  are complementary sequences, and  $b_n = a_n + \lfloor 4n/3 \rfloor$ ; i.e.,  $c_n = \lfloor 4n/3 \rfloor$ . Then  $K_1 \leq c_n - n\gamma \leq K_2$  holds for  $K_1 = -1, K_2 = 0, \gamma = \frac{4}{3}$ . In the proof of Theorem 4.3 we take  $r = 2, s = 1$  to satisfy  $r \geq (K_2 - 2K_1)/\gamma = \frac{3}{2}$  and  $s \geq (2K_2 - K_1)/\gamma = \frac{3}{4}$  (also  $s \geq r - 1$ ). Then we have  $c_{n+m-2} \leq c_n + c_m \leq c_{n+m+1}$ . 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 3.5,  $a_{n+m-2} - 1 \leq a_n + a_m + 2 - 1 \leq a_{n+m+1}$ , equivalently  $a_{n+m} \leq a_n + a_{m+2} + 0 \leq a_{n+m+3} + 1$ . Then in the proof of Lemma 3.6 we get that  $a_{n+m} + h_1 \leq a_n + a_m \leq a_{n+m} + h_2$  holds for  $h_1 = -a_4, h_2 = a_0 + a_5 + 1$ . 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  $h_1 = -6, h_2 = 0 + 7 + 1 = 8$ , and Fekete's corollary yields  $-6 \leq a_n - n\alpha \leq 8$ , where  $\alpha = 1 - \gamma/2 + \sqrt{1 + (\gamma/2)^2} = \frac{1}{3}(1 + \sqrt{13})$ .

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).

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

Computing  $b_i = a_i + c_i$  (line 4) gives

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

The indices of  $a_{518}, \dots, a_{562}$  are derived from Theorem 3.2; that is,  $a_{b_0-0+1}$  and  $a_{b_{33}-33}$  are the first and last elements of  $A$  inside the interval  $[b_0, b_{33}]$ .

By complementing the interval  $[b_0, b_{33}]$  (line 12) we get  $a_1, a_2, \dots, a_{61}$ . From these we extract the desired elements ( $m = 10, n = 44$ ):

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

Computing  $b_i = a_i + c_i$  (line 9), we get

$\vdots$

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

Again, the indices of  $a_{518}, \dots, a_{562}$  are derived from Theorem 3.2, i.e.,  $a_{b_{277}-277+1}$  and  $a_{b_{301}-301}$  are the first and last elements of  $A$  inside the interval  $[b_{277}, b_{301}]$ .

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

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

Computing  $b_i = a_i + c_i$  (line 9), we get

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

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

$a_{1000}$
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  $\alpha k$  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  $\alpha = (1 + \sqrt{13})/3$ . Hence computing  $\alpha k$  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

$$m_{i+1} = \lfloor m_i(\alpha - 1) + u_1 \rfloor, \quad n_{i+1} = \lceil n_i(\alpha - 1) + u_2 + 1 \rceil.$$

From this it's easy to see that

$$m_{i+1} \leq m_i(\alpha - 1) + u_1 \quad \text{and} \quad n_{i+1} - m_{i+1} \leq (n_i - m_i)(\alpha - 1) + u_2 - u_1 + 3,$$

which implies

$$\begin{aligned} m_i &\leq (\alpha - 1)^i \left( n - \frac{u_1}{2 - \alpha} \right) + \frac{u_1}{2 - \alpha}, \\ n_i - m_i &\leq -(\alpha - 1)^i \frac{u_2 - u_1 + 3}{2 - \alpha} + \frac{u_2 - u_1 + 3}{2 - \alpha}. \end{aligned} \quad (2)$$

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

Since  $0 < \alpha - 1 < 1$ , and  $u_1 \leq a_0 - 0 = 0$ , from (2) we see that the recursion stopping condition of the algorithm  $m \leq 0$  (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  $m_i \leq 0$ , 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  $[b_{m_i}, b_{n_i}]$  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 than  $O(n)$ . For the example in Section 5.4, it is  $O(\log^3 n)$ . The arguments in Sections 5.1, 5.2 and 5.5 jointly also establish the validity of the algorithm.

## 6. Conclusion

We have shown that the unwieldy mex function can be harnessed to behave polynomially for a large set of interesting games, such as those given in [Hegarty and Larsson 2006; Duchêne and Gravier 2009], for which no polynomial winning strategy was known.

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 3.2 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  $\mathbb{Z}_{\geq 1}$ .

## Acknowledgments

We wish to thank Eric Duchêne for being one of the triggers of this research by a question in his 2006 PhD 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

- [Boshernitzan and Fraenkel 1981] M. Boshernitzan and A. S. Fraenkel, “Nonhomogeneous spectra of numbers”, *Discrete Math.* **34**:3 (1981), 325–327.
- [Demaine 2001] E. D. Demaine, “Playing games with algorithms: algorithmic combinatorial game theory”, pp. 18–32 in *Mathematical foundations of computer science*, edited by A. Sgall, Jiříand Pultr and P. Kolman, Lecture Notes in Comput. Sci. **2136**, Springer, Berlin, 2001.
- [Demaine and Hearn 2009] E. Demaine and R. Hearn, “Playing games with algorithms: algorithmic combinatorial game theory”, pp. 3–56 in *Games of no chance 3*, Math. Sci. Res. Inst. Publ. **56**, Cambridge, 2009.
- [Duchêne and Gravier 2009] E. Duchêne and S. Gravier, “Geometrical extensions of Wythoff’s game”, *Discrete Math.* **309**:11 (2009), 3595–3608.
- [Duchêne and Rigo 2010] E. Duchêne and M. Rigo, “Invariant games”, *Theoret. Comput. Sci.* **411**:34–36 (2010), 3169–3180.
- [Fekete 1923] M. Fekete, “Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten”, *Math. Zeitschrift* **17**:1 (1923), 228–249.
- [Fraenkel 1982] A. S. Fraenkel, “How to beat your Wythoff games’ opponent on three fronts”, *Amer. Math. Monthly* **89**:6 (1982), 353–361.
- [Fraenkel 1998] A. S. Fraenkel, “Heap games, numeration systems and sequences”, *Ann. Comb.* **2**:3 (1998), 197–210.
- [Fraenkel 2000] A. S. Fraenkel, “Recent results and questions in combinatorial game complexities”, *Theoret. Comput. Sci.* **249**:2 (2000), 265–288.
- [Fraenkel 2004a] A. S. Fraenkel, “Complexity, appeal and challenges of combinatorial games”, *Theoret. Comput. Sci.* **313**:3 (2004), 393–415.
- [Fraenkel 2004b] A. S. Fraenkel, “New games related to old and new sequences”, *Integers* **4** (2004), G6, 18.
- [Fraenkel and Krieger 2004] A. S. Fraenkel and D. Krieger, “The structure of complementary sets of integers: a 3-shift theorem”, *Int. J. Pure Appl. Math.* **10**:1 (2004), 1–49.
- [Hegarty and Larsson 2006] P. Hegarty and U. Larsson, “Permutations of the natural numbers with prescribed difference multisets”, *Integers* **6** (2006), A3, 25.
- [Kimberling 2007] C. Kimberling, “Complementary equations”, *J. Integer Seq.* **10**:1 (2007), Article 07.1.4, 14.
- [Kimberling 2008] C. Kimberling, “Complementary equations and Wythoff sequences”, *J. Integer Seq.* **11**:3 (2008), Article 08.3.3, 8.
- [Larsson 2009] U. Larsson, “2-pile Nim with a restricted number of move-size imitations”, *Integers* **9** (2009), G04, 671–690.
- [Singmaster 1981] D. Singmaster, “Almost all games are first person games”, *Eureka* **41** (1981), 33–37.
- [Singmaster 1982] D. Singmaster, “Almost all partizan games are first person and almost all impartial games are maximal”, *J. Combin. Inform. System Sci.* **7**:4 (1982), 270–274.

- [Smith and Stănică 2002] F. Smith and P. Stănică, “Comply/constrain games or games with a Muller twist”, *Integers* **2** (2002), G3, 10.
- [Sun 2005] X. Sun, “Wythoff’s sequence and  $N$ -heap Wythoff’s conjectures”, *Discrete Math.* **300**:1-3 (2005), 180–195.
- [Sun and Zeilberger 2004] X. Sun and D. Zeilberger, “On Fraenkel’s  $N$ -heap Wythoff’s conjectures”, *Ann. Comb.* **8**:2 (2004), 225–238.
- [Wythoff 1906] W. A. Wythoff, “A modification of the game of Nim”, *Nieuw Arch. Wisk.* **7** (1906), 199–202.

aviezri.fraenkel@weizmann.ac.il    *Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 76100 Rehovot, Israel*

udi@udipeled.co.il                    *Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel*