Journal of Integer Sequences, Vol. 29 (2026),
Article 26.1.6

Variants of Wythoff Game With Terminal Positions or Blocking Maneuvers

Antoine Renard and Michel Rigo
Department of Mathematics
University of Liège
Allée de la Découverte 12 (B37)
B-4000 Liège
Belgium
Antoine.Renard@uliege.be
M.Rigo@uliege.be

Abstract

We show how the software `Walnut` can be used to obtain concise proofs of results concerning variants of the famous Wythoff game, incorporating blocking maneuvers or additional terminal positions, as discussed by Larsson (2011) and Komak et al. (2025), respectively. Our approach provides automatic proofs that both confirm and extend their results, and the same techniques apply equally to newly introduced variants.

Then, using classical techniques, we obtain new recursive and morphic characterizations of Wythoff-type games in which the terminal positions (x, y) satisfy $x + y \leq \ell$.

The use of `Walnut` in combinatorial game theory is relatively recent, and only a few examples have been investigated so far. The Wythoff game, being directly connected to the Fibonacci numeration system, proves especially well-suited to this kind of approach. It permits us to solve instances for a fixed value of a parameter.

1 Introduction

In combinatorial game theory, proofs of many results are obtained by using a suitable numeration system [7, 10, 11, 25]. For instance, the Zeckendorf numeration based on the Fibonacci sequence is used to characterize the $\mathcal{P}$-positions (i.e., losing positions) of the Wythoff game

[9]. Fraenkel showed that a pair (a, b) of integers such that $a \leq b$ is a $\mathcal{P}$-position of Wythoff game if and only if $\text{rep}_F(a)$ ends with an even number of zeroes and $\text{rep}_F(b)$ is the left-shift of $\text{rep}_F(a)$, i.e., $\text{rep}_F(b) = \text{rep}_F(a)0$.

When the numeration system has good properties (i.e., it is an *addable* system, in the sense of Shallit [26]), the formalism of first-order logic can be used to express certain properties of combinatorial games. We have recently employed this approach and used `Walnut` to reprove classical results in combinatorial game theory, as well as to obtain new ones [23]. `Walnut` is a free, open-source prover for first-order statements dealing with automatic sequences [24, 26].

We assume that the reader is sufficiently familiar with the use of `Walnut`. For background on combinatorial game theory, we refer to [25] and [23].

The Wythoff game is a *take-away* game: two players alternately remove tokens from two piles. One can remove a positive number of tokens from one pile or remove the same positive number of tokens from both piles. The Wythoff game is *impartial*: the set of options (i.e., positions reachable in one legal move) of a given position does not depend on which player is to move next. The Wythoff game can be seen as a queen on an infinite chessboard $\mathbb{N}^2$: in a position $(x, y) \neq (0, 0)$, the queen can move horizontally to $(x - n, y)$ with $0 < n \leq x$, vertically to $(x, y-n)$ with $0 < n \leq y$, or diagonally to $(x-n, y-n)$ with $0 < n \leq \min\{x, y\}$. According to the *normal convention*, a player who is unable to move loses the game; passing is not allowed. In other words, the player who takes the last token wins or, in an equivalent way, when the queen reaches $(0, 0)$.

In this article, we are mainly interested in two kinds of variants of the Wythoff game.

Komak et al. [19] introduced a variant of the Wythoff game: the set $\{(x, y) \mid x + y \leq 2\}$ is declared to be the set of *terminal* positions. If a player moves the queen into this terminal set with the usual Wythoff moves, that player wins the game. Let us call this game K^2. We adopt this notation because we will later introduce a generalization of the game obtained by varying the set of terminal positions. Let ϕ be the golden ratio $(1 + \sqrt{5})/2$. We first define the sequence $(g(n))_{n\geq 0}$ as follows.

Definition 1. We let $g(0) = 1$, $g(1) = 0$ and

$$g(n) = \begin{cases} 1 - g(m), & \text{if } \lfloor n\phi \rfloor = \lfloor m(\phi + 1) \rfloor + 1 \text{ for some } m \geq 0; \\ 1, & \text{otherwise.} \end{cases}$$

The first few values of the sequence g are given in Table 1.

The main result of Komak et al. provides an algebraic description of the set of $\mathcal{P}$-positions of K^2:

Theorem 2. *The set of $\mathcal{P}$-positions of the variant of the Wythoff game where the set of terminal positions is $\{(x, y) \mid x + y \leq 2\}$ is exactly*

$$\left\{(\lfloor n\phi \rfloor + g(n) - 1, \lfloor n\phi^2 \rfloor + g(n)) \mid n \geq 0\right\} \cup \left\{(\lfloor n\phi^2 \rfloor + g(n), \lfloor n\phi \rfloor + g(n) - 1) \mid n \geq 0\right\}, \tag{1}$$

where $g : \mathbb{N} \to \mathbb{N}$ is the function given in Definition 1.

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

Table 1: The first few values of the sequence g and the Hofstadter sequence h.

The authors provide a classical proof of their result by showing that the set of $\mathcal{P}$-positions described by (1) is both stable and absorbing; see Definition 10. The proof is quite long and requires a detailed case analysis. Without diminishing their achievement, once the set (1) has been conjectured, if the set can be handled by `Walnut`, it becomes straightforward to produce an automated proof using just a few lines. Our approach, therefore, complements the work carried out by Komak et al. [19].

Larsson [20] studied other variants of the Wythoff game, in which some blocking maneuvers are added. Let $k \geq 1$. In the game denoted by W^k, for each move, before the next player (i.e., the player who is about to play) moves, the previous player (i.e., the one who has just played) may declare at most $k-1$ of the options as forbidden. When the next player has moved, any blocking maneuver is forgotten and has no further impact on the game. This terminology of *previous* and *next* player explains why we speak of $\mathcal{N}$- and $\mathcal{P}$-positions, respectively. For $k = 1$, this is the classical game of Wythoff. When $k = 2, 3$, we reconsider Larsson's result and obtain an automated proof, replacing a 2-page-long analysis by some easy-to-describe first-order formulas.

As Shallit noted on several occasions, the two approaches are complementary. A classical, purely combinatorial proof usually provides structural insights, whereas the use of automated provers helps avoid lengthy case analyses and enables the exploration of directions that are difficult to access through traditional techniques. In particular, in this article, we easily obtain a variety of results that would otherwise require significantly more effort using "classical" methods. Moreover, these results lead us to formulate general theorems.

This article is organized as follows. Section 2—which was the starting point of this paper—aims to give an automatic proof of the algebraic characterization (1) of the $\mathcal{P}$-positions of K^2. We begin with preliminary results about Fibonacci-automatic sequences. We define the notion of φ-morphism and describe a heuristic that we extensively use throughout the paper to construct such φ-morphisms. Given a sufficiently long prefix of an infinite word, we obtain morphisms that can be used in our automatic proofs (and we can therefore prove the correctness of the procedure). Basic results from combinatorial game theory are given in Section 2.3.

In Section 3, we provide results about new games: a parametrized version of a variant of the Wythoff game denoted by K^ℓ, where the set of terminal states is

$$\{(x, y) \in \mathbb{N}^2 \mid x + y \leq \ell\}$$

for some $\ell \in \mathbb{N}$. We study the games K^1, K^3, and K^4 and obtain algebraic characterizations

of the $\mathcal{P}$-positions similar to those described by Komak et al. [19]. In particular, as for the classical Wythoff game K^0, the game K^1 has a nice set of $\mathcal{P}$-positions: a pair (a, b) of integers such that $a \leq b$ is a $\mathcal{P}$-position of K^1 if and only if its Fibonacci representation $\mathrm{rep}_F(a)$ ends with zero and $\mathrm{rep}_F(b)$ is of the form $\mathrm{rep}_F(a)1$; see Theorem 12.

In Section 4, we go further in the analysis of the games K^ℓ for an arbitrary $\ell \geq 1$. We provide a recursive characterization of the set of $\mathcal{P}$-positions. The non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\geq 0}$ of K^ℓ are given, in Theorem 17, by

$$(a_0, b_0) = (\ell + 1, 2\ell + 2) \quad \text{and} \quad \forall n \geq 1, \begin{cases} a_n = \mathrm{MeX}(\{a_i, b_i \mid i < n\} \cup \{0, 1, \ldots, \ell\}) \\ b_n = a_n + n + \ell + 1. \end{cases}$$

This is a very natural extension of Wythoff's characterization [28]. We automatically prove this result for $\ell = 1, 2$, then extend it for all values of the parameter using classical arguments. With Corollary 19, we generalize Theorem 2 and obtain that

$$a_n = \lfloor (n + \ell)\phi \rfloor + \lambda_\ell(n) \quad \text{and} \quad b_n = \lfloor (n + \ell)\phi^2 \rfloor + \lambda_\ell(n) + 1$$

for some bounded integer-valued function λ_ℓ. Our result can be related to the recent work of Letouzey [22], where the discrepancy of generalized Hofstadter functions is studied (and formalized in `Coq`). Our methods are inspired by Fraenkel and Peled [12], Kimberling [17], and Larsson [21].

It is well-known that the $\mathcal{P}$-positions of the Wythoff game are coded by the Fibonacci word $\mathbf{f} = abaababa\cdots$; see [5, 7]. If we start indexing letters of $\mathbf{f}$ with 1, the indices of the letters a (resp., b) in $\mathbf{f}$ correspond to the sequence $(a_n)_{n\geq 0}$ (resp., $(b_n)_{n\geq 0}$). Using the recursive definition from the previous section, we get similar results with K^1, K^2, and K^3: their $\mathcal{P}$-positions are coded by morphic words obtained by φ-morphisms and codings.

In Section 6, we provide automatic proofs of Larsson's result about W^2 and W^3. Interestingly, the automata arising in the computations are much larger than those for K^2.

In Section 7, we are concerned with redundant moves of the games discussed so far. A move is considered *redundant* if, upon its removal from the rule-set, the set of $\mathcal{P}$-positions remains unchanged. This is a classical question in combinatorial game theory: do distinct rule-sets yield the same set of $\mathcal{P}$-positions? In particular, may certain moves be added or removed without altering the $\mathcal{P}$-positions?

Remark 3. A `Jupyter` notebook recording all the `Walnut` computations is available (it has been produced using Ollinger's `Walnut-Kernel`). The `Mathematica` code and the various `Walnut` files are also available online[1]. In particular, the package permits us to compute $\mathcal{P}$-positions and φ-morphisms.

[1]Available at `https://hdl.handle.net/2268/338482`; you may also download a standalone Wolfram Player.

2 Our starting point: The game K^2

This section is about the game K^2. The aim is to give an automatic proof of the characterization of its $\mathcal{P}$-positions. We first show that the sequence g appearing in (1) is *Fibonacci-automatic*, i.e., for all $n \geq 0$, its nth term is the output of a DFAO fed with the Fibonacci representation of n; see Definition 5. To discover this DFAO, we apply a heuristic that we describe in Section 2.2 (and then apply several times in Section 3). Finally, we give the automatic proof. To be self-contained, classical definitions from combinatorial game theory are given in Section 2.3.

2.1 A first result on the sequence g

To make use of `Walnut`, all the ingredients need to be definable in a suitable extension of $\langle \mathbb{N}, + \rangle$. Here, we show that the sequence g can be computed with a DFAO fed with Fibonacci representations.

Let us follow the presentation given by Komak et al. [19], where an alternative definition of g is provided.

Proposition 4. *We have $g(0) = g(1) = 1$, and for all $n \geq 2$,*

$$g(n) = \begin{cases} 1 - g(h(n-1)) & \text{if } h(n-2) < h(n-1), \\ 1 & \text{otherwise,} \end{cases}$$

where h is the Hofstadter G-sequence A005206*; see [15, p. 145].*

A characterization of the entries of the Hofstadter G-sequence in terms of the lower and upper Wythoff sequence is given by Dekking [4].

Definition 5. Let $F = (F_n)_{n \geq 0}$ be the Fibonacci sequence, where $F_0 = 1$, $F_1 = 2$ and $F_{n+2} = F_{n+1} + F_n$ for all $n \geq 0$. We let $\mathrm{rep}_F(n)$ denote the *greedy representation* of the integer $n > 0$, that is the unique word $d_k \cdots d_0$ over $\{0, 1\}$ such that

$$n = \sum_{i=0}^{k} d_i F_i,$$

with $d_k \neq 0$ and $d_{i+1} d_i \neq 11$ for $i < k$. The representation of 0 is the empty word, i.e. $\mathrm{rep}_F(0) = \varepsilon$.

The Beatty sequences $\lfloor n\phi \rfloor$ and $\lfloor n\phi^2 \rfloor$, also known as the lower A000201 and upper A001950 Wythoff sequences, respectively, can be defined in `Walnut` using Shallit's method [26, p. 278]:

```
reg shift {0,1} {0,1} "([0,0]|([0,1][1,1]*[1,0]))*":
def phin "?msd_fib (s=0 & n=0) | Ex $shift(n-1,x) & s=x+1":
def phi2n "?msd_fib (s=0 & n=0) | Ex,y $shift(n-1,x) & $shift(x,y) & s=y+2":
```

For instance, the command

```
eval test "?msd_fib $phin(3,4)":
```

returns `TRUE` because $\lfloor 3\phi \rfloor = 4$.

Proposition 6. *The sequence $(g(n))_{n\geq 0}$ is Fibonacci-automatic. The DFA depicted in Fig. 1 accepts $\mathrm{rep}_F(n)$ if and only if $g(n) = 1$.*

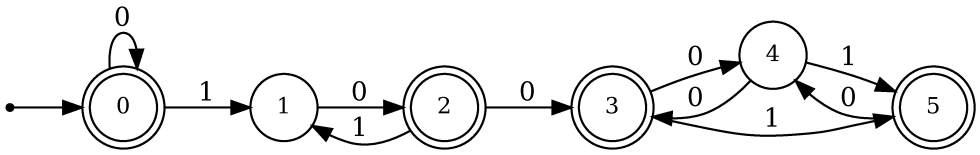

Figure 1: A DFA recognizing $\{\mathrm{rep}_F(n) : g(n) = 1\}$.

Proof. The candidate automaton depicted in Figure 1 can be encoded and stored in `Word Automata Lib`. We now have access to a candidate function `G` in `Walnut`. A discussion about how we have obtained this DFA is given in Section 2.2. Now, it is well-known [13, 14] that the Hofstadter sequence satisfies, for $n > 0$,

$$h(n) = \left\lfloor \frac{n+1}{\phi} \right\rfloor = \lfloor (n+1)\phi \rfloor - n - 1.$$

Hence, $h(n) = y$ can be easily coded as a binary predicate $h(n, y)$ by

```
def hofstadter "?msd_fib (y=1 & n=0) | Ex $phin(n+1,x) & y+n+1=x":
```

Now we have to check that the automaton satisfies the properties expressed in Proposition 4

```
eval test "?msd_fib An n>=2 => (Es,t ($hofstadter(n-2,s) & $hofstadter(n-1,t)
     & (s<t => G[n]+G[t]=1)))":
eval test2 "?msd_fib An n>=2 => (Es,t ($hofstadter(n-2,s) & $hofstadter(n-1,t)
     & (s=t => G[n]=1)))":
```

and both commands return `TRUE`. □

2.2 A heuristic generating φ-morphisms

We present here a heuristic that, given a (long enough) prefix of an infinite word $\mathbf{w}$ being the coding of a fixed point of what we call a φ-morphism, suggests a morphism μ and a coding ρ such that $\mathbf{w} = \rho(\mu^\omega(0))$. This heuristic will be used many times (it was already used by Duchêne et al. [6] but not explicitly described). Once the morphism is obtained, we can then prove its correctness with `Walnut`.

Definition 7. A morphism $\mu : A^* \to A^*$ is a *φ-morphism*, if for any fixed point $\mathbf{w} = w_0 w_1 \cdots$ of μ,

- if $\mathrm{rep}_F(n)$ ends with 1, then $\mu(w_n) = w_{\mathrm{rep}_F(n)0}$,
- if $\mathrm{rep}_F(n)$ ends with 0, then $\mu(w_n) = w_{\mathrm{rep}_F(n)0} w_{\mathrm{rep}_F(n)1}$.

We have chosen a simplified presentation, adapted to the situation considered here. Indeed, this is a special case of what is called a U-substitution by Bruyère and Hansel [2] for a wide class of numeration systems. Let $\sigma : a \mapsto ab$, $b \mapsto a$ be the usual Fibonacci morphism generating the Fibonacci word $\mathbf{f} = abaababaa\cdots$. If $\mu : A^* \to A^*$ is a φ-morphism, then there is a coding $f : A^* \to \{a, b\}^*$ such that

$$f(\mu(i)) = \sigma(f(i)), \text{ for all } i \in A. \tag{2}$$

Roughly speaking, μ is defined on a larger alphabet than $\{a, b\}$ but it preserves the underlying structure of the morphism σ.

Example 8. The morphism

$$\mu : 0 \mapsto 01,\ 1 \mapsto 2,\ 2 \mapsto 31,\ 3 \mapsto 45,\ 4 \mapsto 35,\ 5 \mapsto 4$$

is a φ-morphism. Consider the coding $f : 0, 2, 3, 4 \mapsto a$ and $1, 5 \mapsto b$.

Definition 9. We will often associate a DFA with a morphism $\mu : A^* \to A^*$. The set of states is the alphabet A, the alphabet of the automaton is $0, 1$. This is a classical construction that goes back to Cobham [3] and it corresponds to the `promote` operation in `Walnut`. The transitions of the DFA are given by the morphism: if $\mu(c) = de$, $c, d, e \in A$, then there are edges from c to d and e with respective labels 0 and 1; if $\mu(c) = d$, $c, d \in A$, then there is a single edge from c to d labeled by 0. If we have an extra coding $\rho : A \to B$ (i.e., a letter-to-letter morphism), then the DFA can be turned into a DFAO where the output function is exactly ρ.

With the morphism from Example 8, the corresponding automaton is depicted in Figure 2. Property (2) means that there is a morphism of automata between the one associated with μ and the one associated with the Fibonacci morphism: if there is a transition between c and d with label i, then there is also one between $f(c)$ and $f(d)$ with the same label i.

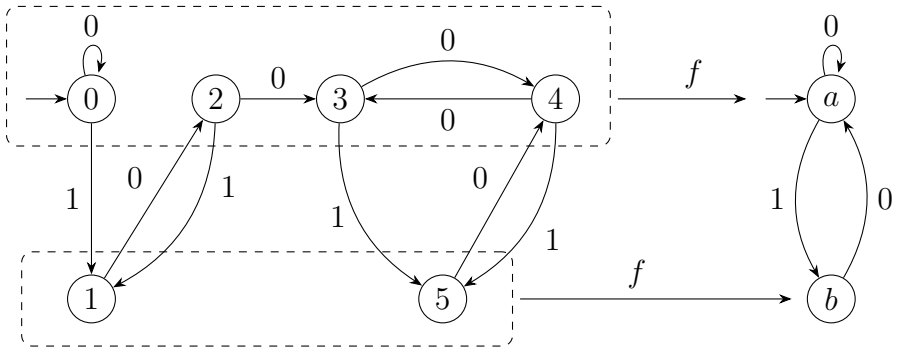

Figure 2: Automata associated with μ and the Fibonacci morphism.

Heuristic. We describe a useful heuristic to devise the automaton in Figure 1 (as well as many other ones in the article). Assume that the sequence g in Table 1 is the fixed point of a φ-morphism μ. In this case, we start from a single letter g_n, $n \geq 0$, then iterate the morphism a finite number of times (here, three times) and record the different factors obtained. Since we assume that μ is a φ-morphism, we know precisely where the factors $\mu^i(g_n)$ are located in the sequence.

Precisely, if $\mathrm{rep}_F(n) = u$, consider the Fibonacci representations of length $|u|+i$ having u as a prefix. These words give the positions of the letters in $\mu^i(g_n)$. Let $\alpha(i,n)$ and $\beta(i,n)$ be the first and last positions (Figure 3 may help the reader). Formally, a t-type is a $(t+1)$-tuple made of factors in prescribed positions depending only on n

$$(g_n,\ g_{\alpha(1,n)} \cdots g_{\beta(1,n)},\ \ldots,\ g_{\alpha(t,n)} \cdots g_{\beta(t,n)}).$$

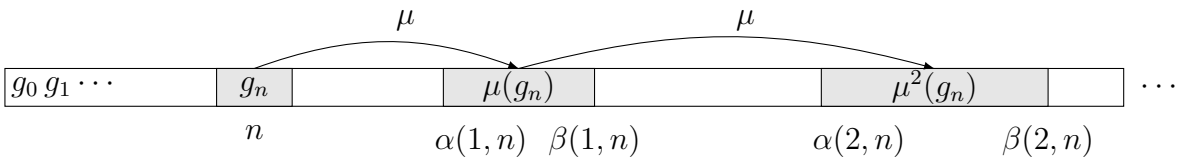

Figure 3: Construction of a type.

Over a long prefix, we observe only six distinct 3-types (and this number remains stable when the number of iterations increases). As an example, starting from $g_4 = 0$, since $\mathrm{rep}_F(4) = 101$, this word can be extended to 1010 whose value is 7, and $g_7 = 1$. Now 1010 can be extended to 10100 and 10101 giving $g_{11}g_{12} = 10$. Finally these two Fibonacci representations can again be extended to 101000, 101001 and 101010 giving $g_{18}g_{19}g_{20} = 011$. This is the second line in Table 2. The reader may notice that the six elements in the column $\mu^2(g_n)$ are pairwise distinct. The last column does not bring more information to distinguish the rows of the table. So on this example, considering 2-types would have been enough. However, 1-types are not enough: we cannot distinguish between the first and last rows.

	g_n	$\mu(g_n)$	$\mu^2(g_n)$	$\mu^3(g_n)$
0	1	0	1	10
1	0	1	10	011
2	1	10	011	11010
3	1	01	110	01011
4	0	11	010	11011
5	1	0	11	010

Table 2: The different 3-types when applying a φ-morphism to g.

Now we associate each symbol g_n with its 3-type and get the infinite word

$$01|2|31|45|2|35|4|31|45|4|35|45|\cdots,$$

where the vertical bars indicate the factorization of the Fibonacci word $ab|a|ab|ab|a|\cdots$ with factors $\sigma(a) = ab$ and $\sigma(b) = a$, the images of the letters by the Fibonacci morphism. From this, we deduce the morphism μ given in Example 8: the image of the nth letter has to be the nth block in the factorization. So $0 \mapsto 01$, then $1 \mapsto 2$ and so on. From the morphism, it is a routine to obtain the DFA in Figure 1; see Definition 9.

Finally, one may define a coding ρ to recover the original sequence where each type is mapped to the first element of the tuple. So, $0, 2, 3, 5 \mapsto 1$ and $1, 4 \mapsto 0$. This can be equivalently performed by comparing the morphic word and the original sequence

$$01231452354314543545\cdots$$
$$10110011110100101101\cdots$$

and matching corresponding elements.

2.3 An automatic proof for K^2

Now that g has been defined by a DFA fed with Fibonacci representations (and stored as `G.txt` in `Word Automata Lib`), the set of $\mathcal{P}$-positions (1) obtained by Komak et al. [19] can be described by the following commands:

```
def pposK2_asym "?msd_fib (a+b<=1) | En,x $phin(n,x) & a+1=x+G[n]
                                                    & b=x+n+G[n]":
def pposK2 "?msd_fib $pposK2_asym(a,b) | $pposK2_asym(b,a) ":
```

The resulting automaton has 27 states.

For the sake of presentation, let us recall some basics on combinatorial games.

Definition 10. A set S of positions is

- *stable* if, for all $s, t \in S$ with $s \neq t$, there is no move between s and t;

- *absorbing* if, for all $t \notin S$, there exists $s \in S$ such that there is a move from t to s (or, t has an option in S).

It is a well-known property in graph theory that every acyclic graph has a unique kernel, i.e., a stable and absorbing set of vertices. In a take-away game where tokens are removed from piles, the corresponding game graph is acyclic: it is not possible to visit the same position twice during a game.

Proposition 11 (Folklore)**.** *The sets of $\mathcal{P}$- and $\mathcal{N}$-positions of an impartial acyclic game are uniquely determined by the following two properties:*

1. *Every move from a $\mathcal{P}$-position leads to an $\mathcal{N}$-position; equivalently, there is no move between two $\mathcal{P}$-positions (stability property of $\mathcal{P}(G)$).*

2. *From every $\mathcal{N}$-position, there exists a move leading to a $\mathcal{P}$-position (absorbing property of $\mathcal{P}(G)$).*

As observed in our previous work [23], it is then a routine test to check the stability and absorbing properties of a given candidate set:

```
eval stableK2 "?msd_fib Ap,q,r,s (($pposK2(p,q) & $pposK2(r,s)
     & p>=r & q>=s & p+q>2) => ((p=r & q=s) | (p>r & q>s & p+s!=q+r)))":
```

In this first command, the predicate `(p=r & q=s)` takes care of one-pile moves: if (p, q) and (r, s) are two $\mathcal{P}$-positions, then we need to have

$$\neg(p = r \wedge q \neq s) \wedge \neg(p \neq r \wedge q = s),$$

otherwise there would exist a one-pile move between those two positions. It is then straightforward to see that the previous assertion is equivalent to

$$(p = r \wedge q = s) \vee (p \neq r \wedge q \neq s).$$

The second part of the statement is then handled by the end of the formula, since it deals with diagonal two-pile moves.

```
eval absorbingK2 "?msd_fib Ap,q (~$pposK2(p,q) => Ex,y
     (x<=p & y<=q & $pposK2(x,y) & (p+y=q+x | p=x | q=y))) ":
```

These two commands return `TRUE`, resulting in the result of Komak et al. [19, Thm. 3]. From Proposition 11, we immediately get an alternative proof of Theorem 2.

3 Towards an algebraic characterization for K^ℓ

Komak et al. [19] chose $\{(x, y) \mid x + y \leq 2\}$ as the set of terminal positions. It is natural to consider a parameterized version where the set of terminal states is

$$\{(x, y) \in \mathbb{N}^2 \mid x + y \leq \ell\}$$

for some $\ell \in \mathbb{N}$. The corresponding game is denoted by K^ℓ. Note that the game K^0 is the classical Wythoff game. In this section, we obtain new results concerning the games K^1 and K^3.

3.1 The K^1-game

Up to our knowledge, the game K^1 does not seem to have been studied in the literature. Computing the first few $\mathcal{P}$-positions of K^1 and writing their Fibonacci expansions permitted us to state a conjecture that can be tested with `Walnut`. This result is quite similar to the famous characterization of the $\mathcal{P}$-positions of the Wythoff game obtained by Fraenkel [9].

a_n	$\text{rep}_F(a_n)$	b_n	$\text{rep}_F(b_n)$
0	ε	1	1
2	10	4	101
3	100	6	1001
5	1000	9	10001
7	1010	12	10101
8	10000	14	100001
10	10010	17	100101
11	10100	19	101001
13	100000	22	1000001
15	100010	25	1000101

Table 3: Representations of the first few $\mathcal{P}$-positions in K^1.

Theorem 12. *A position (a, b), with $a \leq b$, is a $\mathcal{P}$-position of the game K^1, with a set of terminal positions being $\{(x, y) \mid x + y \leq 1\}$ and Wythoff's moves, if and only if $\text{rep}_F(a)$ ends with 0 and $\text{rep}_F(b) = \text{rep}_F(a)1$.*

The candidate set of $\mathcal{P}$-positions can be readily implemented.

```
reg end_zero msd_fib "(0|1)*0":
reg add_one {0,1} {0,1} "([0,0]|([0,1][1,1]*[1,0]))*[0,1]":
def pposK1_asym "?msd_fib (a=0 & b=0) | ($end_zero(a) & $add_one(a,b))":
def pposK1 "?msd_fib $pposK1_asym(a,b) | $pposK1_asym(b,a)":
```

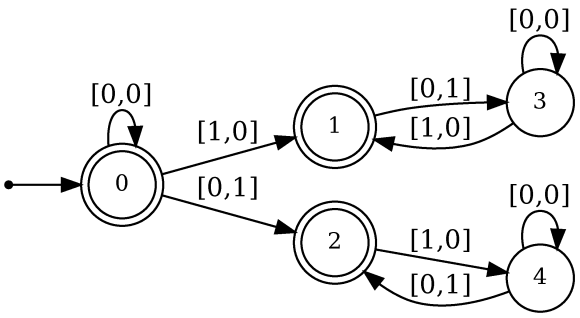

Figure 4: A DFA recognizing $\mathcal{P}$-positions of K^1.

The corresponding DFA is depicted in Figure 4. The sequence $(a_n)_{n\geq 0} = 0, 2, 3, 5, 7, 8, 10, \ldots$ (first component of the $\mathcal{P}$-positions)corresponds to the Fibonacci-even numbers, listed as A022342 (as the binary expansion of an even integer ends with zero). The complementary sequence $(b_n)_{n\geq 0}$ appears in the OEIS as A003622.

Proof. The proof is a routine procedure. One has to check that the candidate set is stable and absorbing:

```
eval stableK1 "?msd_fib Ap,q,r,s (($pposK1(p,q) & $pposK1(r,s) & p>=r & q>=s
      & p+q>1) => ((p=r & q=s) | (p>r & q>s & p+s!=q+r)))":
eval absorbingK1 "?msd_fib Ap,q (~$pposK1(p,q) => Ex,y (x<=p & y<=q
      & $pposK1(x,y) & (p+y=q+x | p=x | q=y)))":
```

Both commands evaluate to `TRUE`. The conclusion follows from Proposition 11. □

3.2 The K^3-game

The case of K^3 is more interesting. In Figure 5, we have depicted the first few $\mathcal{P}$-positions of K^2 and K^3. Green (resp., red) squares represent the $\mathcal{P}$-positions of K^3 (resp., K^2). The common $\mathcal{P}$-positions are colored in blue.

The first few non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\geq 0}$ are given in Table 4. From now on, we take the following convention: the terminal positions are not taken into account when indexing the $\mathcal{P}$-positions of the game. Thus, (a_0, b_0) represents the first non-terminal $\mathcal{P}$-position. Due to the symmetry of the game, we only consider sequences where $a_n < b_n$.

Theorem 13. *There exists a Fibonacci-automatic function $g_3 : \mathbb{N} \to \{0, 1, 2\}$ such that, for all $n \geq 0$, the nth non-terminal $\mathcal{P}$-position (a_n, b_n) of K^3 is given by*

$$a_n = \lfloor (n+2)\phi \rfloor + g_3(n+1) - 1 \quad \text{and} \quad b_n = \lfloor (n+2)\phi^2 \rfloor + g_3(n+1) + 1. \tag{3}$$

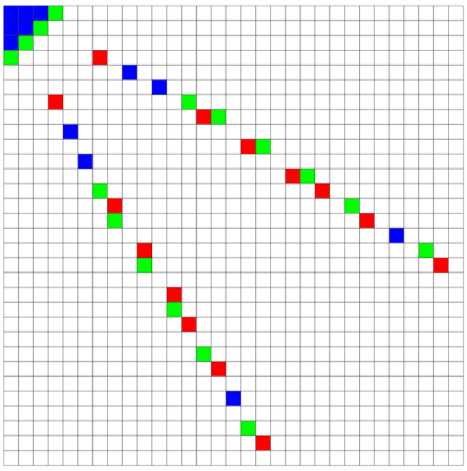

Figure 5: Comparing the $\mathcal{P}$-positions of K^2 (red) and K^3 (green).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_n	4	5	6	7	9	11	13	15	16	18	19	21	22	24	25	27	29	30
b_n	8	10	12	14	17	20	23	26	28	31	33	36	38	41	43	46	49	51

Table 4: The first few (non-terminal) $\mathcal{P}$-positions of K^3.

The first few values of $g_3(n)$ are

$$122101122121111111112112112121121112111121111111211112\cdots.$$

Using the heuristic described in Section 2.2, assuming that the above infinite word is generated by a φ-morphism, we obtain the morphism

$$\mu_3 : 0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 34, 3 \mapsto 56, 4 \mapsto 7, 5 \mapsto 78,$$
$$6 \mapsto 9, 7 \mapsto ab, 8 \mapsto a, 9 \mapsto 56, a \mapsto ab, b \mapsto 7$$

and the coding

$$\rho_3 : 0, 3, 5, 6, 8, a, b \mapsto 1, \quad 1, 2, 7, 9 \mapsto 2, \quad 4 \mapsto 0.$$

We thus define $g_3(n)$ as the nth symbol in $\rho_3(\mu_3^\omega(0))$. Note that producing enough values of the sequence g_3 requires computing sufficiently many $\mathcal{P}$-positions of K^3.

Proof. Since we have a morphism μ_3 and a coding ρ_3, we can store in `Word Automata Lib` a text file `K3.txt` describing a 12-state DFAO. The transitions are given by μ_3 and the outputs by ρ_3. Since we have a candidate set in the statement, (3) can be encoded as follows.

```
def pposK3_asym "?msd_fib (a+b<=3) | (En,x $phin(n+2,x)
                    & a+1=x+K3[n+1] & b=x+n+3+K3[n+1])":
def pposK3 "?msd_fib $pposK3_asym(a,b) | $pposK3_asym(b,a)":
```

Thanks to Proposition 11, we simply have to check the stability and absorption of this set.

```
eval stableK3 "?msd_fib Ap,q,r,s (($pposK3(p,q) & $pposK3(r,s)
  & p>=r & q>=s & p+q>3) => ((p=r & q=s) | (p>r & q>s & p+s!=q+r)))":
eval absorbingK3 "?msd_fib Ap,q (~$pposK3(p,q) => Ex,y
        (x<=p & y<=q & $pposK3(x,y) & (p+y=q+x | p=x | q=y))) ":
```

Both commands return `TRUE` (intermediate computations need at most 449 states). □

3.3 Even further

For the game K^4, we have applied the same procedure. The first few non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\geq 0}$ are given in Table 5. To get a long enough prefix, we have computed

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_n	5	6	7	8	9	11	13	15	17	19	20	22	23	25	26	28	29	31
b_n	10	12	14	16	18	21	24	27	30	33	35	38	40	43	45	48	50	53

Table 5: The first few (non-terminal) $\mathcal{P}$-positions of K^4.

the first 600 $\mathcal{P}$-positions (a_n, b_n) with $a_n \leq b_n$ (up to a_n close to 1000). From this, we conjectured that

$$a_n = \lfloor (n+2)\phi \rfloor + g_4(n+1) \quad \text{and} \quad b_n = \lfloor (n+2)\phi^2 \rfloor + g_4(n+1) + 3, \tag{4}$$

where the first few values of $g_4(n)$ are

$$122100011121111110101101101111111111111111\cdots.$$

Having the first 600 values of $g_4(n)$, we apply the heuristic of Section 2.2. Note that, in order to distinguish the types correctly, we have to consider 4-types, because there exist distinct 4-types whose restrictions to 3-types coincide. Since we have to take fourth-order iterations, this explains why we need a longer prefix. We obtain 18 different 4-types and the heuristic gives the following morphisms:

$$\mu_4 : 0 \mapsto 01,\ 1 \mapsto 2,\ 2 \mapsto 34,\ 3 \mapsto 56,\ 4 \mapsto 7,\ 5 \mapsto 89,\ 6 \mapsto a,\ 7 \mapsto bc,\ 8 \mapsto dc,$$
$$9 \mapsto d,\ a \mapsto ef,\ b \mapsto ef,\ c \mapsto e,\ d \mapsto 7g,\ e \mapsto eh,\ f \mapsto b,\ g \mapsto d,\ h \mapsto b,$$

$$\rho_4 : 0, 3, 7, 8, 9, b, c, d, e, h \mapsto 1,\ 1, 2, a \mapsto 2,\ 4, 5, 6, f, g \mapsto 0.$$

With $g_4(n)$ being the nth symbol of $\rho_4(\mu_4^\omega(0))$, we have a 18-state DFAO. Formulas (4) translate into `Walnut` as follows:

```
def pposK4_asym "?msd_fib (a+b<=4) | (En,x $phin(n+2,x) & (a=x+K4[n+1])
                                                & (b=x+K4[n+1]+n+5))":
```

and we then follow exactly the same proof to get the result.

Theorem 14. *The $\mathcal{P}$-positions (a, b), with $a \leq b$, of the game K^4, with a set of terminal positions being $\{(x, y) \mid x + y \leq 4\}$ and Wythoff's moves, are characterized by (4).*

Thus, aside from the fact that the morphisms μ_ℓ and ρ_ℓ used to describe the set of $\mathcal{P}$-positions are defined over increasingly large alphabets—which requires computing a sufficiently long prefix—these preliminary results suggest that such a result could be extended to any arbitrary value of the parameter ℓ. In Section 4, we give a recursive characterization of the $\mathcal{P}$-positions for all values of the parameter $\ell \geq 1$. From that result, we are able to give, with Corollary 19, the general behavior of the sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$.

4 A recursive characterization for K^ℓ

It is common in the combinatorial game literature [7, 9, 10, 11] to have several characterizations of the set of $\mathcal{P}$-positions (algebraic, recursive, combinatorial, morphic, ...). In this section, we provide a recursive characterization of the set of $\mathcal{P}$-positions of K^ℓ, for all $\ell \geq 1$.

First, we consider small values of the parameter ($\ell = 1$ and $\ell = 2$) and give an automatic proof. It is then not difficult to state a general conjecture that we are able to prove by classical methods (stability and absorption). Indeed, it is not possible to obtain an automated parameterized proof. Here we clearly see how the two approaches complement each other: `Walnut` allows us to conjecture general statements that are then proved in the classical way.

Let us recall our indexing convention: (a_0, b_0) is the first non-terminal $\mathcal{P}$-position of K^ℓ (with $a_0 < b_0$). It is indeed not necessary to encode the terminal positions, which are set in advance and known to both players.

Let us begin with some classical and well-known results on the Wythoff game [28] before moving on to the games discussed in this article. We recall the *recursive definition* of the $\mathcal{P}$-positions of the Wythoff game K^0, with our convention:

$$(a_0, b_0) = (1, 2) \quad \text{and} \quad \forall n \geq 1, \begin{cases} a_n = \text{MeX}(\{a_i, b_i \mid i < n\} \cup \{0\}), \\ b_n = a_n + n + 1, \end{cases}$$

where MeX stands for *minimum excluded value*, i.e. the least non-negative integer not in the set. In other words, $\text{MeX}\, S = \min(\mathbb{N} \setminus S)$. So $a_1 = \text{MeX}\{0, 1, 2\} = 3$ and $b_1 = a_1 + 2 = 5$, then $a_2 = \text{MeX}\{0, 1, 2, 3, 5\} = 4$ and $b_2 = a_2 + 3 = 7$ and so on.

We now turn to the games discussed in this paper. For K^1 (see Table 3), we have the following result.

Proposition 15. *Define the sequence $(a_n, b_n)_{n\geq 0}$ recursively by*

$$(a_0, b_0) = (2, 4) \quad \textit{and} \quad \forall n \geq 1, \begin{cases} a_n = \text{MeX}(\{a_i, b_i \mid i < n\} \cup \{0, 1\}), \\ b_n = a_n + n + 2. \end{cases}$$

The set of non-terminal $\mathcal{P}$-positions of K^1 is $\{(a_n, b_n) \mid n \geq 0\} \cup \{(b_n, a_n) \mid n \geq 0\}$.

Proof. It is clear from Theorem 12 that the sequences $(a_n)_{n\geq 0} = 2, 3, 5, 7, \ldots$ and $(b_n)_{n\geq 0} = 4, 6, 9, 12, \ldots$ form a partition of $\mathbb{N}_{\geq 2}$. Indeed, any integer has a unique representation in the Fibonacci system, and each representation ends with either 0 or 01. We first define two predicates for the sets of pairs (a_n, n) and (b_n, n), respectively. We simply make use of the shift (observe that, in Table 3, suppressing the last zero from every representation in the first column gives the set of all valid Fibonacci representations). We use $n+1$ instead of n because of our indexing convention.

```
def pposK1_an "?msd_fib $shift(n+1,a)":
```

As an example, we have

```
eval test "?msd_fib $pposK1_an(2,0) & $pposK1_an(3,1) &
                    $pposK1_an(5,2) & $pposK1_an(7,3)":
```

Similarly, we shift twice and add 1 to get the last column in Table 3,

```
def pposK1_bn "?msd_fib Et,c $shift(n+1,t) & $shift(t,c) & b=c+1":
```

For instance,

```
eval test "?msd_fib $pposK1_bn(4,0) & $pposK1_bn(6,1) &
                    $pposK1_bn(9,2) & $pposK1_bn(12,3)":
```

Since we have a partition, we only have to check that, for all $n \geq 0$, $b_n = a_n + n + 2$, which is readily verified with

```
eval test "?msd_fib An Ea,b $pposK1_an(a,n) & $pposK1_bn(b,n) & b=a+n+2":
```

□

For K^2 (see Figure 5), we get the following recursive characterization.

Proposition 16. *Define the sequence $(a_n, b_n)_{n\geq 0}$ recursively by*

$$(a_0, b_0) = (3, 6) \quad \text{and} \quad \forall n \geq 1, \begin{cases} a_n = \mathrm{MeX}(\{a_i, b_i \mid i < n\} \cup \{0, 1, 2\}), \\ b_n = a_n + n + 3. \end{cases}$$

The set of non-terminal $\mathcal{P}$-positions of K^2 is $\{(a_n, b_n) \mid n \geq 0\} \cup \{(b_n, a_n) \mid n \geq 0\}$.

Proof. The sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ are Fibonacci-synchronized; see [26, Sec. 10.11].

```
def pposK2_an "?msd_fib Ex ($phin(n+2,x) & a+1=x+G[n+2])":
```

This predicate defines the pairs (a_n, n), with $n \geq 0$. As an example, the following command returns `TRUE`:

```
eval test "?msd_fib $pposK2_an(3,0) & $pposK2_an(4,1)
                 & $pposK2_an(5,2) & $pposK2_an(7,3)":
```

Similarly, the following predicate defines the pairs (b_n, n), with $n \geq 0$.

```
def pposK2_bn "?msd_fib Ex $phin(n+2,x) & b=x+n+2+G[n+2]":
eval test "?msd_fib $pposK2_bn(6,0) & $pposK2_bn(8,1)
                  & $pposK2_bn(10,2) & $pposK2_bn(13,3)":
```

The difference of indices with Section 2.3 comes from our indexing convention. The next two commands show that the two sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ make a partition of $\mathbb{N}_{\geq 3}$:

```
eval covering "?msd_fib Ax (x>=3 => En ($pposK2_an(x,n) | $pposK2_bn(x,n)))":
eval notboth "?msd_fib ~(Em,n,x ($pposK2_an(x,m) & $pposK2_bn(x,n)))":
```

The first one means that every integer $x \geq 3$ belongs to at least one of the two sequences. The second one means that there is no $x \geq 3$ that is simultaneously of the form a_m and b_n. Synchronization is only needed to show that $b_n - a_n - n = 3$. Finally, we can verify that if $a_n = x$, then $b_n = x + n + 3$.

```
eval alinkb "?msd_fib  Ax (x>=3 => (En ($pposK2_an(x,n)
                                  => $pposK2_bn(x+n+3,n))))":
```

This could have also been immediately derived from the expression (1). The fact that $b_n = a_n + n + 3$ and that the sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ make a partition of $\mathbb{N}_{\geq 3}$ imply the MeX definition of the a_n's (roughly speaking, gaps are filled with a's). □

Now that we have inspected the cases K^1 and K^2, the next statement seems natural.

Theorem 17. *Let $\ell \geq 1$ and define $(a_n, b_n)_{n\geq 0}$ recursively by*

$$(a_0, b_0) = (\ell + 1, 2\ell + 2) \quad \text{and} \quad \forall n \geq 1, \begin{cases} a_n = \mathrm{MeX}(\{a_i, b_i \mid i < n\} \cup \{0, 1, \ldots, \ell\}), \\ b_n = a_n + n + \ell + 1. \end{cases} \tag{5}$$

The set of non-terminal $\mathcal{P}$-positions of K^ℓ is $\{(a_n, b_n) \mid n \geq 0\} \cup \{(b_n, a_n) \mid n \geq 0\}$.

We will repeatedly use the following properties all along the proofs of the next three results.

Lemma 18. *For the sequences (5) defined above, we have the following properties.*

1. *The sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ are strictly increasing.*
2. *They make a partition of $\mathbb{N}_{>\ell}$.*
3. *For each $j > \ell$, there exist a unique n such that $j = b_n - a_n$.*
4. *For all n, $a_{n+1} - a_n \in \{1, 2\}$ and $b_{n+1} - b_n \in \{2, 3\}$. In particular, $b_{n+1} - b_n = a_{n+1} - a_n + 1$.*

Proof. These properties obviously follow from the MeX definition; we provide a proof for the sake of completeness.

From the MeX definition, each a_n is a new integer strictly larger than a_{n-1}, so $(a_n)_{n\geq 0}$ is strictly increasing. From this, we get $b_n = a_n + n + \ell + 1 > a_{n-1} + n + \ell = b_{n-1}$. Hence $(b_n)_{n\geq 0}$ is strictly increasing, and the two sequences are disjoint.

Suppose some $j > \ell$ never appears in the two sequences, and take the smallest such j. Since $(a_n)_{n\geq 0}$ is strictly increasing, let n be minimal such that $a_n > j$. By minimality of j, all integers in $\{\ell + 1, \ldots, j - 1\}$ already appear among a_i, b_i for $i < n$, so j is the smallest unused integer at step n, forcing $a_n = j$, a contradiction. Hence $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ form a partition of $\mathbb{N}_{>\ell}$.

The identity $b_n - a_n = n + \ell + 1$ shows that for each $j > \ell$ there is a unique n with $j = b_n - a_n$. Finally, since a_{n+1} is the MeX of the previous values, it cannot exceed $a_n + 2$, for otherwise either $a_n + 1$ or $a_n + 2$ would be unused and smaller; thus $a_{n+1} - a_n \in \{1, 2\}$. From $b_n = a_n + \ell + 1 + n$ we obtain $b_{n+1} - b_n = (a_{n+1} - a_n) + 1 \in \{2, 3\}$.

□

Proof of Theorem 17. Let $\mathcal{P}$ be the union of the set $\{(a_n, b_n) \mid n \geq 0\} \cup \{(b_n, a_n) \mid n \geq 0\}$ and the set of terminal positions $\{(x, y) \mid x + y \leq \ell\}$. We use Proposition 11 and prove the stability and absorption of $\mathcal{P}$.

Stability. Let (x, y) be in $\mathcal{P}$. We may assume that $x + y > \ell$, since there is no allowed move from a terminal position. Let thus (x, y) be of the form (a_n, b_n) for some $n \geq 0$. We have to show that every move leads to a position not in $\mathcal{P}$. The case (b_n, a_n) is symmetric and not discussed below.

- A vertical move will lead to a position (a_n, y'), with $0 \leq y' < b_n$. Since $a_n \geq \ell + 1$, for all $n \geq 0$, (a_n, y') is not a terminal position: $a_n + y' > \ell$. To get a contradiction, assume that (a_n, y') is in $\mathcal{P}$. Then (a_n, y') is either of the form (a_m, b_m) or (b_m, a_m).
  - If $(a_n, y') = (a_m, b_m)$, then $a_n = a_m$. Since the sequence $(a_n)_{n\geq 0}$ is increasing, we get $n = m$ and thus $y' = b_n$, contradicting $y' < b_n$.
  - If $(a_n, y') = (b_m, a_m)$, then $a_n = b_m$. This contradicts the fact that
$$\{a_n \mid n \geq 0\} \cap \{b_n \mid n \geq 0\} = \emptyset.$$
- The proof is the same for a horizontal move leading to (x', b_n), with $0 \leq x' < a_n$.
- Now consider a diagonal move leading to a position $(a_n - t, b_n - t)$ for some $t \in \{1, \ldots, a_n\}$. First, such a position is not a terminal one because
$$a_n - t + b_n - t = a_n + b_n - 2t \geq a_n + b_n - 2a_n = n + \ell + 1 > \ell.$$
Assume that $(a_n - t, b_n - t)$ is of the form (a_m, b_m), then
$$b_m - a_m = b_n - t - (a_n - t) = n + \ell + 1,$$

and thus $m = n$. It implies $t = 0$, a contradiction.

Finally, assume that $(a_n - t, b_n - t)$ is of the form (b_m, a_m), then

$$b_m - a_m = a_n - t - (b_n - t) = -(n + \ell + 1) < 0,$$

which is again a contradiction because $b_m - a_m = m + \ell + 1 > 0$.

Absorption. Let (x, y) be a position not in $\mathcal{P}$. In particular, $x + y > \ell$. We have to prove that there is a move from (x, y) into $\mathcal{P}$. By symmetry, we may assume that $x \leq y$.

- Case 1: $y - x \leq \ell$. We will prove that there is a diagonal move leading to a terminal position. Consider the quantity

$$t = \left\lceil \frac{x + y - \ell}{2} \right\rceil.$$

Since $x + y > \ell$, $t \geq 1$. Note that

$$x + y - \ell = 2x + (y - x) - \ell \leq 2x.$$

Hence $t \leq x$ and $(x, y) \to (x - t, y - t)$ is a legal move. To conclude with this part, $(x - t, y - t)$ is a terminal move:

$$x - t + y - t \leq x + y - (x + y - \ell) = \ell.$$

- Case 2: $y - x > \ell$. There exists a unique $n \geq 0$ such that $y - x = n + \ell + 1 = b_n - a_n$.
  - If $x \geq a_n$, consider $t = x - a_n$. If $t > 0$, we can play the diagonal move $(x, y) \to (x - t, y - t) = (a_n, y - x + a_n) = (a_n, b_n)$. If $t = 0$, then $x = a_n$, and thus $y = b_n$. However, (x, y) is not in $\mathcal{P}$, so this case does not occur.
  - If $x \leq \ell < a_n$, then a vertical move $(x, y) \to (x, 0)$ leads to a terminal position.
  - If $\ell < x < a_n$, then x is of the form a_m or b_m for some m (because we have a partition of $\mathbb{N}_{>\ell}$). If $x = a_m$, then $m < n$, since the sequence $(a_n)_{n \geq 0}$ is increasing. Thus $y - x = b_n - a_n > b_m - a_m$, so that $y > b_m$. A vertical move $(x, y) \to (x, b_m) = (a_m, b_m)$ is thus enough. Finally, if $x = b_m$, observe that $y \geq x = b_m > a_m$, so we may consider the move $(x, y) \to (x, a_m) = (b_m, a_m)$.

□

We now show that the points (a_n, b_n) remain at bounded distance from a line of slope ϕ (see, for instance, Figure 5). In the proof, recursively indexed sequences of the form x_{x_n} play an important role. Note that the solution of Wythoff satisfies the complementary equation $x_{x_n} = y_n - 1$; see [16, 17, 18]. Our approach is in the same spirit as the work of Fraenkel and Peled [12], where uniform bounds for MeX-defined complementary sequences are derived.

Corollary 19. *For the sequences defined by (5), there exists a bounded function $\lambda_\ell : \mathbb{N} \to \mathbb{Z}$ such that, for all $n \geq 0$,*

$$a_n = \lfloor (n+\ell)\phi \rfloor + \lambda_\ell(n).$$

In particular, $b_n = \lfloor (n+\ell)\phi^2 \rfloor + \lambda_\ell(n) + 1$.

Proof. Let

$$S_n := \sum_{k=1}^{n} (a_k - a_{k-1} - 1).$$

Since $a_k - a_{k-1} = 2$ if and only if some b_j is inserted between a_{k-1} and a_k, we have

$$S_n = \#\{i \mid b_i < a_n\}.$$

That is, $b_0 < b_1 < \cdots < b_{S_n-1} < a_n < b_{S_n}$. From the MeX definition, $a_0, \ldots, a_\ell$ are consecutive integers, so that $S_0 = S_1 = \cdots = S_\ell = 0$. Moreover, $S_{\ell+1} = 1$, and $(S_n)_{n\geq 0}$ is non-decreasing.

The interval $[a_0, a_n]$ contains $n+1$ terms a_j's and S_n terms b_j's, i.e., $n+1+S_n = a_n - a_0 + 1$. Since $a_0 = \ell + 1$, we have

$$a_n = \ell + 1 + n + S_n. \tag{6}$$

By definition of S_n, we have

$$\begin{aligned} & b_{S_n-1} < a_n < b_{S_n} \\ \Leftrightarrow\ & a_{S_n-1} + S_n - 1 + \ell + 1 < a_n < a_{S_n} + S_n + \ell + 1 \\ \Leftrightarrow\ & a_{S_n-1} - 1 < a_n - S_n - \ell - 1 < a_{S_n} \\ \Leftrightarrow\ & a_{S_n-1} \leq a_n - S_n - \ell - 1 < a_{S_n}. \end{aligned}$$

Now using (6),

$$\begin{aligned} & \ell + 1 + S_n - 1 + S_{S_n-1} \leq n < \ell + 1 + S_n + S_{S_n} \\ \Leftrightarrow\ & S_{S_n-1} - 1 \leq n - \ell - 1 - S_n < S_{S_n}. \end{aligned}$$

Note that $S_{S_n} - S_{S_n-1} = a_{S_n} - a_{S_n-1} - 1 \in \{0, 1\}$. So, if $S_{S_n} - S_{S_n-1} = 0$, then the integer $n - \ell - 1 - S_n$ belongs to $[S_{S_n} - 1, S_{S_n})$, and is thus equal to $S_{S_n} - 1$. If $S_{S_n} - S_{S_n-1} = 1$, then $n - \ell - 1 - S_n$ belongs to $[S_{S_n} - 2, S_{S_n})$. Consequently,

$$n - \ell - 1 - S_n = S_{S_n} - 1 - \epsilon_n, \text{ for some } \epsilon_n \in \{0, 1\}.$$

That is,

$$S_n + S_{S_n} = n - \ell + \epsilon_n.$$

We want to prove that $|a_n - n\phi|$ is bounded by a constant. Thanks to (6), observe that

$$a_n - n\phi = \ell + 1 + S_n - (\phi - 1)n = \ell + 1 + S_n - \frac{n}{\phi}.$$

So it is enough to bound $|D_n|$, where

$$D_n := S_n - \frac{n}{\phi}.$$

We have

$$S_n + S_{S_n} = D_n + \frac{n}{\phi} + D_{S_n} + \frac{S_n}{\phi} = n - \ell + \epsilon_n$$

and

$$\frac{S_n}{\phi} = \frac{1}{\phi}\left(D_n + \frac{n}{\phi}\right).$$

Putting these together, we get

$$\left(1 + \frac{1}{\phi}\right) D_n + D_{S_n} + n\left(\frac{1}{\phi} + \frac{1}{\phi^2}\right) = n - \ell + \epsilon_n.$$

Hence,

$$\phi D_n = -\ell + \epsilon_n - D_{S_n}$$

and

$$\phi\,|D_n| \le \ell + |D_{S_n}|\,. \tag{7}$$

We claim that, for all n,

$$|D_n| \le \frac{\ell}{\phi - 1} = \phi\ell.$$

Note that for $i \le \ell$, $|D_i| = i/\phi < \ell/(\phi - 1) = \phi\ell$. For $i > \ell$, we proceed by contradiction. Assume that there exists a minimal $N > \ell$ such that $|D_N| > \phi\ell$. We have $S_N < N$, so the minimality of N implies

$$|D_{S_N}| \le \phi\ell.$$

From (7), we have

$$|D_N| \le \frac{\ell}{\phi} + \frac{|D_{S_N}|}{\phi} \le \frac{\ell}{\phi} + \frac{\phi\ell}{\phi} = \left(\frac{1}{\phi} + 1\right)\ell = \phi\ell,$$

which is a contradiction. Consequently, using (6),

$$a_n - \lfloor (n+\ell)\phi \rfloor = a_n - (n+\ell)\phi + \{(n+\ell)\phi\} = \ell(1-\phi) + 1 + D_n + \{(n+\ell)\phi\},$$

which is a bounded function. The form of b_n comes again from the fact that $b_n = a_n + n + \ell + 1$. □

The following statement is an immediate consequence of the previous corollary. However, we provide a proof because we develop arguments (about density) different from those used earlier. This result is in fact weaker than the previous one: the asymptotic behavior described below is that $a_n - n\phi = o(n)$. However, it does not yield a bounded discrepancy $a_n - n\phi = O(1)$ as in Corollary 19: the points (a_n, b_n) remain at bounded distance from a line of slope ϕ.

Corollary 20. *For the sequences defined by (5), the following limits exist:*

$$\lim_{n\to\infty}\frac{a_n}{n}=\phi \quad \textit{and} \quad \lim_{n\to\infty}\frac{b_n}{n}=\phi^2.$$

Proof. Let

$$\pi_A(x):=\#\{n\mid a_n<x\} \quad \text{and} \quad \pi_B(x):=\#\{n\mid b_n<x\}.$$

Since $a_0<\cdots<a_{n-1}<a_n$, we have $\pi_A(a_n)=n$. Similarly, $\pi_B(b_n)=n$. The two sequences make a partition of $\mathbb{N}_{>\ell}$, so that

$$\pi_A(x)+\pi_B(x)=x-\ell-1 \tag{8}$$

for every integer $x>\ell+1$. Particularly, $\pi_A(b_n)+\pi_B(b_n)=b_n-\ell-1$. Using $\pi_B(b_n)=n$, we get[2]

$$\pi_A(b_n)=b_n-n-\ell-1=a_n.$$

Consider

$$\alpha:=\limsup_{n\to\infty}\frac{a_n}{n} \quad \text{and} \quad \beta:=\liminf_{n\to\infty}\frac{a_n}{n}.$$

From Lemma 18, we have $2\geq\alpha\geq\beta\geq 1$. Our aim is to show that $\alpha=\beta$.

The intervals $[a_n,a_{n+1})$ make a partition of $\mathbb{N}_{>\ell}$. If $x\in[a_n,a_{n+1})$, then $\pi_A(x)\in\{n,n+1\}$ and

$$\frac{n+1}{a_{n+1}}\cdot\underbrace{\frac{n}{n+1}}_{\to 1}=\frac{n}{a_{n+1}}<\frac{\pi_A(x)}{x}\leq\frac{n+1}{a_n}=\frac{n}{a_n}\cdot\underbrace{\frac{n+1}{n}}_{\to 1}.$$

From this, we deduce that

$$\limsup_{x\to\infty}\frac{\pi_A(x)}{x}=\limsup_{n\to\infty}\frac{n}{a_n}=\frac{1}{\liminf_{n\to\infty}\frac{a_n}{n}}=\frac{1}{\beta}.$$

Similarly, the intervals $[b_n,b_{n+1})$ make a partition of $\mathbb{N}_{>2\ell+1}$. If $x\in[b_n,b_{n+1})$, then

$$\frac{a_n}{b_n}\cdot\underbrace{\frac{b_n}{b_{n+1}}}_{\to 1}=\frac{\pi_A(b_n)}{b_{n+1}}\leq\frac{\pi_A(x)}{x}\leq\frac{\pi_A(b_{n+1})}{b_n}=\frac{a_{n+1}}{b_{n+1}}\cdot\underbrace{\frac{b_{n+1}}{b_n}}_{\to 1}.$$

So

$$\limsup_{x\to\infty}\frac{\pi_A(x)}{x}=\limsup_{n\to\infty}\frac{a_n}{b_n}.$$

Let $g(x)=\frac{x}{x+1}$. First observe that

$$\frac{a_n}{b_n}=\frac{\frac{a_n}{n}}{\frac{a_n}{n}+1+\frac{\ell+1}{n}}.$$

[2]With the notation of the proof of Corollary 19, $S_n=\pi_B(a_n)$.

Second,

$$\left|\frac{a_n}{b_n} - g\left(\frac{a_n}{n}\right)\right| = \left|\frac{\frac{a_n}{n}}{\frac{a_n}{n}+1+\frac{\ell+1}{n}} - g\left(\frac{a_n}{n}\right)\right| = \left|\frac{\frac{a_n}{n}\left(-\frac{\ell+1}{n}\right)}{\left(\frac{a_n}{n}+1+\frac{\ell+1}{n}\right)\left(\frac{a_n}{n}+1\right)}\right| \to 0.$$

The function g is continuous and strictly increasing on $[0,+\infty)$. Hence,

$$\frac{1}{\beta} = \limsup_{x\to\infty}\frac{\pi_A(x)}{x} = \limsup_{n\to\infty}\frac{a_n}{b_n} = \limsup_{n\to\infty} g\left(\frac{a_n}{n}\right) = g\left(\limsup_{n\to\infty}\frac{a_n}{n}\right) = g(\alpha) = \frac{\alpha}{\alpha+1}.$$

Proceeding in the same way with $\liminf$, we get

$$\frac{1}{\alpha} = \frac{\beta}{\beta+1}.$$

Hence, $\alpha+1 = \beta\alpha = \beta+1$ and therefore, $\alpha=\beta$. This means that the sequence $(a_n/n)_{n\geq 1}$ is converging to α. Note that

$$\frac{b_n}{n} = \frac{a_n}{n} + 1 + \frac{\ell+1}{n} \to \alpha+1, \quad \text{if } n\to\infty.$$

The proof also shows that $(\pi_A(n)/n)_{n\geq 1}$ converges to $1/\alpha$. Proceeding as before, the intervals $[b_n, b_{n+1})$ make a partition of $\mathbb{N}_{>2\ell+1}$. If $x\in[b_n,b_{n+1})$, then $\pi_B(x)\in\{n,n+1\}$ and

$$\frac{n+1}{b_{n+1}}\cdot\underbrace{\frac{n}{n+1}}_{\to 1} = \frac{n}{b_{n+1}} < \frac{\pi_B(x)}{x} \leq \frac{n+1}{b_n} = \frac{n}{b_n}\cdot\underbrace{\frac{n+1}{n}}_{\to 1}.$$

From this, we deduce that

$$\limsup_{x\to\infty}\frac{\pi_B(x)}{x} = \limsup_{n\to\infty}\frac{n}{b_n} = \frac{1}{\liminf_{n\to\infty}\frac{b_n}{n}} = \frac{1}{\alpha+1}.$$

Since

$$\frac{\pi_A(x)+\pi_B(x)}{x} \to 1 \text{ (from (8))} \quad \text{and} \quad \frac{\pi_A(x)}{x}\to\frac{1}{\alpha},$$

it follows that $\frac{\pi_B(x)}{x}$ is converging to $\frac{1}{\alpha+1}$ as $x\to\infty$. We thus get

$$1 = \frac{1}{\alpha} + \frac{1}{\alpha+1}.$$

Since $1\leq\alpha\leq 2$, we obtain that the limit α is the golden ratio $\phi=(1+\sqrt{5})/2$. □

Remark 21. One may also ask whether the sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ studied here are complementary Beatty sequences. Boshernitzan and Fraenkel [1] showed that the prefix

$P = (c_n)_{1\le n\le L}$ of length L of a sequence $(c_n)_{n\ge 1}$ is the form $(\lfloor n\alpha+\beta\rfloor)_{n\ge 1}$ for some reals α, β, if and only if $\underline{d}(P) < \overline{d}(P)$, where

$$\underline{d}(P) = \max_{1\le i<k\le L} \frac{c_k - c_{k-i} - 1}{i} \quad \text{and} \quad \overline{d}(P) = \max_{1\le i<k\le L} \frac{c_k - c_{k-i} + 1}{i}.$$

For $\ell \ge 2$, the prefix of length L of the corresponding sequence $(a_n)_{n\ge 0}$ is such that

$$\underline{d}(P) = \frac{2\ell+1}{\ell+1} > \overline{d}(P) = \frac{\ell+1}{\ell}.$$

Hence, $(a_n)_{n\ge 0}$ is not a Beatty sequence.

For the game K^1, the situation is different; we have (with our convention where $(a_0, b_0) = (2, 4)$)

$$a_n = \lfloor (n+1)\phi - 1\rfloor \quad \text{and} \quad b_n = \lfloor (n+1)\phi^2 - 1\rfloor.$$

For more on complementary sequences, we refer the reader to [8, 21].

5 A morphic characterization for $K^1, K^2, K^3, \ldots$

The *morphic characterization*, as introduced by Duchêne and Rigo [7], is given by the Fibonacci word. If we start indexing letters of the Fibonacci word $\mathbf{f}$ with 1, the indices of the letters a (resp., b) in $\mathbf{f}$ correspond to the sequence $(a_n)_{n\ge 0}$ (resp., $(b_n)_{n\ge 0}$)

n	1	2	3	4	5	6	7	8	9	10	$\cdots$
$\mathbf{f}$	a	b	a	a	b	a	b	a	a	b	$\cdots$

We say that $\mathbf{f}$ *codes* the $\mathcal{P}$-positions: the indices of the nth letter a and the nth letter b in $\mathbf{f}$ give (a_{n-1}, b_{n-1}), for $n \ge 1$. The reader may notice that the first a and b occur in positions $(1, 2)$, the second a and b in positions $(3, 5)$, and so on. For more on morphic characterization, we refer to the work of Duchêne et al. [5, Sec. 3.2], where the Wythoff game is coined as the "Fibonacci game", showing the interplay between combinatorial games and combinatorics on words.

In this section, the various morphisms f_ℓ and g_ℓ have been obtained with the heuristic of Section 2.2. We also use the commands `pposK1_an`, `pposK1_bn`, `pposK2_an` and `pposK2_bn` introduced at the beginning of Section 4.

Proposition 22. *The non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\ge 0}$ of K^1 are coded by the morphic word $g_1(f_1^\omega(0))$, where f_1 is a φ-morphism over a 5-letter alphabet and g_1 is a coding defined by*

$$f_1 : 0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 34, 3 \mapsto 31, 4 \mapsto 2,$$
$$g_1 : 0, 1, 3 \mapsto a, \quad 2, 4 \mapsto b,$$

and the indexing of the letters starts with 2.

n	2	3	4	5	6	7	8	9	10	11	$\cdots$
$f_1^\omega(0)$	0	1	2	3	4	3	1	2	3	1	$\cdots$
$g_1(f_1^\omega(0))$	a	a	b	a	b	a	a	b	a	a	$\cdots$

Proof. We turn f_1 and g_1 into an automaton stored in `Word Automata Library`, where outputs are 1 and 2 instead of a and b. Let $\mathbf{w}$ be the corresponding infinite word. We have to check that for all a_n (resp., b_n), the corresponding symbol in $\mathbf{w}$ is 1 (resp., 2). Since `Walnut` is indexing infinite words from 0, we must have

$$\forall n \geq 3, \quad \mathbf{w}_{a_n-2} = 1 \text{ and } \mathbf{w}_{b_n-2} = 2.$$

That is verified thanks to

```
eval matcha "?msd_fib  An (Ex (($pposK1_an(x,n)) => (F1G1[x-2]=1)))":
eval matchb "?msd_fib  An (Ex (($pposK1_bn(x,n)) => (F1G1[x-2]=2)))":
```

□

Proposition 23. *The non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\geq 0}$ of K^2 are coded by the morphic word $g_2(f_2^\omega(0))$, where f_2 is a φ-morphism over a 16-letter alphabet and g_2 is a coding defined by*

$$\begin{aligned} f_2 &: 0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 34, 3 \mapsto 56, 4 \mapsto 7, 5 \mapsto 89, 6 \mapsto (10), 7 \mapsto (11)(12), 8 \mapsto (10)(13), \\ &\quad 9 \mapsto (14), 10 \mapsto (10)(13), 11 \mapsto 56, 12 \mapsto (15), 13 \mapsto 5, 14 \mapsto 89, 15 \mapsto (11)(12), \\ g_2 &: 0, 1, 2, 4, 6, 8, 9, 11, 12, 13, 14, 15 \mapsto a, \quad 3, 5, 7, 10 \mapsto b, \end{aligned}$$

and the indexing of the letters starts with 3.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\cdots$
$f_2^\omega(0)$	0	1	2	3	4	5	6	7	8	9	10	11	12	10	13	14	10	13	$\cdots$
$g_2(f_2^\omega(0))$	a	a	a	b	a	b	a	b	a	a	b	a	a	b	a	a	b	a	$\cdots$

Proof. Same proof as the previous one. We turn f_2 and g_2 into an automaton and check

```
eval matcha "?msd_fib  An (Ex (($pposK2_an(x,n)) => (F2G2[x-3]=1)))":
eval matchb "?msd_fib  An (Ex (($pposK2_bn(x,n)) => (F2G2[x-3]=2)))":
```

□

One proceed in the same way for the next value of ℓ.

Proposition 24. *The non-terminal $\mathcal{P}$-positions $(a_n, b_n)_{n\geq 0}$ of K^3 are coded by the morphic word $g_3(f_3^\omega(0))$, where f_3 is a φ-morphism over a 22-letter alphabet and g_3 is a coding defined by*

$$\begin{aligned} f_3 : {} & 0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 34, 3 \mapsto 56, 4 \mapsto 7, 5 \mapsto 89, 6 \mapsto (10), 7 \mapsto (11)(12), 8 \mapsto (13)(12), \\ & 9 \mapsto (14), 10 \mapsto (15)(16), 11 \mapsto (14)(17), 12 \mapsto (18), 13 \mapsto (14)(17), 14 \mapsto (19)(12), \\ & 15 \mapsto (18)(20), 16 \mapsto (21), 17 \mapsto (18), 18 \mapsto (13)(12), 19 \mapsto (19)(12), 20 \mapsto (14), 21 \mapsto (15)(16), \\ g_3 : {} & 0, 1, 2, 3, 5, 7, 9, 11, 12, 14, 16, 18, 21 \mapsto a, \quad 4, 6, 8, 10, 13, 15, 17, 19, 20 \mapsto b, \end{aligned}$$

and the indexing of the letters starts with 4.

Remark 25. When applying the heuristic, we had to consider 4-types (resp., 5-types) to get f_2 and g_2 (resp., f_3 and g_3) right.

6 Blocking Wythoff

Larsson characterized [20, Thm. 1.1] the set of $\mathcal{P}$-positions of W^2, a variant of the Wythoff game where one option may be blocked; see Figure 6. We provide an automatic proof of this result.

Theorem 26. *The set of $\mathcal{P}$-positions of W^2 is given by*

$$\mathcal{R}_2 = \{(0,0)\} \cup \{\{n, 2n+1\} \mid n \geq 0\} \cup \{\{2\lfloor n\phi \rfloor + 2, 2\lfloor n\phi^2 \rfloor + 2\} \mid n \geq 0\}.$$

Definition 27. It is convenient to define the options of a position (for Wythoff's moves) by

```
def optW "?msd_fib (c=a & d<b) | (c<a & d=b) | (a+d=c+b & c<a)":
```

This means that (a, b, c, d) belongs to the set if and only if there is a Wythoff move from the position (a, b) to (c, d), i.e., (c, d) is an option of (a, b).

Proof. The set $\mathcal{R}_2$ can be defined by

```
def pposW2_asym "?msd_fib En,a,b ((x=0 & y=0) | (x=n & y=2*n+1) |
                ($phin(n,a) & $phi2n(n,b) & x=2*a+2 & y=2*b+2))":
def pposW2 "?msd_fib $pposW2_asym(a,b) | $pposW2_asym(b,a)":
```

requiring a 73-state automaton.

We use Proposition 11 again. Clearly, for each $\mathcal{P}$-position (p, q), at most one of its options (r, s) is a $\mathcal{P}$-position (if this is the case, the previous player will forbid this option). The following formula expresses that, if there is indeed such a $\mathcal{P}$-position (r, s), then for any other option (a, b), if it is also a $\mathcal{P}$-position, then it must coincide with (r, s).

```
eval stable "?msd_fib Ap,q ($pposW2(p,q) =>
     (Er,s ($pposW2(r,s) & $optW(p,q,r,s)) =>
     (Aa,b ($optW(p,q,a,b) & $pposW2(a,b)) => (a=r&b=s))))":
```

Intermediate computations require 6452 states and the result is TRUE.

Similarly, for each $\mathcal{N}$-position (p,q), there are at least two $\mathcal{P}$-positions $(r,s) \neq (t,u)$ in its set of options; so that the previous player cannot forbid both of these. This is expressed as

```
eval absorbing "?msd_fib Ap,q (~$pposW2(p,q) =>
                Er,s,t,u ($optW(p,q,r,s) & $optW(p,q,t,u)
                & (r!=t|s!=u) & $pposW2(r,s) & $pposW2(t,u)))":
```

Intermediate computations require 34813 states, and it returns TRUE. □

We also provide an automatic proof of the following result by Larsson [20].

Theorem 28. *The set of $\mathcal{P}$-positions of W^3, where 2 options may be blocked, is given by*

$$\mathcal{R}_3 = \{(0,0)\} \cup \{\{n, 2n+1\}, \{n, 2n+2\} \mid n \geq 0\}.$$

Proof. The set $\mathcal{R}_3$ can be defined by

```
def pposW3_asym "?msd_fib En ((x=0 & y=0) | (x=n & y=2*n+1) | (x=n&y=2*n+2))":
def pposW3 "?msd_fib $pposW3_asym(a,b) | $pposW3_asym(b,a)":
```

requiring 28 states.

The strategy is the same as for W^2 (using Proposition 11). For each $\mathcal{P}$-position (p,q), at most two of its options (r,s) and (t,u) are $\mathcal{P}$-positions (here, the previous player may forbid two moves). The following formula expresses that, if there is indeed such $\mathcal{P}$-positions, then for any other option (a,b), if it is also a $\mathcal{P}$-position, then it must coincide with (r,s) or (t,u).

```
eval stableW3 "?msd_fib Ap,q ($pposW3(p,q) =>
      (Er,s,t,u ($pposW3(r,s) & $pposW3(t,u) & $optW(p,q,r,s) & $optW(p,q,t,u)
      & (r!=t|s!=u)) =>
      (Aa,b ($optW(p,q,a,b) & $pposW3(a,b)) => ((a=r&b=s)|(a=t&r=u)))))":
```

Similarly, for each $\mathcal{N}$-position (p,q), there are at least three $\mathcal{P}$-positions in its set of options; so that the previous player cannot forbid all of these. This is expressed as

```
eval absorbW3 "?msd_fib Ap,q (~$pposW3(p,q) =>
             Er,s,t,u,v,w ($optW(p,q,r,s) & $optW(p,q,t,u) & $optW(p,q,v,w)
             & (r!=t|s!=u) & (r!=v|s!=w) & (v!=t|w!=u)
             & $pposW3(r,s) & $pposW3(t,u) & $pposW3(v,w)))":
```

Both commands evaluate to TRUE, intermediate computations requiring at most 116141 states. □

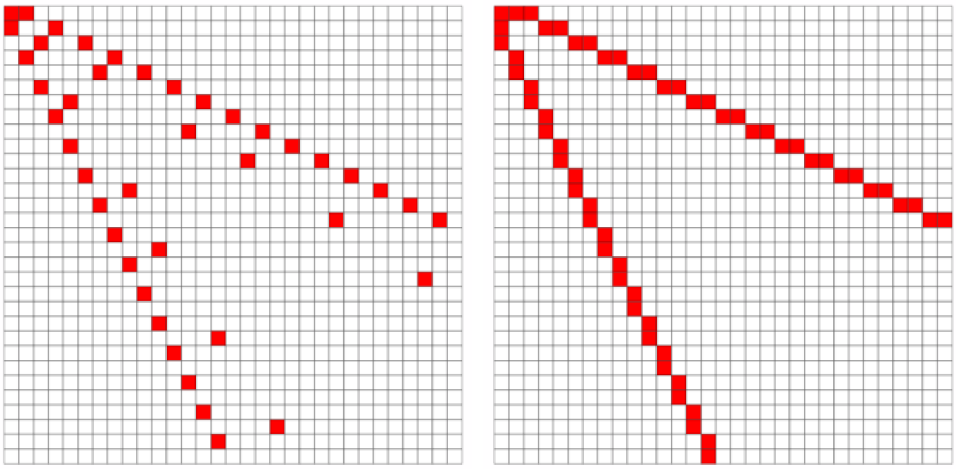

Figure 6: The $\mathcal{P}$-positions of W^2 and W^3.

7 About redundant moves

A classical question about games is the following: can we modify the rule-set in such a way that the set of $\mathcal{P}$-positions remains the same? One may add any move provided that it does not permit moving between two $\mathcal{P}$-positions, breaking the stability of the set. Therefore, we focus on the question of removing a move. With a smaller rule-set, stability still holds trivially. The question thus concerns absorption. Let us recall what a redundant move is.

Definition 29. A move of a game G is said to be *redundant* if removing it from the rule-set does not change the set of $\mathcal{P}$-positions of G.

Duchêne et al. [6] showed that the classical Wythoff game has no redundant move. This result has also been verified automatically with `Walnut` in our previous work [23]. Using a similar approach, we show that the variants of the Wythoff game considered in this paper have no redundant move.

7.1 No redundant moves in K^ℓ, $\ell \leq 4$

We follow the procedure described in our previous paper [23]. For each move m, that is, for all pairs of the set

$$\bigcup_{i>0} \{(i,0),(0,i),(i,i)\},$$

one needs to check that there exists an $\mathcal{N}$-position (p,q) such that m is the only winning move leading to a $\mathcal{P}$-position. This shows that m is necessary to preserve the absorbing property, and thus the set of $\mathcal{P}$-positions.

This can be expressed in `Walnut` using the following three commands (one for each type of move). We make use of Definition 27 for the options `optW`. We illustrate this on the game K^2:

```
eval nr_1 "?msd_fib Ai (i>0 => (Ep,q ~$pposK2(p,q) & $pposK2(p-i,q)
    & (Ar,s ((r!=p-i & s!=q & $optW(p,q,r,s)) => ~$pposK2(r,s)))))":

eval nr_2 "?msd_fib Ai (i>0 => (Ep,q ~$pposK2(p,q) & $pposK2(p,q-i)
    & (Ar,s ((r!=p & s!=q-i & $optW(p,q,r,s)) => ~$pposK2(r,s)))))":

eval nr_3 "?msd_fib Ai (i>0 => (Ep,q ~$pposK2(p,q) & $pposK2(p-i,q-i)
    & (Ar,s ((r!=p-i & s!=q-i & $optW(p,q,r,s)) => ~$pposK2(r,s)))))":
```

All three commands evaluate to `TRUE`, with intermediate steps computing automata with up to 2130 states.

Theorem 30. *Let $\ell \in \{1, 2, 3, 4\}$. The game K^ℓ has no redundant move, i.e., removing any move from the rule-set changes the set of $\mathcal{P}$-positions.*

Proof. Simply repeat the above three commands, replacing the predicate `pposK2` conveniently. The longest computation is for `nr_3` and K^4, which requires up to 5140 states. □

7.2 No redundant moves in W^2 and W^3

Let us first consider the game W^2. Here, we have to take into account the blocking maneuver: since the opponent can forbid one option, a move m is non-redundant if there exists an $\mathcal{N}$-position (p, q) with exactly two options in $\mathcal{P}$ that can be reached using two moves m and m', respectively.

Indeed, if m is withdrawn from the rule-set, then m' will be the only option remaining to go from (p, q) to a $\mathcal{P}$-position. Since the previous player may prevent the next one from playing along this move, the absorbing property is not satisfied, and m is thus necessary to preserve the set of $\mathcal{P}$-positions.

Again, this can be expressed in `Walnut`. Let us start with a move of the form $(i, 0)$. The first formula defines the set of pairs (i, j) such that there exists an $\mathcal{N}$-position (p, q) from which there are exactly two moves of the form $(i, 0)$ and $(j, 0)$, with $i \neq j$, leading to $\mathcal{P}$-positions. Any option (r, s) from (p, q) which is a $\mathcal{P}$-position is thus reached using one of these two. If a pair belongs to this set, then $(i, 0)$ and $(j, 0)$ are non-redundant. The second (resp., third) formula handles the situation where there exists an $\mathcal{N}$-position (p, q) from which exactly two moves of the form $(i, 0)$ and $(0, j)$ (resp., $(i, 0)$ and (j, j)) are leading to $\mathcal{P}$-positions. Here, there is no restriction on i and j because we use different types of moves.

```
def opt2W2a "?msd_fib i!=j & Ep,q (p>=i & p>=j & ~$pposW2(p,q)
                          & $pposW2(p-i,q) & $pposW2(p-j,q)
                         & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                           => ((r=p-i& s=q) | (r=p-j& s=q))))":
```

```
def opt2W2b "?msd_fib  Ep,q (p>=i & q>=j & ~$pposW2(p,q)
                           & $pposW2(p-i,q) & $pposW2(p,q-j)
                          & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                            => ((r=p-i& s=q) | (r=p& s=q-j))))":

def opt2W2c "?msd_fib  Ep,q (p>=i & p>=j & q>=j & ~$pposW2(p,q)
                           & $pposW2(p-i,q) & $pposW2(p-j,q-j)
                          & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                            => ((r=p-i& s=q) | (r=p-j& s=q-j))))":
```

Intermediate computations for each of these formulas require approximately 11000 states. Now, we evaluate

```
eval opt2W2 "?msd_fib Ai (i>0=> (Ej ($opt2W2a(i,j)|$opt2W2b(i,j)
                                                  |$opt2W2c(i,j))))":
```

which returns `TRUE`. This means that $(i, 0)$ is a non-redundant move for all $i > 0$. By symmetry, this is also the case for a move of the form $(0, i)$, with $i > 0$.

We now consider moves of the form (i, i). The strategy is similar; we just have to ensure, in the third formula, that the moves (i, i) and (j, j) are distinct with $i \neq j$.

```
def opt2W2d "?msd_fib  Ep,q (p>=i & q>=i & p>=j & ~$pposW2(p,q)
                           & $pposW2(p-i,q-i) & $pposW2(p-j,q)
                          & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                            => ((r=p-i& s=q-i) | (r=p-j& s=q))))":

def opt2W2e "?msd_fib  Ep,q (p>=i & q>=i & q>=j & ~$pposW2(p,q)
                           & $pposW2(p-i,q-i) & $pposW2(p,q-j)
                          & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                            => ((r=p-i& s=q-i) | (r=p& s=q-j))))":

def opt2W2f "?msd_fib  i!=j & Ep,q (p>=i & q>=i & ~$pposW2(p,q)
                           & $pposW2(p-i,q-i) & $pposW2(p-j,q-j)
                          & Ar,s (($optW(p,q,r,s) & $pposW2(r,s))
                            => ((r=p-i& s=q-i) | (r=p-j& s=q-j))))":
```

We then evaluate

```
eval opt2W2 "?msd_fib Ai (i>0 => (Ej ($opt2W2d(i,j)|$opt2W2e(i,j)
                                                  |$opt2W2f(i,j))))":
```

returning `TRUE`. Consequently, we have the following result.

Theorem 31. *The game W^2 (resp., W^3) has no redundant move, i.e., removing any move from the rule-set changes the set of $\mathcal{P}$-positions.*

For the sake of readability, the commands for W^3 are given in the appendix.

8 Conclusions

The deep connections existing between combinatorial games and numeration systems have long been exploited by many authors [6, 7, 9, 10, 11], particularly to determine whether a given position is a $\mathcal{P}$-position in polynomial time. The Wythoff game and many of its variants are closely related to the Fibonacci numeration system. In this article, following the line of our previous work [23], we take these ideas one step further, providing more evidence. Using a decidable theory implemented in the freely available software `Walnut`, we have obtained many results with little effort! These examples are particularly well-suited, because the Fibonacci numeration system admits a simple automaton recognizing addition. This ensures that the automata produced during intermediate formula-reduction steps hopefully remain of manageable size.

The starting point of this article was the rather unexpected connection between Hofstadter's G-sequence A005206 and the combinatorial game K^2 introduced by Komak et al. [19]. This led us to generalize K^2 to a family of games K^ℓ. It would therefore be interesting to relate this family to the generalized Hofstadter sequences studied by Letouzey [22], for which connections with numeration systems are of interest A005374, A005375, A005376, A100721. This could also be linked to Narayana representations, as discussed by Shallit [27]. In particular, can one define an analogue of the function g from Proposition 4 and Theorem 2 for other values of the parameter ℓ? We did not pursue this direction, as our aim was to obtain Fibonacci-automatic functions g_2, g_3, g_4 with Proposition 6 and Theorems 13 and 14.

From a different perspective, although there is no unique logical formula defining properties such as stability or absorption, our automatic proofs reveal a clear difference in the order of magnitude of the automata constructed by `Walnut`: for the games K^ℓ ($\ell \leq 4$), the automata have a few thousand states, whereas for the blocking games W^2 and W^3, they exceed 10^5 states. Thus, assuming a notion of minimal representation/logical formula can be defined, one could quantify the complexity of a combinatorial game by the size of the automata involved in the proof characterizing its $\mathcal{P}$-positions. Of course, this is reasonable for games having a description within the same numeration system, since the size of the automaton recognizing addition plays a role in these constructions.

9 Acknowledgments

We thank the reviewer for the careful reading of the manuscript.

10 Appendix

Here are the commands showing that W^3 has no redundant move. With the command `opt2W3_mn`, we consider a move of the first kind $(i, 0)$ and two moves of kind `m` and `n` ; the three being pairwise distinct. Moves of the second (resp. third) kind are of the form $(0, i)$

and (i,i). When at least two moves are of the same kind, we have to ensure that they are distinct.

```
def opt2W3_11 "?msd_fib i!=j & j!=k & i!=k & Ep,q (p>=i & p>=j & p>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q) & $pposW3(p-j,q) & $pposW3(p-k,q)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p-j & s=q) | (r=p-k & s=q))))":

def opt2W3_12 "?msd_fib i!=j & Ep,q (p>=i & p>=j & q>=k & ~$pposW3(p,q)
 & $pposW3(p-i,q) & $pposW3(p-j,q) & $pposW3(p,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p-j & s=q) | (r=p & s=q-k))))":

def opt2W3_22 "?msd_fib j!=k & Ep,q (p>=i & q>=j & q>=k & ~$pposW3(p,q)
 & $pposW3(p-i,q) & $pposW3(p,q-j) & $pposW3(p,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p & s=q-j) | (r=p & s=q-k))))":

def opt2W3_13 "?msd_fib i!=j & Ep,q (p>=i & p>=j & p>=k & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q) & $pposW3(p-j,q) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p-j & s=q) | (r=p-k & s=q-k))))":

def opt2W3_23 "?msd_fib Ep,q (p>=i & q>=j & p>=k & q>=k & ~$pposW3(p,q)
 & $pposW3(p-i,q) & $pposW3(p,q-j) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p & s=q-j) | (r=p-k & s=q-k))))":

def opt2W3_33 "?msd_fib j!=k & Ep,q (p>=i & p>=j & q>=j & p>=k & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q) & $pposW3(p-j,q-j) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q) | (r=p-j & s=q-j) | (r=p-k & s=q-k))))":
```

and finally, we check:

```
eval redundantW3_1 "?msd_fib Ai (i>0
 => Ej,k ($opt2W3_11(i,j,k) | $opt2W3_12(i,j,k) | $opt2W3_22(i,j,k) |
          $opt2W3_13(i,j,k) | $opt2W3_23(i,j,k) | $opt2W3_33(i,j,k)))":
```

By symmetry, we do not have to consider a first move of the second kind. But we still have to consider a first move of the third kind.

```
def opt3W3_11 "?msd_fib j!=k & Ep,q (p>=i & q>=i & p>=j & p>=k
```

```
 & ~$pposW3(p,q) & $pposW3(p-i,q-i) & $pposW3(p-j,q) & $pposW3(p-k,q)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p-j & s=q) | (r=p-k & s=q))))":

def opt3W3_12 "?msd_fib Ep,q (p>=i & q>=i & p>=j & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q-i) & $pposW3(p-j,q) & $pposW3(p,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p-j & s=q) | (r=p & s=q-k))))":

def opt3W3_22 "?msd_fib j!=k & Ep,q (p>=i & q>=i & q>=j & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q-i) & $pposW3(p,q-j) & $pposW3(p,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p & s=q-j) | (r=p & s=q-k))))":

def opt3W3_13 "?msd_fib i!=k & Ep,q (p>=i & q>=i & p>=j & p>=k & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q-i) & $pposW3(p-j,q) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p-j & s=q) | (r=p-k & s=q-k))))":

def opt3W3_23 "?msd_fib i!=k & Ep,q (p>=i & q>=i & q>=j & p>=k & q>=k
 & ~$pposW3(p,q) & $pposW3(p-i,q-i) & $pposW3(p,q-j) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p & s=q-j) | (r=p-k & s=q-k))))":

def opt3W3_33 "?msd_fib i!=j & j!=k & i!=k & Ep,q
 (p>=i & q>=i & p>=j & q>=j & p>=k & q>=k & ~$pposW3(p,q)
 & $pposW3(p-i,q-i) & $pposW3(p-j,q-j) & $pposW3(p-k,q-k)
 & Ar,s (($optW(p,q,r,s) & $pposW3(r,s))
 => ((r=p-i & s=q-i) | (r=p-j & s=q-j) | (r=p-k & s=q-k))))":

eval redundantW3_2 "?msd_fib Ai (i>0
 => Ej,k ($opt3W3_11(i,j,k) | $opt3W3_12(i,j,k) | $opt3W3_22(i,j,k) |
          $opt3W3_13(i,j,k) | $opt3W3_23(i,j,k) | $opt3W3_33(i,j,k)))":
```

References

[1] M. Boshernitzan and A. S. Fraenkel, Nonhomogeneous spectra of numbers, *Discrete Math.* **34** (1981), 325–327.

[2] V. Bruyère and G. Hansel, Bertrand numeration systems and recognizability, *Theoret. Comput. Sci.* **181** (1997), 17–43.

[3] A. Cobham, Uniform tag systems, *Math. Systems Theory* **6** (1972), 164–192.

[4] F. M. Dekking, On Hofstadter's G-Sequence, *J. Integer Sequences* **26** (2023), Article 23.9.2.

[5] E. Duchê ne, A. S. Fraenkel, V. Gurvich, N. B. Ho, C. Kimberling, and U. Larsson, Wythoff visions, *Games of No Chance* **5** (2017), 35–87.

[6] E. Duchêne, A. S. Fraenkel, R. Nowakowski, and M. Rigo, Extensions and restrictions of Wythoff's game preserving its $\mathcal{P}$-positions, *J. Combin. Theory Ser. A* **117** (2010), 545–567.

[7] E. Duchêne and M. Rigo, A morphic approach to combinatorial games: the Tribonacci case, *RAIRO Theor. Inform. Appl. (RAIRO:ITA)* **42** (2008), 375–393.

[8] A. S. Fraenkel, Complementary systems of integers, *Amer. Math. Monthly* **84** (1977), 114–115.

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

[10] A. S. Fraenkel, The use and usefulness of numeration systems, *Inform. and Comput.* **81** (1989), 46–61.

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

[12] A. S. Fraenkel and A. Peled, Harnessing the unwieldy MEX function, *Games of No Chance* **4** (2015), 77–94.

[13] D. Gault and M. Clint, "Curiouser and curiouser" said Alice. Further reflections on an interesting recursive function, *Int. J. Comput. Math.* **26** (1988), 35–43.

[14] V. Granville and J.-P. Rasson, A strange recursive relation, *J. Number Theory* **30** (1988), 238–241.

[15] D. R. Hofstadter, *Goedel, Escher, Bach: an Eternal Golden Braid*, Random House, 1980.

[16] C. Kimberling, Stolarsky interspersions, *Ars Combin.* **39** (1995), 129–138.

[17] C. Kimberling, Complementary equations, *J. Integer Sequences* **10** (2007), Article 07.1.4.

[18] C. Kimberling, Complementary equations and Wythoff sequences, *J. Integer Sequences* **11** (2008), Article 08.3.3.

[19] K. Komak, R. Miyadera, and A. Murakami, A variant of Wythoff's game defined by Hofstadter's G-sequence, arxiv preprint arXiv:2510.11767 [math.CO], 2025. Available at https://arxiv.org/abs/2510.11767.

[20] U. Larsson, Blocking Wythoff Nim, *Electron. J. Combin.* **18** (2011), P120.

[21] U. Larsson, Restrictions of m-Wythoff Nim and p-complementary Beatty sequences, *Games of No Chance* **4** (2015), 137–160.

[22] P. Letouzey, Generalized Hofstadter functions G, H and beyond: numeration systems and discrepancy, arxiv preprint arXiv:2502.12615 [cs.DM], 2025. Available at https://arxiv.org/abs/2502.12615.

[23] B. Mignoty, A. Renard, M. Rigo, and M. A. Whiteland, Automatic proofs in combinatorial game theory, *Internat. J. Game Theory* **54** (2025).

[24] H. Mousavi, Automatic theorem proving in Walnut, arxiv preprint arXiv:1603.06017 [cs.FL], 2016. Available at https://arxiv.org/abs/1603.06017.

[25] M. Rigo, From combinatorial games to shape-symmetric morphisms, In *Substitution and Tiling Dynamics: Introduction to Self-Inducing Structures*, Lecture Notes in Math., Vol. 2273, Springer, 2020, pp. 227–291.

[26] J. Shallit, *The Logical Approach To Automatic Sequences: Exploring Combinatorics on Words with Walnut*, Cambridge University Press, 2023.

[27] J. Shallit, The Narayana morphism and related words, arxiv preprint arXiv:2503.01026 [math.CO], 2025. Available at https://arxiv.org/abs/2503.01026.

[28] W. A. Wythoff, A modification of the game of Nim, *Nieuw Arch. Wiskd.* **7** (1906), 199–202.

2020 *Mathematics Subject Classification*: Primary 91A46 ; Secondary 11A67, 68Q45, 68V15.
Keywords: combinatorial game, Wythoff game, Walnut theorem-prover, first-order logic.

(Concerned with sequences A000201, A001950, A003622, A005206, A005374, A005375, A005376, A022342, and A100721.)

Received December 17 2025; revised versions received December 26 2025; December 28 2025; February 16 2026. Published in *Journal of Integer Sequences*, March 2 2026.

Return to Journal of Integer Sequences home page.