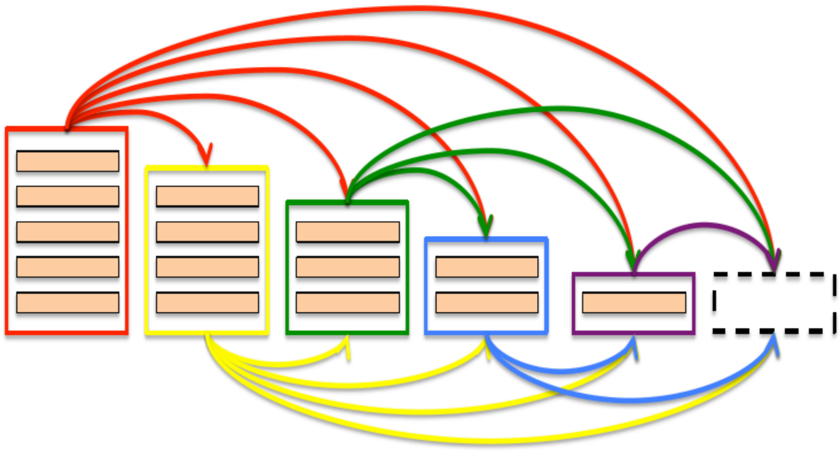
Optimal resolution of the Wythoff game and its variants.

I - The framework of zero-sum games: The Wythoff game

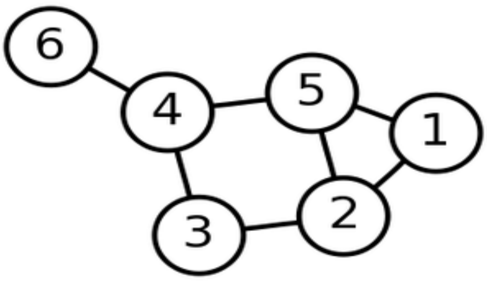
The games I'm interested in are two-player games (A and B play turn-based) where each player has all the information about the previous rounds, and where the game is over and necessarily ends with one person winning.

Example: Nim's game: You have a finite number of matches and in each round, one player will take any number of matches and whoever takes the last one wins.



(The arrows indicate the possible moves for the player)

These games are often modeled by an acyclic oriented graph where each vertex corresponds to a state of the game that we will call 'position' and so the players start from an initial position and move forward in turn, the first one to reach the final position wins.



1) Definition :

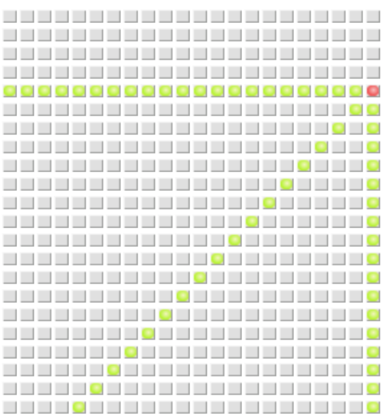
The Wythoff game is a variant of Nim's game and the fundamental object of my study. In the game of Wythoff, there are two piles of matches. At each turn, a player has three possible moves: play in the first or second stack and take any number of matches he wants, or play in both stacks simultaneously and take the same number of matches from both stacks.

2) Solve Wythoff :

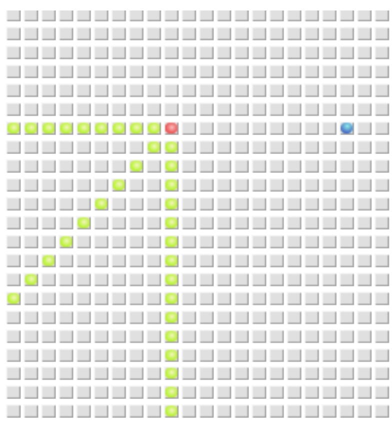
Winning positions in a game are positions that guarantee victory as long as player A (for example) follows them. This means first of all that the final position is one of these positions and that from a winning position one necessarily moves to a non-winning position, but also that there is always a way to move from a non-winning position to a winning one (while a set of winning positions may exist, there is no guarantee that it is unique, one can have two sets of winning positions).

In Nim’s game, player A is in position 5 if there are 5 matches left in the game and it is player B's turn to play.

To understand the notion of positions in Wythoff's game, I will first model it by a queen moving on a chessboard where the two axes correspond respectively to the two heaps. We can therefore move vertically, horizontally or diagonally.



Player A first plays diagonally, then player B plays horizontally.



It is now clear that every position in the Wythoff game is of the form (x,y). The best known method for solving this game is Beatty's method, which directly gives winning positions. Beatty's sequences are the two sequences of general terms $\lfloor \phi n \rfloor$ and $\lfloor \phi n+n \rfloor$ and the set of pairs of the type $(\lfloor \phi n \rfloor, \lfloor \phi n \rfloor + n)$ is indeed a set of winning positions.

Theorem :

If $1/\alpha + 1/\beta = 1$ (with α and β irrational) then : $A_i = \lfloor \alpha i \rfloor$ and $B_i = \lfloor \beta i \rfloor$ partition N.

This solving method is efficient because one can directly access the winning positions of a Wythoff game of size n x m with complexity in O(n x m), however for n and m big enough, the calculation of the integer part of ϕn and $\phi n+n$ can become very expensive. More generally to solve this kind of game, we use the Grundy function.

3) Resolution with the Grundy function :

The Grundy function (G) is recursively defined as a function of all positions to $\mathbb{R}$. G vanishes for the final position (i.e. G(0,0) = 0) and for any (x,y) position: G(x,y) = mex(J) where J is the set of positions (a,b) with (a,b) a position reachable from (x,y) . (a,b) is therefore necessarily of the type (x-k, y), (x, y-k) or (x-k, y-k), and

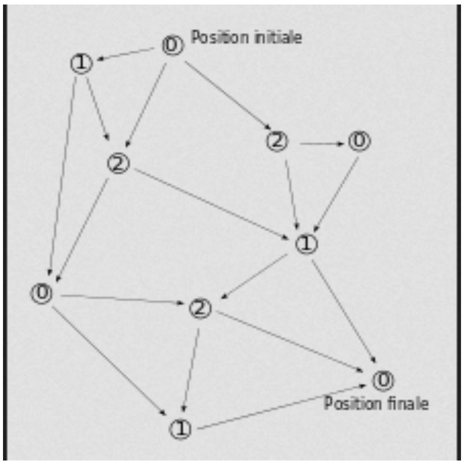
$$\operatorname{mex}(I) = \min(\mathbb{N} \setminus I).$$

Theorem :

The set of positrons that have a null Grundy is a set of winning positions. (Which is easily shown using the definition of Grundy and winning positions as I announced them)

For example, if we are in a non-zero Grundy position, if 0 is not part of the Grundy of the positions reachable from it, then 0 is the smallest integer that is not in the set of Grundy of these positions and therefore the Grundy of our position is zero, which is absurd. So from a non-zero Grundy position, we can always go to a winning position.

It is clear from this example that the positions of Grundy 0 are winning positions, which is consistent with its definition.



We can therefore recursively calculate the Grundy of all positions in a Wythoff game.

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

We notice that the 0's form about two straight lines. The slopes of these two lines tend respectively towards the golden section and the opposite. This corresponds well to the sequences of Beatty.

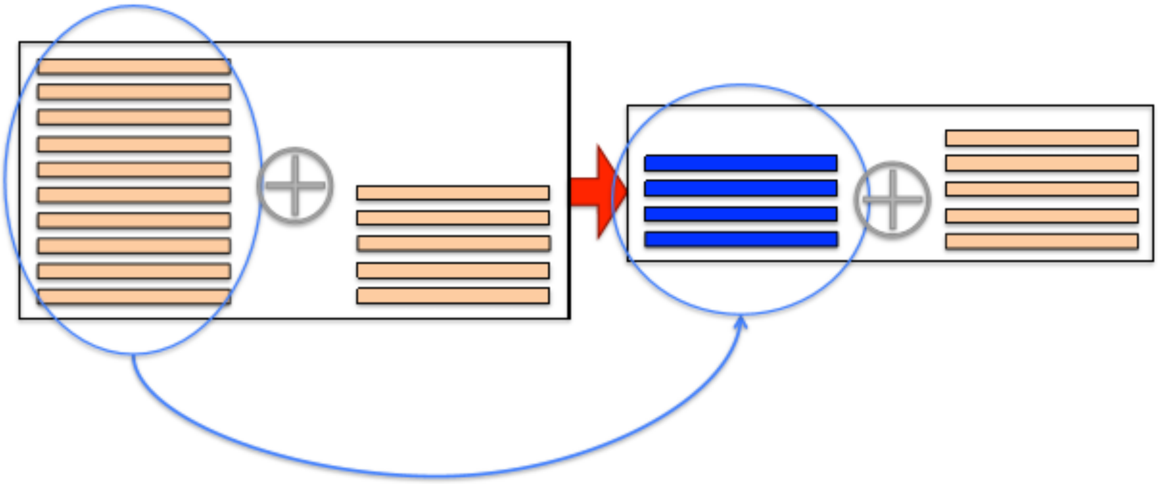
$$\lfloor \phi n \rfloor / n \sim \phi \quad \text{and } (\lfloor \phi n \rfloor + n)/n \sim \phi + 1$$

Programming the Grundy function can be difficult and requires a lot of preliminary results on the function itself. On the other hand, this function is still useful for the resolution of several variants.

4) Wythoff game ⊕ N Nim games :

A sum of games is a game where at each round, the player chooses a game and plays in it. The goal is to be the one who plays at the end.

For example, a turn in a sum of two Nim games can be represented like this:



Sprague-Grundy's theorem states how to calculate the Grundy number of any mixed position (*x*, *y) of a* sum of two sets. The Grundy numbers of the *x* and *y* positions are decomposed into binary, and the two binary numbers are summed without taking into account the holdbacks. This sum is called Nim's sum and has been rigorously defined by Conway, it is noted ⊕. The result obtained is the Grundy number of the position (*x*, *y*). This result is generalized to several games. So we can easily access the number of Grundy of a position in our game. Knowing that in a game of Nim, the Grundy of a position p is exactly p, the Grundy of any position in our game will be written: G(x,y)⊕X1⊕... ⊕Xn.

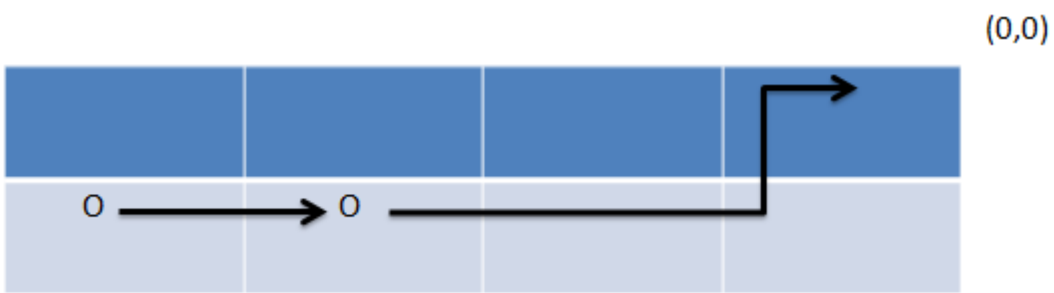
So if for example G(x,y) is greater than the sum S = X1 ⊕... ⊕Xn we know that you have to play in the Wythoff game to lower G in order to have a null Grundy and thus reach a winning position (and vice versa). Indeed, this game is studied by my partners (Jules Samaran and Yacob Ozdalkiran). And they managed to write a python code that gives at each time where to play and how many matches to take to get to a winning position using of course the Grundy function. All in all, using the Grundy function is useful for solving this kind of game.

Does this remain true when the game becomes more complicated?

II - Generalization of the Wythoff :

Let c be a strictly positive natural number. A generalized version of the Wythoff game can be defined as follows: We always have the chessboard and the queen (i.e. the two piles of matches). Each player has the right to take any number of matches from one of the two stacks (move as he wants vertically or horizontally) but this time, when player A or B decides to play in both stacks at the same time, he may take k matches from the first stack and h from the second one as long as **l k-h l < c.** This trivially means that the player can take the same number of matches from both heaps (i.e. play on the diagonal: k = h), so this is a generalization of the Wythoff game, which can be obtained by taking c = 1.

Example: Let's place ourselves in the case c = 2 and consider this figure where player A is in position (1,1).



If player B advances to square (2,1), then player A can win by advancing two squares horizontally and one vertically (2-1 = 1 < 2).

1) Generalized game resolution :

Even if it is a more complicated game, as in the Wythoff game (c =1), we have direct access to winning positions. These are moreover given by sequences that generalize the Beatty sequences. To find them, just consider the polynomial $X^2+(c-2)X-c$. By naming the two roots of this polynomial α and β (α< β), we immediately have that β= α+c and that 1/α +1/β = 1. So according to the theorem I mentioned in Part I, the two suites $\mathbf{A_i = \lfloor \alpha i \rfloor}$ **and** $\mathbf{B_i = \lfloor \alpha i \rfloor + i}$ partition **N.** Positions of the type **(A_i,B_i) are** then winning positions.

Proof :

It is easy to show that a move from a position of type ($\lfloor \alpha n \rfloor$, $\lfloor \alpha n \rfloor$+n) necessarily leads to a position of type (a, b) where (a, b) is not written as ($\lfloor \alpha m \rfloor$, $\lfloor \alpha m \rfloor$+m) (where m is a positive integer).

Now assume that Player A is in a position (a, b) (which does not put itself in the form ($\lfloor \alpha m \rfloor$, $\lfloor \alpha m \rfloor$+m)) and show that he can always go to a position ($\lfloor \alpha p \rfloor$, $\lfloor \alpha p \rfloor$+p) where p is a positive integer. Let's assume p such that: b-a = cp + r where r is an integer in the interval [0, c-1] (this is the Euclidean division of b-a by c). If a> $\lfloor \alpha p \rfloor$, removing a-$\lfloor \alpha p \rfloor$ matches from the first heap and a-$\lfloor \alpha p \rfloor$-r from the second heap (which is possible since r<c), we find ourselves in the position ($\lfloor \alpha p \rfloor$, $\lfloor \alpha p \rfloor$+cp), and if a< $\lfloor \alpha p \rfloor$, then either there is k in N such that **a** = $\lfloor \alpha k \rfloor$, in this case we take in the second heap: b-$\lfloor \alpha k \rfloor$-ck-r (i.e. c(p-k) which is positive since k<p), so we arrive in the position ($\lfloor \alpha k \rfloor$, $\lfloor \alpha k \rfloor$+ck), or there exists k in N **such** that a =$\lfloor \alpha k \rfloor$+k, in this case we can advance to the position ($\lfloor \alpha k \rfloor$+ck, $\lfloor \alpha k \rfloor$).

We can then write an algorithm that gives a list containing winning positions of this game for a size n x m of the chessboard. This algorithm will have a complexity in O ('the number of winning positions we want') but we will still have the problem of calculating the integer part.

Note that for this game, the use of the Grundy function is not optimal. Since for c large enough ($c \geq 3$), the number of positions reachable from a fixed position (x, y) becomes very large.

2) Search for an optimal method :

Let's consider the two letters ‘a’ and ‘b’, then A the set of words that can be formed from ‘a’ and ‘b’ (including the empty set), we can define an application on A by defining the image of ‘a’ and ‘b’ by morphism and using: $\sigma(xy) = \sigma(x)\sigma(y)$ where x and y are words in ‘a’ and ‘b'. We have thus defined a morphism on A. In this part I am satisfied with a particular type of morphism and I am interested in the iterates of σ in ‘a’. Let's take for example the Fibonacci morphism, defined by $\sigma(a) = ab$ and $\sigma(b) = a$. It is easy to show by recurrence that $\sigma^{n+1}(a)$ is always a prefix of $\sigma^n(a)$, i.e. the word $\sigma^{n+1}(a)$ starts with the word $\sigma^n(a)$. The $+\infty$ limit of $\sigma^n(a)$ is what is called the Fibonacci word (and which I will call in the general case: limit word), its first letters are: abaababaabaab... What is surprising is that if we name A_i and B_i the sequences of the positions of ‘a' and ‘b’ in the word respectively and we consider the couples (A_i, B_i), we get the winning positions of the Wythoff game (i.e. the Beatty positions).

Positions of a and b

in the word of

Fibonacci

a positions	b positions
1	2
3	5
4	7
6	10
8	13
9	15
$\vdots$	$\vdots$

We notice that we still have $B_i = A_i + i$ (Beatty's positions).

What is even more remarkable is that the Fibonacci morphism can be generalized into a morphism that similarly gives winning positions in the game for c any. This morphism is defined by σ(a) = (a^c)b and σ(b) = a.

For c = 2 :

a positions	b positions
1	3
2	6
4	10
5	13
7	17
8	20
9	23
⋮	⋮

We notice that this time: $B_i = A_i + 2i$. I thought then directly about a generalization $B_i = A_i + c*i$ except that it is not enough to define both sequences. On the two examples A_i is each time worth the smallest integer not found before. Indeed, $A_i = \text{mex}(A_r, B_r \text{ for } r<i)$.

So I shall announce that winning positions in this game will be in the form $A_i = \text{mex}(A_r, B_r \text{ for } r<i)$ and $B_i = A_i + c*i$ with $A_o=0$, $B_o=0$. I first verified using Python that these two sequences corresponded well to the positions of a and b in the word in question for c=3 and c=4, then it is simple to show by recurrence that these two sequences coincide respectively with the sequences $\lfloor \alpha i \rfloor$ **and** $\lfloor \alpha i \rfloor$ **+i** where **α** is the smallest root of the polynomial $X^2+(c-2)X-c$.

I managed to write a computer program that gives the n-first winning positions (for any c) by first calculating the n-th iterate of σ in a and listing the positions of a and b in that word (I know that the positions of a and b in this word will be the same as in the limit word since $\sigma^n(a)$ is a prefix of the limit word). Moreover, this program is easy to write (recursively) and has a complexity in O (n^2). So I have a not very complicated program that compensates for the calculation problem I had previously. I can thus affirm that this method is optimal to solve this game.

3) Create a game and solve it in an optimal way at the same time:

The notion of morphism allowed me to see things differently. Indeed, In the article 'Generalizing the Wythoff game', Cody Schwent generalizes the

Wythoff game by taking a word formed of a and b (obtained using the morphism defined by σ(a)=(a^2)b and σ(a)=(a^2) and decides himself to introduce new rules that generalize the rules of the Wythoff game, they then shows that the positions given by the positions of a and b in the word which is the limit of (σ^n)(a) when n tends towards +∞ are the winning positions in this new game. So this gave me a new idea, I first wondered if it was possible from any word formed by 'a' and 'b' to always choose a game in which the positions of 'a' and 'b' in the word limit are winning positions. This was of course too optimistic on my part, but by looking at the example given in the article, I managed to generalize the result for any limit word obtained from the morphism defined by σ(a)=(a^p)b and σ(a)=(a^p). So I managed to create a game in an optimal way, what I mean by this is that the way I create this game immediately gives the winning positions in an optimal way (if you try for example to solve this game by using the Grundy function or by looking for an expression of the winning positions that looks like the one in the Beatty sequences, it will be more expensive or more complicated).

So I define new rules in the game of Wythoff :

- You can choose any number of matches if you play in one of the two piles.
- We can take k matches from the first pile and h from the second pile, such that $k \leq h \leq pk$. (for p = 1, we find Wythoff's game)

Using the winning positions in the example of the article (p=2) which are written ($A_0=0$, $B_0=0$) then ($A_i = \text{mex}(A_r, B_r \text{ for } r \leq i)$, $B_i = 2A_i + i$), I have shown that in the general case : ($A_0=0$, $B_0=0$) then ($A_i = \text{mex}(A_r, B_r \text{ for } r \leq i)$, $B_i = pA_i + i$) are winning positions.

III - Conclusion :

So I managed to avoid the complexity of the Grundy function and to offer a method that does not require a huge computation from the machine. However, this method is only valid for games that generalize Wythoff. The Grundy function remains a very important tool in the resolution of accessibility games.

IV - References & inspirations :

Nivasch, G. (2005, July). More on the Sprague-Grundy function for Wythoff's game. In *Games of No Chance III, Proc. BIRS Workshop on Combinatorial Games* (pp. 377-410).

Hoggatt Jr, V. E., Bicknell-Johnson, M., & Sarsfield, R. (1979). A generalization of Wythoff's game. *Fibonacci Quart*, *17*, 198-211.

Schwent, C. Generalizing the Wythoff Game.

Duchêne, E., & Rigo, M. (2008). A morphic approach to combinatorial games: the Tribonacci case. *RAIRO-Theoretical Informatics and Applications*, *42*(2), 375-393.