HIGH SCHOOL EDITION
Journal of Student Research
Volume 10 Issue 3 (2021)

An Algorithm to Solve the Zeckendorf Game

William Lee[1] and Robert Bitler[1#]

[1]Delbarton School, Morristown, NJ, USA
[#]Advisor

ABSTRACT

Zeckendorf proved that every positive integer N can be written uniquely as the sum of non-adjacent Fibonacci numbers. This property can be used to create a two-player Zeckendorf game. A recent paper proved that player 2 has the winning strategy for all $N > 2$. However, the proof was non-constructive. In fact, the paper only provided computer code of the winning strategy of player 2 by brute force. In this paper, we present an algorithm to efficiently solve the Zeckendorf game. Specifically, we convert the game to a directed graph, prove that the graph has no cycles and only one terminal node, and construct an iterative algorithm to find all the winning strategies of player 2. We provide an example to show that the proposed algorithm works much more efficiently than a brute force approach.

Introduction

Define the Fibonacci sequence by

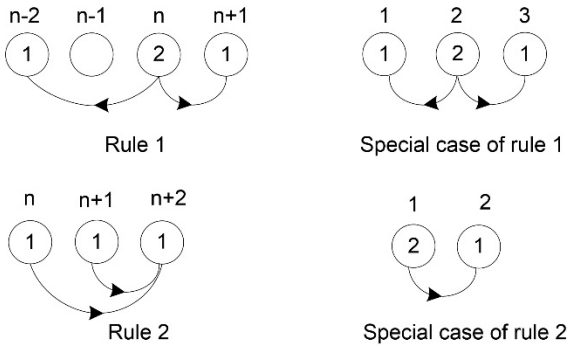
$$F_1 = 1, F_2 = 2, F_3 = 3, F_4 = 5, F_5 = 8, F_6 = 13, F_7 = 21, \ldots,$$

instead of the usual 1, 1, 2, 3, 5, …. Two properties follow from the definition of the Fibonacci sequence. First, for any number x in the Fibonacci sequence, two numbers y, z with $y < z$ exist in the Fibonacci sequence such that $y + z = 2x$. It can be easily shown that if $F_n = x$ for some n, then $F_{n-2} = y$ and $F_{n+1} = z$. Second, suppose that two numbers x, y with $x < y$ are in the Fibonacci sequence. If x and y are adjacent with each other, then $x + y$ is immediately after y in the Fibonacci sequence. If x and y are not adjacent, then $x + y$ is not in the Fibonacci sequence, because $x + y$ is smaller than any number after y in the Fibonacci sequence.

The famous theorem of Zeckendorf (Zeckendorf 1972) states that each positive integer N can be written uniquely as the sum of distinct, non-adjacent Fibonacci numbers. This property can be used to create a two-player Zeckendorf game. Specifically, consider positions, $n = 1,2, \ldots$, and place M_n coins position n where M_n is a non-negative integer. The value of a coin at position n is given by F_n. For any set of coins, the total value of the coins is equal to the sum of the values of all the individual coins, $\sum_{n=1,2,\ldots} M_n F_n$.

From the above two properties, the following two moves are considered legal to exchange coins at different positions because the total value of the coins are preserved with the moves. Figure 1 illustrates the two legal moves.

1. If there are at least two coins at position n, one can exchange the two coins with one coin at position $n - 2$ and another one at position $n + 1$. A special case is that one can exchange two coins at position 2 with one coin at position 1 and one at position 3.
2. If there are at least one coin at position n and at least one coin at position $n + 1$, one can exchange the two coins for one coin at position $n + 2$. A special case is that one can exchange two coins at position 1 with one coin at position 2.

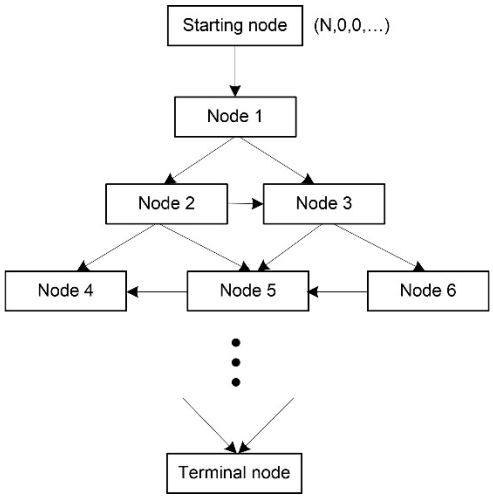


The two-player Zeckendorf game works as follows. Initially there are N coins at position 1. Player 1, referred to as A, makes a first legal move. Player 2, referred to as B, then makes a second legal move, and so on. The player who cannot find a legal move loses. The game stops when there is no legal next move.

It is easy to check that A wins the game for $N = 2$. For all $N > 2$, it was shown in (Baird-Smith et al. 2020) that B always has a winning strategy. However, the proof was non-constructive. In fact, the paper only provided computer code of the winning strategy of B by brute force. Subsequent studies, e.g., (Li 2020), have not adequately investigated constructive algorithms of the winning strategy. In this paper, we present an algorithm to efficiently solve the Zeckendorf game.

A Directed Graph Model

The state of the game at any moment can be uniquely described by the vector $(M_1, M_2, \dots)$. The evolution of the game can be described by a directed graph, where nodes α and β represent two states and an edge $E_{\alpha\to\beta}$ that connects from node α to node β represents a legal move from one state to the other. An edge corresponds to either rule 1 or 2. Because the total value of the coins is preserved and equal to N, the number of nodes in the directed graph is finite. Figure 2 illustrates the concept of the directed graph description of the game. Note that the figure shows only one terminal node. We will show in the following that there is indeed only one terminal node.



First, we prove that the directed graph consists of no cycles. Consider the following two scores,
$S_1 = \sum_{n=1,2,\dots} 2^{n-1} M_n$,

$S_2 = \sum_{n=1,2,\dots} M_n$.
Note that each time when rule 2 is applied, S_2 decrements by 1. A cycle, if exists, cannot have an edge corresponding to rule 2, and thus must go through nodes using only rule 1. However, each time when rule 1 is applied, S_1 strictly increases[1] because

$$2 \cdot 2^{n-1} < 1 \cdot 2^{n-3} + 1 \cdot 2^n.$$

Hence, it is impossible for a cycle to exist.

Without a cycle, the directed graph therefore has a finite number of terminal nodes. A terminal node α_T is where the game stops. We show next that there is only one terminal node that satisfies the following two properties.
$M_n = 0,1$ for any n.
If $M_n = 1$, for some n, then M_{n+1}.
The first property holds because if $M_n > 1$ at some node, one can apply rule 1 to make a legal move. The node cannot be a terminal node. Similarly, the second property holds because if $M_n = M_{n+1} = 1$, one can apply rule 2 to make a legal move.

Furthermore, assume that there are two distinct terminal nodes $(M_1, M_2, \dots)$ and $(M'_1, M'_2, \dots)$. Suppose that k is the largest positive integer for which M_k and M'_k differ. That is, for any $n > k$, $M_n = M'_n$. Without loss of generality, suppose that $M_k = 1$ and $M'_k = 0$. From the definition of the Fibonacci sequence,

$$F_k > F_{k-1} + F_{k-3} + F_{k-5} + \cdots.$$

Therefore, it follows from the above two properties of a terminal node that

$$\sum_{n=1,2,\dots} M_n F_n > \sum_{n=1,2,\dots} M'_n F_n,$$

which is impossible because the two terminal nodes must have the same total value. The fact that the terminal node is unique can also be established from the Zeckendorf Theorem, which states that any number has a unique representation as the sum of distinct, non-consecutive Fibonacci numbers.

In summary, we have shown that the game takes a finite number of steps evolving along the directed graph from the starting node $\alpha_0 = (N, 0,0, \dots)$ to the unique terminal node α_T. Figure 3 provides a few examples of the directed graph. The nodes can be grouped in multiple levels where $\sum_{n=1,2,\dots} M_n$ is the same for all nodes on the same level. Observe that any edge is from a node in one level to another node either on the same level or one level below.

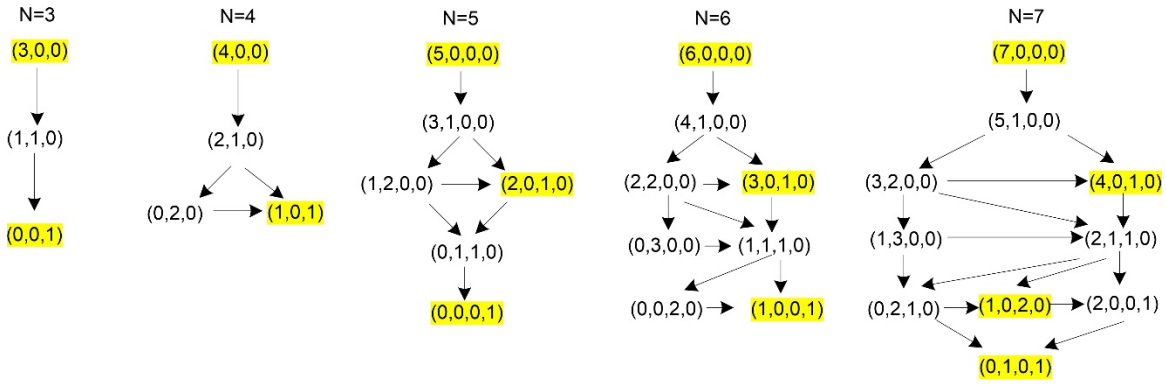


Figure 3. Directed graph examples of $N = 3,4,5,6,7$. B's selections along a winning path are marked in yellow so that one can trace the winning path from the starting node to the terminal node.

[1] S_1 remains the same when the special case of rule 1 is applied. However, it is impossible to keep on applying only the special case of rule 1 forever.

In the simple case of $N = 3$, the winning strategy is straightforward because there is only one path from α_0 to α_T. In any other cases, there are multiple paths and B may lose the game if a wrong move is selected. For example, at $N = 3$, if B selects $(1,2,0,0)$ after A's move $(3,1,0,0)$, then A can select $(2,0,1,0)$, leading to B's loss. The winning path for B is

$$(5,0,0,0) \to A\colon (3,1,0,0) \to B\colon (2,0,1,0) \to A\colon (0,1,1,0) \to B\colon (0,0,0,1)$$

Similarly, the winning paths for B at $N = 4,6,7$ are given by

$$(4,0,0) \to A\colon (2,1,0) \to B\colon (1,0,1),$$
$$(6,0,0,0) \to A\colon (4,1,0,0) \to B\colon (3,0,1,0) \to A\colon (1,1,1,0) \to B\colon (1,0,0,1),$$
$$(7,0,0,0) \to A\colon (5,1,0,0) \to B\colon (4,0,1,0) \to A\colon (2,1,1,0) \to B\colon (1,0,2,0) \to A\colon (2,0,0,1) \to B\colon (0,1,0,1).$$

B's selections along the winning paths are marked in yellow in Figure 3. Observe in the above examples that the winning path for B is unique. However, later we will show that this observation does not hold in general.

An Algorithm to Find All Winning Strategies

We present an algorithm that can find all winning strategies for B. For any N, α_T can be found with an iterative greedy algorithm. First, find k such that

$$F_k \le N < F_{k+1}.$$

Set $M_k = 1$. Let

$$N = N - F_k.$$

Repeat the above steps until $N = 0$. It can be shown from the definition of the Fibonacci sequence that N must become 0 eventually. Therefore, the above algorithm converges. For example, for $N = 16$, $\alpha_T = (0,0,1,0,0,1)$.

Define the winning paths the set of paths along the directed graph such that B is *guaranteed* to win. For a node taken by B on the winning paths, A has one or multiple legal next moves to select. For *each* of the moves, B must have a corresponding move, i.e., winning strategy, to stay on the winning paths. Denote Ω_B the set of the nodes that B takes on the winning paths. Clearly, $\alpha_T \in \Omega_B$. We next determine Ω_B iteratively from α_T.

If an edge $E_{\alpha\to\beta}$ exists in the directed graph, then node α is referred to as a parent node of β and β as a child node of α. Denote Γ_β the set of all the parent nodes of β and Δ_α the set of all the child nodes of α. For any α, we can employ the two legal moves shown in Figure 1 to find Δ_α. For any β we can employ the reverse rules, defined in Figure 4, to find Γ_β. The reverse rules are the opposite operations of the two legal moves.

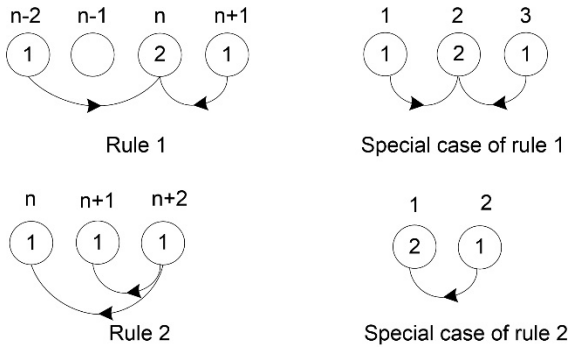


Figure 4. Illustration of the reverse rules, which are reverse from the two legal moves shown in Figure 1.

Denote Ω_A the set of the parent nodes of the nodes in Ω_B. Initially, Ω_A and Ω_B are both empty. Each iteration of the algorithm consists of the following two steps.

Step 1. Denote O_B the set of the newly added node(s) to Ω_B, i.e., update Ω_B

$$\Omega_B = \Omega_B \cup O_B$$

In the first iteration, set

$$O_B = \{\alpha_T\}.$$

Denote O_A the set of the parent nodes of all the nodes in O_B,

$$O_A = \bigcup_{\beta \in O_B} \Gamma_\beta.$$

Add O_A to Ω_A, i.e., update Ω_A

$$\Omega_A = \Omega_A \cup O_A.$$

Step 2. Reset O_B to empty. $\forall \alpha \in O_A$, check every parent node $\gamma \in \Gamma_\alpha$ to see whether *all* the child nodes of γ belong to Ω_A, i.e.,

$$\Delta_\gamma \subseteq \Omega_A.$$

If so, add γ to O_B,

$$O_B = O_B \cup \{\gamma\}.$$

After all the α in O_A have been checked, proceed to the first step of the next iteration.

The above iteration end when O_B constructed in the second step consists of only α_0, in which case the algorithm has already reached the starting node and B has a winning strategy.

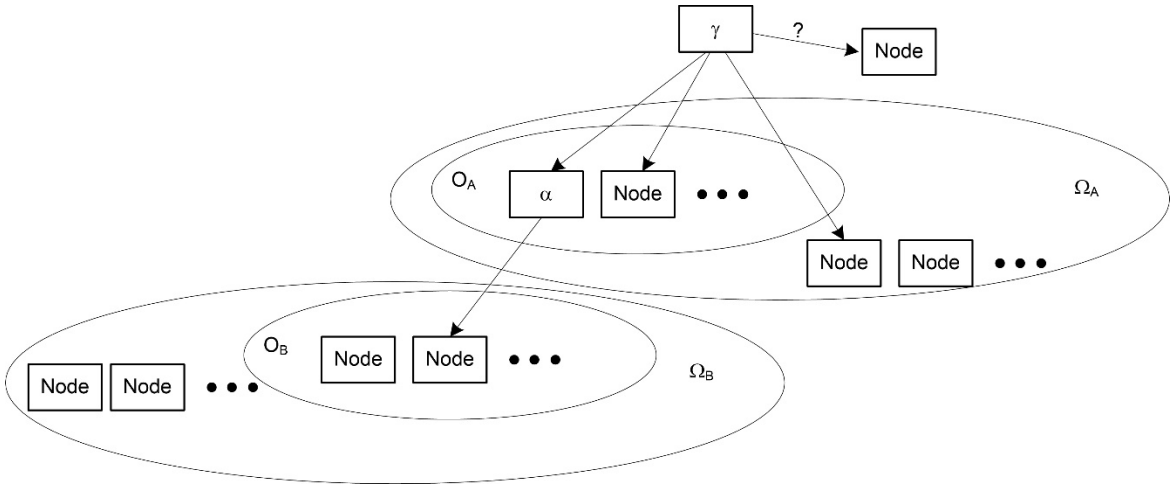


Figure 5. Illustration of one iteration of the algorithm.

Figure 5 depicts the two steps in an iteration of the algorithm. The first step ensures that if A takes any move $\alpha \in O_A$, B has *at least one* legal move to reach a node in O_B (and thus Ω_B). The second step ensures that any new node γ to be added to O_B (and thus Ω_B), *all* its child nodes belong to Ω_A. As a result, after B takes move γ, A will not be able to move out of set Ω_A. By construction, B will then have a legal move to stay in Ω_B.

As examples, Ω_B for $N = 4,5,6,7$ are marked in yellow in Figure 3. The winning strategy can be easily determined from Ω_B: for any move by A, select a legal move that leads to a node in Ω_B.

The salient features of the iterative algorithm are as follows.

- The algorithm can find all the winning strategies.
- Unlike a brute-force approach, the algorithm only examines a subset of nodes and edges of the directed graph, therefore reducing the search complexity.

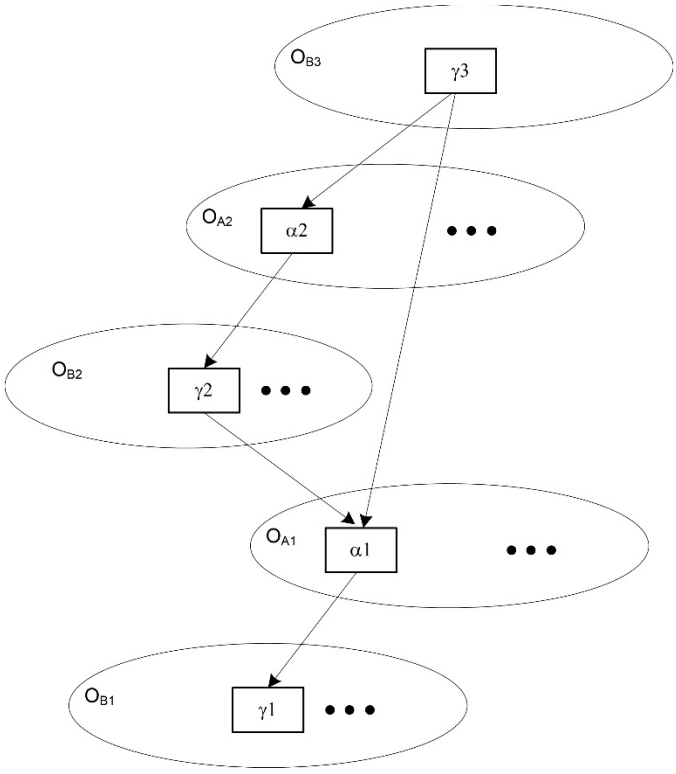


Figure 6. Illustration of a subtle point in the algorithm.

Figure 6 depicts a subtle but interesting scenario in the algorithm. In one iteration, suppose that node $\gamma_1 \in \mathrm{O}_B$, which is labeled as O_{B1}. Suppose that α_1 is γ_1's parent node and added to O_A, which is labeled as O_{A1}. Suppose that γ_2 and γ_3 are both parent nodes of α_1. γ_2 has only one child node $\alpha_1 \in O_{A1}$, and is thus added to O_B, which is labeled as O_{B2}, for the next iteration. Meanwhile, γ_3 has another child node α_2, which is not in O_{A1} or its superset Ω_A (not shown in the figure). Thus, γ_3 is thus not added to O_{B2}. In the next iteration, α_2 is a parent node of γ_2 and added to O_A, labeled as O_{A2}. γ_3 is a parent node of α_2. Now because all γ_3's child nodes are in Ω_A, a superset of O_{A1} and O_{A2}, in this iteration γ_3 is now added to O_B, which is labeled as O_{B3}.

An Example and Some Observations

We use the iterative algorithm to find the winning strategies $N = 16$. Figure 7 shows the complete directed graph including all the valid nodes and legal moves. Like Figure 3, Ω_B found by the algorithm is marked in yellow.

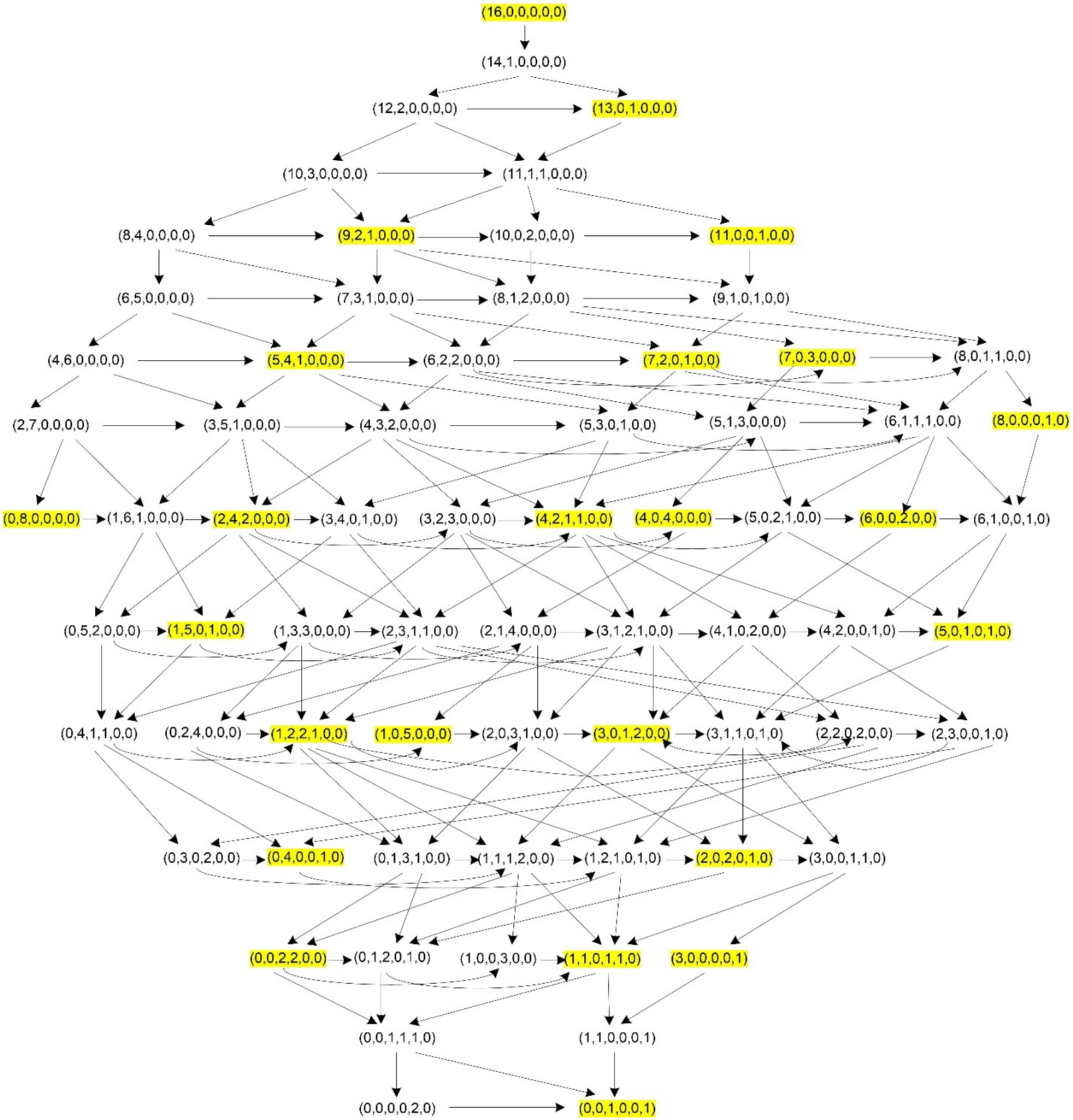


Figure 7. Complete directed graph of $N = 16$.

Figure 8 shows all the winning strategies in the form of a reduced directed graph, which are easily derived from Ω_B. Specifically, all the nodes in Ω_B, marked in yellow, remain in the reduced directed graph. Their child nodes also remain and are unmarked in Figure 8. All other nodes are discarded and marked in grey. Any edge connected with a grey node is discarded. Any edge between two unmarked nodes is discarded. The remaining edges are either from a yellow node to an unmarked node or vice versa.

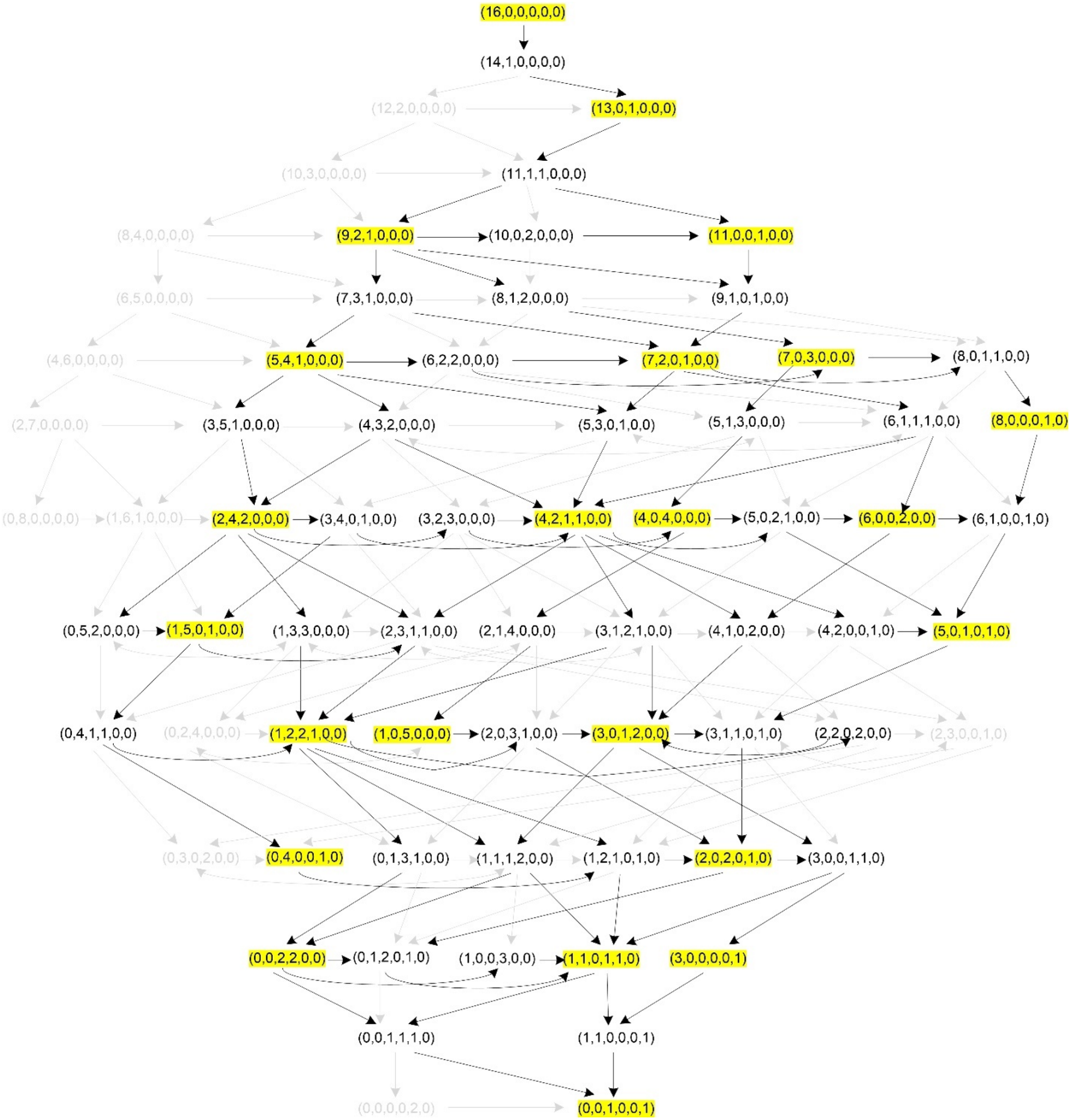


Figure 8. Diagram of all the winning strategies of $N = 16$ as a reduced directed graph.

A few observations are worth highlighting.

- Comparison between Figures 7 and 9 indicates that the reduced directed graph is significantly simpler than the complete directed graph because a number of nodes and edges are discarded.
- A winning path starts from α_0, marked in yellow. Player B always takes a yellow node, forcing player A to take an unmarked node. The game state alternates between the unmarked and yellow nodes, and finally reaches α_T, also marked in yellow, indicating that B wins the game.
- Observe that there are multiple winning paths. The reason is that a yellow node may have multiple child nodes, meaning that A can choose to take any of them. In addition, an unmarked node may have multiple yellow child nodes, meaning that B has more than one choice to respond to A's move.

Conclusion and Future Work

In this paper, we have formulated the Zeckendorf game in a directed graph model and proved a few properties. We have then presented an iterative algorithm that finds all winning strategies and shown that the proposed algorithm works much more efficiently than a brute force approach.

For the future work, we would like to extend the algorithm to other two-player games, such as the generalized Zeckendorf game (Baird-Smith 2018) and theoretically analyze the complexity of the algorithm.

Acknowledgements

I would like to thank Professor Stephen Miller from Williams College whose lecture at the Hampshire College Summer Studies in Mathematics (HCSSiM) in 2018 introduced me to this topic. Additionally, I would like to acknowledge Japheth Wood who mentored me at HCSSiM and encouraged me to pursue this research.

References

Zeckendorf, E. (1972), Représentation des nombres naturels par une somme des nombres de Fibonacci ou de nombres de Lucas, Bulletin de la Société Royale des Sciences de Liège **41**, pages 179–182.

Baird-Smith P., Epstein A., Flint K., Miller S.J. (2020). The Zeckendorf Game. In: Nathanson M. (eds) Combinatorial and Additive Number Theory III. CANT 2018. Springer Proceedings in Mathematics & Statistics, vol 297. Springer, Cham. https://doi.org/10.1007/978-3-030-31106-3_3.

Li R., Li X., Miller S.J., Mizgerd C., Sun C., Xia D., Zhou Z., (2020). Deterministic Zeckendorf Games. arXiv eprint 2006.16457, https://arxiv.org/abs/2006.16457.

Baird-Smith P., Epstein A., Flint K., Miller S.J. (2018). The Generalized Zeckendorf Games. arXiv eprint 1809.04883, https://arxiv.org/abs/1809.04883.