Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.7462

Impartial Games on Graphs: Solving Kayles via Dynamic Programming and Periodicity Properties

Marcos Felipe Medeiros-de-Souza [1] [**Fluminense Federal Institute** | ***marcos.souza@iff.edu.br***]
Felipe Eizo Hanada [2] [**Fluminense Federal University** | ***felipehanada@id.uff.br***]
Fábio Protti [2] [**Fluminense Federal University** | ***fabio@ic.uff.br***]

[1] *Instituto Federal Fluminense, Campus Santo Antônio de Pádua, Av. João Jazbick, s/n, Aeroporto, Santo Antônio de Pádua, RJ, 28470-000, Brazil*

[2] *Universidade Federal Fluminense, Institute of Computing, Av. Gal. Milton Tavares de Souza, s/n, São Domingos, Niterói, RJ, 24210-590, Brazil.*

Received: 25 January 2026 • **Accepted:** 06 July 2026 • **Published:** 31 July 2026

Abstract In this work, we consider combinatorial games in which two players alternately choose vertices from a finite graph until a winning condition is achieved. Specifically, we focus our investigation on the well-known game *Kayles*, in which the selected vertices must form an independent set, and the player who makes the last valid move wins (i.e., the player who chooses a vertex that completes a maximal independent set). Zermelo's Theorem guarantees that, in this scenario, one of the players has a winning strategy—that is, a sequence of moves that ensures a win regardless of the opponent's choices. Given a graph, the typical decision problem associated with this type of game consists of determining which player has a winning strategy. Answering this question means *solving the game*. We first consider Kayles played on caterpillars. Since caterpillars are interval graphs, an $O(n^3)$-time algorithm for solving Kayles on this graph class is already known [Bodlaender and Kratsch, 2002]. However, we investigate scenarios in which this time complexity can be reduced to $O(1)$ by extending the periodicity property presented in [Guignard and Sopena, 2009] to caterpillars. We prove that the nimber of any caterpillar is equal to the nimber of an equivalent reduced caterpillar, obtained by appropriately removing certain leaves from the original graph. By partitioning these reduced caterpillars into classes, we show that a period of 34 emerges in each investigated class, allowing the computation of nimbers to scale to graphs with a large number of vertices. We present a sufficient condition for a class of caterpillars to exhibit periodicity 34, using it to identify many periodic classes and to calculate the nimber of their caterpillars in $O(1)$ time. Furthermore, we present an $O(n^2)$-time dynamic programming algorithm for solving Kayles on powers of paths, improving upon the $O(n^4)$-time complexity given in [Bodlaender and Kratsch, 2002] for graphs with an asteroidal number of at most 2. Finally, we show how this same $O(n^2)$-time algorithm can be adapted to solve Kayles on powers of cycles, thereby reducing the $O(n^3)$-time complexity previously established in [Bodlaender and Kratsch, 2002] for circular-arc graphs.

Keywords: Caterpillars, graphs, combinatorial games, Kayles, nimbers.

1 Introduction

In this work we consider combinatorial games that satisfy the following conditions (see [Saldanha, 2008]): there are two players who make their moves alternately; the number of possible moves a player can make on their turn is finite; the game is a perfect information game, i.e. both players know the possible moves and can identify the current situation of the game, having full knowledge of the rules and possible outcomes; there is no chance/probability involved in each move; there is a criterion that determines the end of the game, which is previously agreed with the players; and finally, the end result will be either a win (for one of the players) or a draw. Under these conditions, Zermelo's Theorem is valid, which uses the following terminology in its statement: a *winning strategy* is a sequence of moves that guarantees a win, regardless of the opponent's moves; analogously, a *draw strategy* guarantees that the player will not be defeated.

Theorem 1.1 [Zermelo, 1912] Given a combinatorial game, we have only three possible outcomes: either the first player (*Alice*) has a winning strategy, or her opponent (*Bob*) has a winning strategy, or they both have a draw strategy.

An important class of combinatorial games consists of *impartial games*, which have the following additional feature: given a certain configuration of the game, either of the two players can make exactly the same moves if it were their turn to play (see [Ferguson, 2000]). In other words, the identity of the players does not change the available moves at all (see [DeVos and Kent, 2016]). For example, Chess, Checkers and Tic-Tac-Toe are not impartial, because the possible moves of one player are different from the possible moves of the other player. The well-known game *Nim* is a prominent example of an impartial game. This game consists of l piles of stones of sizes $a_1, a_2, \ldots, a_l$. To execute the move, the player must remove any number of stones greater than zero (including all of them) from a selected pile. The last player to remove stones, leaving the opponent with no possible move, is declared the winner. This variant is called *normal* (the last player to play is the winner). There is also the *misère* variant, in which the last to play is the loser, but we will not consider this variant in this work.

Note that the Nim game does not allow draws. If the possibility of a draw is eliminated, according to Zermelo's Theorem the winner will only be determined by whoever has the winning strategy. This is the case for the *Kayles* game, the subject of this paper (see [Bodlaender and Kratsch, 2002]). In the game Kayles (or *node-Kayles*), the board consists of an undirected graph $G = (V, E)$. On his/her turn, the player chooses a vertex $v \in V$, and cannot choose a previously chosen vertex or a vertex adjacent to a previously chosen vertex. The player who fails to choose a new vertex loses the game. The game can also be described as follows: when a player chooses a vertex $v \in V$, v together with all of its neighbors (the closed neighborhood $N[v]$) are removed from the graph. The first player to finish their move with an empty graph (i.e., completes a maximal independent set) wins the game. As an example, consider the board consisting of the path $P = v_1 v_2 \dots v_7$. Alice starts by choosing v_3. After the move, the graph becomes the disjoint union of $P' = v_1$ and $P'' = v_5 v_6 v_7$. Next, Bob chooses v_6, and the remaining graph is trivial. Finally, Alice chooses v_1 and wins the game. It is easy to see that Alice has a winning strategy and can start the game by choosing other vertices.

We consider the following combinatorial decision problem: given a graph G, does Alice have a winning strategy for Kayles in G? Answering this question means *solving the game*. In [Schaefer, 1978], it is proved that solving Kayles is a PSPACE-complete problem. Moreover, in [Bodlaender *et al.*, 2015] it is proved that Kayles can be solved in time $O(1.6031^n)$ for any graph with n vertices, and in time $O(1.4423^n)$ for forests. In [Bodlaender and Kratsch, 2002], the authors show how to solve Kayles in polynomial time on specific classes of graphs, describing algorithms that run, respectively, in $O(n^{k+2})$ time on graphs with asteroidal number at most k; $O(n^3)$ time on cocomparability graphs and circular arc graphs; and $O(n^{1.631})$ time for cographs. Other references on Kayles are [Sibert and Conway, 1992] and [Fleischer and Trippen, 2006].

Recently, Kobayashi [2021] studied Kayles from the perspective of fixed-parameter tractability, showing that Kayles is FPT when parameterized by the size of a minimum vertex cover $\tau(G)$ or the modular-width $\mathrm{mw}(G)$ of the input graph G, providing algorithms running in $3^{\tau(G)} n^{O(1)}$ and $1.6031^{\mathrm{mw}(G)} n^{O(1)}$ time, respectively. Moreover, the author describes a polynomial kernel with respect to the neighborhood diversity $\mathrm{nd}(G)$, reducing the instance to at most $2\,\mathrm{nd}(G)$ vertices. These are the first nontrivial FPT results for Kayles.

Each position in an impartial game is equivalent to a *nimber* (or *nim-value*), so that Alice has a winning strategy for the game if and only if the nimber of the starting position is non-zero. In [Berlekamp *et al.*, 2001] and [Guignard and Sopena, 2009], the authors come up with an interesting piece of information: Kayles, when played on paths, is *periodic*, i.e., the nimber of P_n is equal to the nimber of P_{n+34}, for all $n > 51$. This leads to a constant-time algorithm for solving Kayles on paths.

Caterpillars are interval graphs, and this directly gives an $O(n^3)$-time algorithm for solving Kayles on caterpillars [Bodlaender and Kratsch, 2002], because interval graphs are cocomparability graphs. In this work, we investigate scenarios in which we can reduce this time complexity to $O(1)$, by extending the periodicity property presented in [Guignard and Sopena, 2009] to caterpillars. We prove that the nimber of any caterpillar is equal to the nimber of an equivalent *reduced* caterpillar, obtained from the original caterpillar by appropriately removing some leaves. This result simplifies the computation of caterpillar nimbers, and allows the periodicity property of nimbers to be stated more simply. By partitioning the reduced caterpillars into classes, we show that the period 34 emerges in each class investigated, extending the computation of nimbers to graphs with a large number of vertices. We present in Theorem 5.1 a sufficient condition for a class of caterpillars to exhibit periodicity 34, using it to find many periodic classes and to calculate the nimber of caterpillars in these classes in $O(1)$ time.

We also present an $O(n^2)$-time dynamic programming algorithm for solving Kayles on powers of paths, reducing the $O(n^{k+2})$-time complexity given in [Bodlaender and Kratsch, 2002] for solving Kayles on graphs with asteroidal number at most k (powers of paths have asteroidal number at most two, yielding a previous $O(n^4)$-time algorithm).

We show how the same $O(n^2)$-time algorithm can be used for solving Kayles on powers of cycles, reducing the $O(n^3)$-time complexity given in [Bodlaender and Kratsch, 2002] for solving Kayles on circular-arc graphs (powers of cycles are contained in this class of graphs).

The remainder of this work is organized as follows. Section 2 contains preliminary concepts and results, including a discussion on the computation of the nimber of a caterpillar. Section 3 presents the nimber periodicity of paths. Section 4 defines reduced caterpillars and establishes their equivalence with caterpillars in general. Section 5 is devoted to a detailed study of nimber periodicity of special classes of caterpillars. Section 6 describes a polynomial-time dynamic programming algorithm for solving Kayles on powers of paths and cycles. Section 7 contains concluding remarks.

2 Preliminaries

The concepts presented in this section can be found in [DeVos and Kent, 2016]. A position in an impartial game is either an $\mathcal{N}$-*position* (a position in which the next player to play has a winning strategy) or a $\mathcal{P}$-*position* (a position in which the player who has just played has a winning strategy). Theorem 2.1 below characterizes the winning positions in the game Nim. In the theorem, $*x_i$ denotes a pile with x_i stones, and $*x_1 + *x_2 + \cdots + *x_l$ denotes a Nim position with l piles (or *components*). In addition, the *Nim-sum* $\oplus$ is defined as the bitwise exclusive OR of the numbers of stones in each pile (as an example, $2 \oplus 3 \oplus 4 = 010 \oplus 011 \oplus 100 = 101 = 5$).

Theorem 2.1 [Bouton, 1901] A position $*x_1 + *x_2 + \cdots + *x_l$ of Nim is a $\mathcal{P}$-position if and only if the Nim-sum of its components is zero, i.e., $x_1 \oplus x_2 \oplus \cdots \oplus x_l = 0$.

As a corollary of the above theorem, if the Nim-sum of the components is non-zero, Alice has a winning strategy; otherwise, Bob has a winning strategy. It is important to think of the Nim positions as *nimbers*, i.e., we associate with each position a nimber in the form $*a$, for a certain non-negative integer a. Thus, $\mathcal{P}$-Positions have nimber $*0$, while $\mathcal{N}$-Positions have nimber different from $*0$. The following theorem states that each Nim position is equivalent to a nimber:

Theorem 2.2 [Bouton, 1901; Sprague, 1935; Grundy, 1939] If $a_1, a_2, \cdots, a_l$ are non-negative integers and $b = a_1 \oplus a_2 \oplus \cdots \oplus a_l$, then $*a_1 + *a_2 + \cdots + *a_l \equiv *b$.

The Sprague-Grundy Theorem states that every impartial game in the normal variant is equivalent to a one-pile Nim game. To state it, we emphasize that a position in a game can be represented as the set of moves that can be made from that position, or, equivalently, as the set $\{\alpha_1, \ldots, \alpha_k\}$ of positions that can be reached from it.

Theorem 2.3 [Sprague, 1935; Grundy, 1939] Every position in an impartial game is equivalent to a nimber.

The function that associates each position in a game with its nimber is called the Sprague-Grundy function. In order to determine the equivalent nimber of a position, we need the definition of the minimum excluded value of a set. For a finite set $S = \{a_1, a_2, \cdots, a_n\}$ of non-negative integers, we define the minimum excluded value (or mex) of S to be the smallest non-negative integer b such that $b \notin S$.

Theorem 2.4 [Sprague, 1935; Grundy, 1939] Let $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ be a position in an impartial game. Suppose that $\alpha_i \equiv *a_i$, $1 \leq i \leq k$. Then, $\alpha \equiv *b$ where $b = \text{mex}\{a_1, \ldots, a_k\}$.

To make things easier, we can also write $*b = \text{mex}\{*a_1, \ldots, *a_k\}$ in the statement above. The set of nimbers $\{*a_1, \ldots, *a_k\}$ will be called the *mex set* of position α. 6103-JBCS-Computer vision to attendance recognition: a bibliometric analysis-working (Copy)

Table 1 shows the calculation of the Sprague-Grundy function in Kayles when the board is a path P_n ($n \leq 5$). The empty graph $\varnothing = P_0$ is a $\mathcal{P}$-position, since Alice has no move to make; thus, its nimber is $*0$. The graph P_1 is characterized by the set of positions $\{\varnothing\}$, since $\varnothing$ is the only position following a move on P_1; thus, the nimber of P_1 is $\text{mex}\{*0\} = *1$. The remaining calculations follow the same method. Note that the nimber of P_4 is *0; thus, starting at this position, Alice has no winning strategy, but Bob does. Furthermore, the position equivalent to P_5 contains the possibility $P_1 + P_1$, which denotes the disjoint union[1] of two graphs P_1; this possibility is the position that follows a move on the central vertex of P_5. Note that the position $P_1 + P_1$ is equivalent to two piles with one stone each in the Nim game, and by Theorem 2.2 the nimber of this position is $*6103 - JBCS - Computervisiontoattendancerecognition : abibliometricanalysis - working(Copy)0$, from the calculation $1 \oplus 1 = 0$.

Graph	Position	Mex calculation	Nimber
$\varnothing$	-	-	*0
P_1	$\{\varnothing\}$	$\text{mex}\{*0\} = *1$	*1
P_2	$\{\varnothing\}$	$\text{mex}\{*0\} = *1$	*1
P_3	$\{\varnothing, P_1\}$	$\text{mex}\{*0, *1\} = *2$	*2
P_4	$\{P_1, P_2\}$	$\text{mex}\{*1, *1\} = *0$	*0
P_5	$\{P_1 + P_1, P_2, P_3\}$	$\text{mex}\{*0, *1, *2\} = *3$	*3

Table 1. Nimbers of some paths.

We now introduce the following notation. Note that, in Kayles, each position corresponds to a graph G (which can be disconnected) or to a set of graphs $\mathcal{J}(G)$, which characterizes the position G through the following positions that can be reached from G by a move. For example, in Table 1, we have: (a) P_5 (last row) is a position; (b) $P_1 + P_1$ is one of the positions following P_5 through a move on the central vertex; (c) $\mathcal{J}(G) = \{P_1 + P_1, P_2, P_3\}$ is equivalent to position $G = P_5$. Two equivalent positions have the same nimber, and we use the symbol $\equiv$[2] for this purpose. Thus, $P_5 \equiv \{P_1 + P_1, P_2, P_3\} \equiv *3$. A singleton $\{H\}$ will be simply denoted as H. In some situations, it will be useful to denote the nimber of a position G as $nb(G)$.

2.1 Caterpillars

A *caterpillar* is a tree for which the removal of all leaves produces a path. A caterpillar is denoted as a sequence $(a_1, \ldots, a_n)$, where: (a) a_k is the number of leaves adjacent to vertex v_k, $1 \leq k \leq n$; (b) $P = v_1 \ldots v_n$ is a path. The caterpillar in Figure 1 can be denoted as $(2, 0, 1, 2, 0)$, $(2, 0, 1, 3)$, $(0, 1, 0, 1, 2, 0)$, or $(0, 1, 0, 1, 3)$, as well as the reverse sequences thereof. The trivial caterpillar is denoted as (0). For a caterpillar G, $R(G)$ is defined as the set of all sequences that denote G.

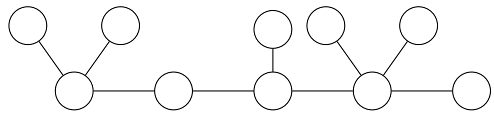

Figure 1. A caterpillar.

2.2 Calculating nimbers of caterpillars

Using the strategy illustrated in Table 1, we describe an algorithm that calculates the Sprague-Grundy function in Kayles when the board is an input caterpillar $G = (a_1, \ldots, a_n)$, given as a vector $A[1..n]$ such that $A[k] = a_k, 1 \leq k \leq n$. The algorithm is presented in a recursive top-down strategy with memoization, as in Bodlaender *et al.* [2015].

In lines 1 to 3, Algorithm 1 checks the vector limits; if $i > j$, then the input caterpillar is empty, and thus the algorithm must return 0. Next, the algorithm checks whether the $nb(A[i..j])$ is already tabulated (lines 4 to 6). If not, the algorithm deals with case $i = j$ (lines 7 to 13), and then with case $i < j$, in which it builds the set S of nimbers of all possible positions that follow a move in A. Observe that the case $i = j$ and $A[i] > 0$ amounts to finding the nimber of a star with $A[i]$

[1]The symbol $+$ in this work denotes the union of disjoint graphs that form a position, as is usual in combinatorial game theory.

[2]Note that the symbol $\equiv$ is also used to denote modular congruence. The intended meaning of the symbol will be clear from the context.

Algorithm 1 CaterNimb - Calculates the nimber of a caterpillar

Require: vector $A[i..j]$ representing a caterpillar
Ensure: $A[k] \geq 0, i \leq k \leq j$
1: **if** $i > j$ **then**
2: **return** 0
3: **end if**
4: **if** $nb(A[i..j])$ has already been computed **then**
5: **return** $nb(A[i..j])$
6: **end if**
7: **if** $i = j$ **then**
8: **if** $A[i] = 0$ **then**
9: **return** 1
10: **else**
11: **return** $((A[i] + 1) \bmod 2) + 1$
12: **end if**
13: **end if**
14: $S \leftarrow \emptyset$
15: $S \leftarrow S \cup \{\text{CaterNimb}(A[i+2..j]) \oplus (A[i+1] \bmod 2)\}$
16: ▷ a move on the first vertex
17: **if** $A[i] > 0$ **then**
18: $A' \leftarrow A$
19: $A'[i] \leftarrow A[i] - 1$
20: $a \leftarrow \text{CaterNimb}(A'[i..j])$
21: $c \leftarrow (A[i] - 1) \bmod 2$
22: $S \leftarrow S \cup \{a \oplus c\}$
23: ▷ a move on a leaf of the first vertex
24: **end if**
25: $S \leftarrow S \cup \{\text{CaterNimb}(A[i..j-2]) \oplus (A[j-1] \bmod 2)\}$
26: ▷ a move on the last vertex
27: **if** $A[j] > 0$ **then**
28: $A' \leftarrow A$
29: $A'[j] \leftarrow A[j] - 1$
30: $a \leftarrow \text{CaterNimb}(A'[i..j])$
31: $c \leftarrow (A[j] - 1) \bmod 2$
32: $S \leftarrow S \cup \{a \oplus c\}$
33: ▷ a move on a leaf of the last vertex
34: **end if**
35: **for** $k = i + 1 \ldots j - 1$ **do**
36: ▷ moves on intermediate vertices
37: $a \leftarrow \text{CaterNimb}(A[i..k-2])$
38: $b \leftarrow \text{CaterNimb}(A[k+2..j])$
39: $c \leftarrow (A[k-1] + A[k+1]) \bmod 2$
40: $S \leftarrow S \cup \{a \oplus b \oplus c\}$
41: ▷ a move on the k-th vertex of the main path)
42: **if** $A[k] > 0$ **then**
43: ▷ if there are leaves adjacent to the k-th vertex
44: $A' \leftarrow A$
45: $A'[k] \leftarrow A[k] - 1$
46: $a \leftarrow \text{CaterNimb}(A'[i..k-1])$
47: $b \leftarrow \text{CaterNimb}(A'[k+1..j])$
48: $c \leftarrow (A[k] - 1) \bmod 2$
49: $S \leftarrow S \cup \{a \oplus b \oplus c\}$
50: ▷ a move on a leaf of the k-th vertex
51: **end if**
52: **end for**
53: $nb(A[i...j]) \leftarrow \text{mex}\, S$
54: ▷ tabulate the result
55: **return** $nb(A[i..j])$

leaves, which is equal to $\text{mex}\{0, (A[i]+1) \bmod 2\} = (A[i]+1) \bmod 2) + 1$.

In line 14, S is initialized. In lines 15 and 25, the algorithm computes the nimbers of the positions resulting from moves on the first and last vertices (v_i and v_j) of the main path, respectively. These moves can disconnect the caterpillar into a smaller caterpillar and a set of isolated vertices, and the nimber is computed using the Nim-sum.

Additionally, in lines 17–24 and 27–34, the algorithm considers moves on leaves adjacent to the endpoints v_i and v_j, whenever $A[i] > 0$ or $A[j] > 0$. These moves reduce the number of leaves at the corresponding endpoint without disconnecting the caterpillar, and are handled by recursively evaluating the modified vector.

In lines 35 – 52, the algorithm computes the nimbers of the positions resulting from moves on intermediate vertices v_k, $i+1 \leq k \leq j-1$. A move on v_k produces $A[k-1]+A[k+1]$ isolated vertices, while a move on a leaf adjacent to v_k reduces $A[k]$ by one and splits the caterpillar into two independent subcaterpillars.

In line 53, $nb(A[i..j])$ is finally computed as the mex of the set S.

Each recursive call CaterNimb($A[i'..j']$) is a subproblem corresponding to a contiguous subcaterpillar; thus, the total number of distinct subproblems is given by the number of contiguous subvectors of $A[1..n]$, which is $O(n^2)$. On the other hand, each subproblem associated with $A[i'..j']$ has size $n' = j' - i' + 1$ and needs $O(n')$ calculations. Since $n' = O(n)$, by converting the recursive top-down strategy into a standard bottom-up iterative method, we can solve Kayles on caterpillars in $O(n^3)$ time. The structure of a caterpillar simplifies the computation of its nimber, in comparison with a cocomparability graph. However, the asymptotic complexity of Algorithm 1 is the same as that provided in [Bodlaender and Kratsch, 2002].

3 Periodicity of Kayles on paths

Let $[\alpha_1, \alpha_2, \ldots]$ be a sequence of positions for a certain game Γ, such that $\alpha_i \equiv *a_i, i \geq 1$. The sequence $[*a_i]_{i \geq 1}$ is called the *nim-sequence* of positions $\alpha_1, \alpha_2, \ldots$ in game Γ.

In Berlekamp *et al.* [2001], the authors discuss the properties of a game called Dawson's Chess, which is played on a $3 \times n$ board, with the first and last rows filled with pawns. The game is impartial: the pawns do not belong to a specific player, and both players can move any pawn available on the board. Each pawn moves as in Chess: one step forward (up/down), capturing on its immediate diagonal squares; moreover, capture is mandatory. As in Kayles, this game uses the normal play rule, that is, the player who cannot move a pawn loses. An interesting property of this game is that the nim-sequence of the $3 \times n$ boards, $n \geq 1$, exhibits a periodic pattern of length $p = 34$.

It is easy to see that playing Dawson's Chess on a $3 \times n$ board is equivalent to playing Kayles on the graph P_n, because moving a pawn forces four alternating capturing turns, "breaking" the board into two parallel games. Therefore, we can transfer the Dawson's Chess periodicity property for Kayles, as stated in Theorem 3.1 below.

Theorem 3.1 (Periodicity of Kayles on paths)
[Berlekamp *et al.*, 2001]

In the game Kayles, the nim-sequence for paths is periodic, with period $p = 34$. In other words, for all $n > 51$,

$$P_n \equiv P_{n+p} \equiv *a_{n \bmod 34},$$

where $[a_0, \ldots, a_{33}] = [3, 3, 0, 1, 1, 3, 0, 2, 1, 1, 0, 4, 5, 3, 7, 4, 8, 1, 1, 2, 0, 3, 1, 1, 0, 3, 3, 2, 2, 4, 4, 5, 5, 9]$.

Note that Theorem 3.1 provides a simple method for solving Kayles on paths, which amounts to calculating a modular function (in constant time[3]). At this point, an interesting question is whether, given an infinite ordered family $\mathcal{F}$ of caterpillars, the nim-sequence of $\mathcal{F}$ also follows this periodic pattern. In addition to the theoretical interest of this investigation, periodicity proofs can drastically reduce the complexity given in Theorem 3.1. This is our subject in the next sections.

4 Reduced caterpillars

The nimbers from here on will always refer to the game of Kayles.

To avoid ambiguity, we will denote a caterpillar as a sequence that starts and ends with 0's. In other words, if $G = (a_1, \ldots, a_n)$ is a caterpillar, we always assume $a_1 = a_n = 0$. For example, the caterpillar illustrated in Figure 1 will be denoted by $(0, 1, 0, 1, 2, 0)$ or its reverse.

Let $G = (a_1, a_2, \ldots, a_n)$ be a caterpillar. The *reduced caterpillar* of G is the caterpillar $G' = (a'_1, a'_2, \ldots, a'_n)$ such that:

$$a'_i = \begin{cases} 0, & \text{if } a_i = 0 \\ 1, & \text{if } a_i > 0 \text{ e } a_i \equiv 1 \pmod 2 \\ 2, & \text{if } a_i > 0 \text{ e } a_i \equiv 0 \pmod 2 \end{cases}$$

In addition, a'_i is the number of leaves adjacent to v'_i, and P' is the path $v'_1, \ldots, v'_n$. As an example, the reduced caterpillar of $G = (0, 5, 6, 3, 0)$ is $G' = (0, 1, 2, 1, 0)$.

Theorem 4.2 below tells us that, to calculate the nimber of a caterpillar, it is enough to restrict ourselves to its reduced caterpillar. This significantly simplifies the execution of Algorithm 1, as well as our subsequent discussions on the periodicity of nimbers. Before proving Theorem 4.2, we need the following lemma:

Lemma 4.1 (Parity of the second vertex)
Let $G = (a_1, \ldots, a_n)$ and $G' = (b_1, \ldots, b_n)$ be caterpillars with $n \geq 3$, $a_2 \equiv b_2 \pmod 2$, and $a_i = b_i$ $(3 \leq i \leq n)$. Then, $G \equiv G'$.

Proof. We use induction on n. If $n = 3$, G and G' are two stars whose numbers of leaves have the same parity, in which case the result holds trivially. Now suppose that the result holds for caterpillars $G = (a_1, \ldots, a_{n'})$ and $G' = (b_1, \ldots, b_{n'})$ with $n' < n$. We consider below the possible moves and the positions that follow these moves.

Case 1: Move on v_1 or on a leaf of v_2. The corresponding moves in G' are, respectively, on v'_1 or on a leaf of v'_2. The position following G is $a_2K_1 + (a_3, \ldots, a_n)$ (if $a_3 = 0$)[4] or $a_2K_1 + (0, a_3 - 1, \ldots, a_n)$ (if $a_3 > 0$), while the position following G' is $b_2K_1 + (b_3, \ldots, b_n)$ (if $b_3 = 0$) or $b_2K_1 + (0, b_3 - 1, \ldots, b_n)$ (if $b_3 > 0$). Since $a_2 \equiv b_2 \pmod 2$, the positions following G and G' have clearly the same nimber.

Case 2: Move on v_n or on a leaf of v_{n-1}. This case is similar to Case 1.

Case 3: A move on v_2. The corresponding move in G' is on v'_2. The position following G is $a_3K_1 + (a_4, \ldots, a_n)$ (if $a_4 = 0$) or $a_3K_1 + (0, a_4 - 1, \ldots, a_n)$ (if $a_4 > 0$), while the position following G' is $b_3K_1 + (b_4, \ldots, b_n)$ (if $b_4 = 0$) or $b_3K_1 + (0, b_4 - 1, \ldots, b_n)$ (if $b_4 > 0$). Since $a_3 = b_3$, the positions following G and G' have the same nimber.

Case 4: Move on v_{n-1}. This case is similar to Case 3.

Case 5: Move on v_i, $3 \leq i \leq n-2$. In this case, the position following G is $(a_{i-1} + a_{i+1})K_1 + G_1 + G_2$, where G_1 is either $(a_1, \ldots, a_{i-2})$ (if $a_{i-2} = 0$) or $(a_1, \ldots, a_{i-2} - 1, 0)$ (if $a_{i-2} > 0$), and G_2 is either $(a_{i+2}, \ldots, a_n)$ (if $a_{i+2} = 0$) or $(0, a_{i+2} - 1, \ldots, a_n)$ (if $a_{i+2} > 0$). Accordingly, the corresponding move in G' is on v'_i, leading to the position $(b_{i-1} + b_{i+1})K_1 + G'_1 + G'_2$, where G'_1 is either $(b_1, \ldots, b_{i-2})$ (if $b_{i-2} = 0$) or $(b_1, \ldots, b_{i-2} - 1, 0)$ (if $b_{i-2} > 0$), and $G'_2 = G_2$. Since $a_2 \equiv b_2 \pmod 2$, by the induction hypothesis $G_1 \equiv G'_1$; in addition, since $(a_{i-1} + a_{i+1}) \equiv (b_{i-1} + b_{i+1}) \pmod 2$, the positions following G and G' have the same nimber.

Case 6: Move on a leaf connected to v_i, $3 \leq i \leq n-2$. This case is similar to the previous one. The position following G is $(a_i - 1)K_1 + G_1 + G_2$, where G_1 is either $(a_1, \ldots, a_{i-1})$ (if $a_{i-1} = 0$) or $(a_1, \ldots, a_{i-1} - 1, 0)$ (if $a_{i-1} > 0$), and G_2 is either $(a_{i+1}, \ldots, a_n)$ (if $a_{i+1} = 0$) or $(0, a_{i+1} - 1, \ldots, a_n)$ (if $a_{i+1} > 0$). Accordingly, the corresponding move in G' is on a leaf of v'_i, leading to the position $(b_i - 1)K_1 + G'_1 + G'_2$, where G'_1 is either $(b_1, \ldots, b_{i-1})$ (if $b_{i-1} = 0$) or $(b_1, \ldots, b_{i-1} - 1, 0)$ (if $b_{i-1} > 0$), and $G'_2 = G_2$. Since $a_2 \equiv b_2 \pmod 2$, by the induction hypothesis $G_1 \equiv G'_1$; in addition, since $a_i \equiv b_i \pmod 2$, the positions following G and G' have the same nimber. This completes the proof of the lemma. □

Theorem 4.2 *Let $G = (a_1, \ldots, a_n)$ be a caterpillar and $G' = (a'_1, \ldots, a'_n)$ its reduced caterpillar. Then, $G \equiv G'$.*

Proof. We use induction on n. If $n \leq 2$, the theorem trivially follows. If $n = 3$, G and G' are stars with $a_2 + 2$ and $a'_2 + 2$ leaves, respectively (recall that $a_1 = a_n = 0$). Recall that the nimber of a star with $f \geq 2$ leaves is $*1$ if f is odd, and $*2$ if f is even. Since $a_2 \equiv b_2 \pmod 2$, the result follows.

Assume $n \geq 4$. We now use a proof structure very similar to that of Lemma 4.1, by listing all the possible corresponding moves in G and G' and showing that in all of them the resulting positions have the same nimber. Analyzing Case 1 in detail will suffice. The remaining ones are only outlined.

Case 1: Move on v_1 or v_n. Assume a move on v_1 (the move on v_n follows the same arguments). The position following

[3]We assume that calculating $n \bmod c$, for some constant c, takes $O(1)$ time.

[4]The notation pK_1 denotes the union of p trivial graphs (when $p > 0$) or the empty graph (when $p = 0$).

G is $G_1 = a_2K_1 + (a_3, \ldots, a_n)$ (if $a_3 = 0$) or $G_1 = a_2K_1 + (0, a_3 - 1, \ldots, a_n)$ (if $a_3 > 0$). The position following G' is $G'_1 = a'_2K_1 + (a'_3, \ldots, a'_n)$ (if $a'_3 = 0$) or $G'_1 = a'_2K_1 + (0, a'_3 - 1, \ldots, a'_n)$ (if $a'_3 > 0$). Observe that:

• Since $a_2 \equiv a'_2 \pmod 2$, $a_2K_1 \equiv a'_2K_1$.

• If $a_3 = 0$, then, by induction, $(a_3, \ldots, a_n) \equiv (a'_3, \ldots, a'_n)$.

• If $a_3 > 0$ and a_3 is even, $(0, a'_3 - 1, \ldots, a'_n)$ is the reduced caterpillar of $(0, a_3 - 1, \ldots, a_n)$. Therefore, by induction, these caterpillars have the same nimber.

• If $a_3 > 0$ and a_3 is odd, let us show that the caterpillars $(0, a_3 - 1, \ldots, a_n)$ and $(0, a'_3 - 1, \ldots, a'_n)$ have the same nimber. The case $a_3 = 1$ is trivial, because in this case $a_3 = a'_3$. For $a_3 > 1$, it suffices to observe that the reduced caterpillar of $(0, a_3 - 1, a_4, \ldots, a_n)$ is $(0, 2, a'_4, \ldots, a'_n)$, and, by induction, these caterpillars have the same nimber. Furthermore, by Lemma 4.1, $(0, 2, a'_4 \ldots, a'_n)$ and $(0, a'_3 - 1, a'_4 \ldots, a'_n) = (0, 0, a'_4, \ldots, a'_n)$ have the same nimber.

From the above facts, $G_1 \equiv G'_1$.

Case 2: Move on v_2 or v_{n-1}. Assume a move on v_2. The position following G is $a_3K_1 + G_1$, where G_1 is one of the following graphs: (0), if $n = 4$; $(0, a_4 - 1, \ldots, a_n)$, if $n > 4$ and $a_4 > 0$; $(a_4, \ldots, a_n)$, if $n > 4$ and $a_4 = 0$. The position following G' is $a'_3K_1 + G'_1$, where G'_1 is one of the following graphs: (0), if $n = 4$; $(0, a'_4 - 1, \ldots, a'_n)$, if $n > 4$ and $a_4 > 0$; $(a'_4, \ldots, a'_n)$, if $n > 4$ and $a_4 = 0$.

Case 3: Move on v_i, $3 \le i \le n-2$. The position following G is $(a_{i-1} + a_{i+1})K_1 + G_1$, where G_1 is the union of two caterpillars originated by removing the vertices v_{i-1}, v_i and v_{i+1}, as well as the a_i leaves connected to v_i. The corresponding position following G' is of the form $(a'_{i-1} + a'_{i+1})K_1 + G'_1$.

Case 4: Move on a leaf connected to v_i, $2 \le i \le n - 1$. Let F_i be the set of leaves connected to vertex v_i. Note that this move produces a position formed by the union of the following graphs: $(a_i - 1)K_1$; the caterpillar G_1 induced by the set of vertices $\{v_1, \ldots, v_{i-1}\} \cup (\cup_{j=1}^{i-1} F_j)$; and the caterpillar G_2 induced by the set of vertices $\{v_{i+1}, \ldots, v_n\} \cup (\cup_{j=i+1}^{n} F_j)$. The corresponding position following G' is the union of the graphs $(a'_i - 1)K_1$, G'_1 and G'_2, where the latter two correspond to caterpillars G_1 and G_2. □

5 Obtaining nimbers via periodicity

The following definitions are useful. A *preperiod* of a periodic nim-sequence $S = [*a_1, *a_2, *a_3, \ldots]$ is an integer t such that, for all $n > t$, $*a_n \equiv *a_{n+p}$, where p is the period of the sequence. In this case, we say that S is *$\langle p, t\rangle$-periodic* or, more simply, *p-periodic*. The *transient* of S is the minimum t for which S is $\langle p, t\rangle$-periodic. For integers $a \ge 0$ and $n \ge 0$, a^n denotes the sequence $a, \ldots, a$ with n elements (a^0 is the empty sequence).

From now on, we assume that a reduced caterpillar is given in the form $(0, 0^j, m_1, m_2, \ldots, m_r, 0)$, $r \ge 1$, where each m_i is a value in the set $\{0, 1, 2\}$. Let k be the decimal number corresponding to the base-3 number with digits $m_1m_2 \ldots m_r$. Then the reduced caterpillar $(0, 0^j, m_1, m_2, \ldots, m_r, 0)$ will be denoted as $\boldsymbol{(}j, k\boldsymbol{)}$ (we use bold parentheses to distinguish this short form from the previous notation). Assume $m_1 \ne 0$, except in the case $k = 0$ (for which $r = 1$ and $m_1 = 0$). For example, $(0, 1, 2, 0)$ is denoted as $\boldsymbol{(}0, 5\boldsymbol{)}$, $(0, 0, 1, 2, 0)$ as $\boldsymbol{(}0, 5\boldsymbol{)}$, and, in general, $(0, 0^j, 1, 2, 0)$ is the caterpillar $\boldsymbol{(}j, 5\boldsymbol{)}$.

Let $C_k = \{\boldsymbol{(}j, k\boldsymbol{)} \mid j \ge 0\}$. Note that C_0 consists of paths with at least three vertices. By Theorem 3.1, the nim-sequence of C_0 is $\langle 34, 49\rangle$-periodic, since $\boldsymbol{(}0, 0\boldsymbol{)} = P_3$ is its first element and $\boldsymbol{(}48, 0\boldsymbol{)} = P_{51}$. See Figure 2. By applying Algorithm 1, we experimentally verified that the period 34 seems to hold; we examined all families C_k with $0 \le k \le 199$. In each family, we calculated the nimbers of all caterpillars $\boldsymbol{(}j, k\boldsymbol{)}$ for $0 \le j \le 500$. Although a period value of 34 remains unchanged, the transient values vary. See Table 2.

The next theorem gives a sufficient condition for the nim-sequence of class C_k to be 34-periodic.

Theorem 5.1 *Let k and t_k be positive constants. Then, C_k is $\langle 34, t_k\rangle$-periodic if the following conditions hold:*

1. *There exist $t_1, \ldots, t_{k-1}$ such that the class $C_{k'}$ is $\langle 34, t_{k'}\rangle$-periodic, for every $1 \le k' \le k - 1$;*
2. *$\boldsymbol{(}i, k\boldsymbol{)} \equiv \boldsymbol{(}j, k\boldsymbol{)}$ for all $i, j \in (t_k, t_k + 122]$ such that $i \equiv j \pmod{34}$;*
3. *$t_k + 88 > \max\{t_1, \ldots, t_{k-1}\}$.*

Proof. Let k and t_k be such that (1), (2), and (3) are satisfied. We need to prove that the interval $(t_k, t_k + 122]$ in the second condition extends infinitely to the right. For this purpose, we use induction on j to show that $\boldsymbol{(}j, k\boldsymbol{)} \equiv \boldsymbol{(}j + 34, k\boldsymbol{)}$ for all $j + 34 > t_k + 122$ (i.e., $j > t_k + 88$). Thus, suppose that the caterpillars are guaranteed to be periodic in the interval $[t_k, j + 34]$ with $j > t_k + 88$. We need to show that the game $\boldsymbol{(}j, k\boldsymbol{)} + \boldsymbol{(}j + 34, k\boldsymbol{)}$ has a winning strategy for the second player. We analyze the effects of the first player's moves in order to provide this strategy. Consider the following representation of caterpillars $\boldsymbol{(}j, k\boldsymbol{)}$ and $\boldsymbol{(}j + 34, k\boldsymbol{)}$, where the corresponding vertices to be played are shown below each representation:

$$\boldsymbol{(}j, k\boldsymbol{)} = (\underset{u^l}{0}, \underbrace{0, \ldots, 0}_{u_j, \ldots, u_1}, \underbrace{a_i, \ldots, a_1}_{u_i^a, \ldots, u_1^a}, \underset{u_r}{0})$$
$$\boldsymbol{(}j + 34, k\boldsymbol{)} = (\underset{v^l}{0}, \underbrace{0, \ldots, 0}_{v_{j+34}, \ldots, v_1}, \underbrace{a_i, \ldots, a_1}_{v_i^a, \ldots, v_1^a}, \underset{v_r}{0})$$

We divide the analysis into the following cases:

1. A move on u^l (resp., v^l) leads to a position $\alpha + \boldsymbol{(}j + 34, k\boldsymbol{)} = \boldsymbol{(}j-2, k\boldsymbol{)} + \boldsymbol{(}j+34, k\boldsymbol{)}$ ($\boldsymbol{(}j, k\boldsymbol{)} + \alpha' = \boldsymbol{(}j, k\boldsymbol{)} + \boldsymbol{(}j + 32, k\boldsymbol{)}$). The second player has a winning move on vertex v^l (u^l) of the second (first) component to obtain $\boldsymbol{(}j - 2, k\boldsymbol{)} + \boldsymbol{(}j + 32, k\boldsymbol{)}$, which is a $\mathcal{P}$-position when $j > t_k + 2$.
2. A move on u_1 (v_1) leads to a position $\alpha + \boldsymbol{(}j + 34, k\boldsymbol{)} = P_{j-1} + \boldsymbol{(}0, k'\boldsymbol{)} + \boldsymbol{(}j + 34, k\boldsymbol{)}$ ($\boldsymbol{(}j, k\boldsymbol{)} + \alpha' = \boldsymbol{(}j, k\boldsymbol{)} + P_{j+33} + \boldsymbol{(}0, k'\boldsymbol{)}$) with $k' < k$. The second player has a winning move on vertex v_1 (u_1) of the second (first) component to obtain $P_{j-1} + \boldsymbol{(}0, k'\boldsymbol{)} + P_{j+33} + \boldsymbol{(}0, k'\boldsymbol{)} = P_{j-1} + P_{j+33}$, which is a $\mathcal{P}$-position since we are interested in $j > t_k + 88$, implying $j - 1 > 51$. The same argument applies to moves on u_2 (v_2).

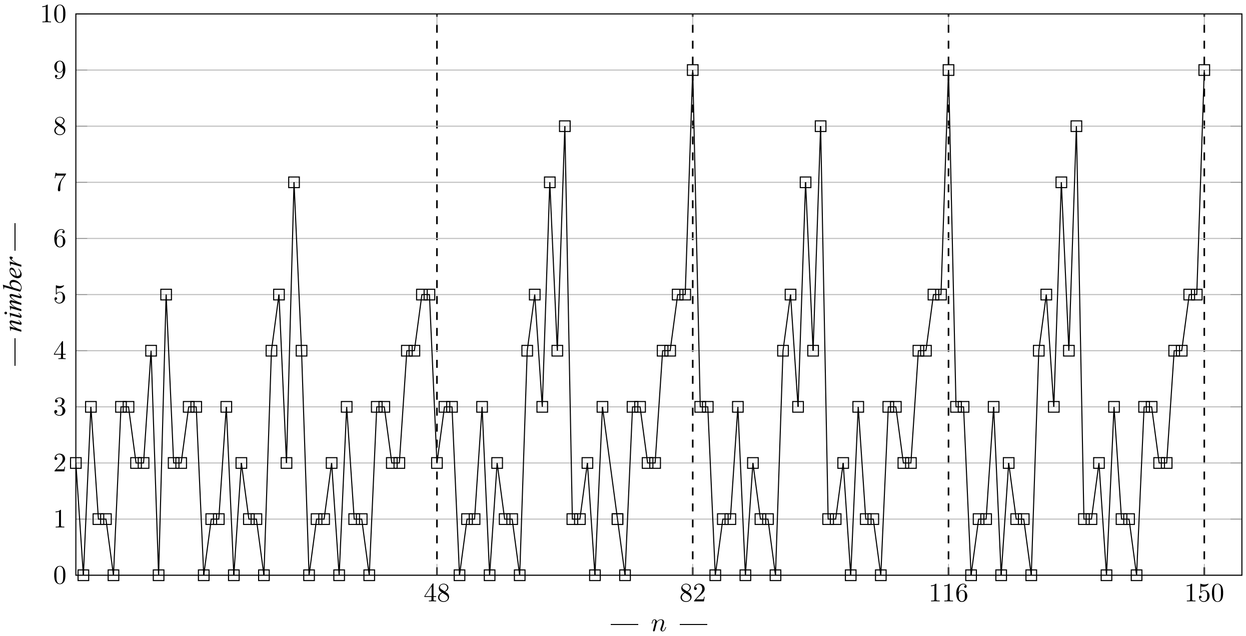

Figure 2. Nim-sequence of C_0, with $0 \leq n \leq 150, p = 34, t = 49$. The dashed line at $n = 48$ indicates the end of the transient part, and the dashed lines to the right of it mark period boundaries. Note that $n = 0$ corresponds to the path P_3, with nimber 2.

Table 2. Minimum values of t for which $(j, k) \equiv (j + 34, k)$, $t < j \leq 466$.

k	t	k	t	k	t	k	t	k	t	k	t	k	t	k	t
0	49	25	292	50	74	75	378	100	323	125	49	150	127	175	214
1	310	26	378	51	202	76	361	101	-1	126	162	151	201	176	213
2	49	27	339	52	145	77	378	102	164	127	49	152	127	177	172
3	49	28	446	53	202	78	292	103	290	128	162	153	136	178	127
4	74	29	339	54	164	79	199	104	164	129	135	154	162	179	172
5	49	30	120	55	307	80	292	105	161	130	135	155	136	180	213
6	310	31	261	56	164	81	289	106	155	131	135	156	202	181	244
7	378	32	120	57	213	82	257	107	161	132	129	157	184	182	206
8	310	33	164	58	172	83	289	108	209	133	129	158	202	183	339
9	120	34	161	59	213	84	339	109	116	134	129	159	145	184	251
10	164	35	164	60	339	85	292	110	209	135	81	160	185	185	339
11	120	36	173	61	292	86	339	111	173	136	184	161	145	186	292
12	49	37	81	62	339	87	446	112	113	137	81	162	398	187	257
13	135	38	173	63	275	88	446	113	173	138	116	163	136	188	292
14	49	39	49	64	180	89	446	114	81	139	267	164	398	189	292
15	74	40	49	65	275	90	117	115	187	140	116	165	164	190	361
16	202	41	49	66	310	91	310	116	81	141	209	166	161	191	292
17	74	42	135	67	310	92	117	117	49	142	214	167	164	192	275
18	213	43	129	68	310	93	120	118	49	143	209	168	307	193	187
19	339	44	135	69	257	94	292	119	49	144	209	169	328	194	275
20	213	45	116	70	122	95	120	120	49	145	185	170	307	195	180
21	310	46	209	71	257	96	261	121	49	146	209	171	201	196	114
22	257	47	116	72	361	97	292	122	49	147	74	172	49	197	180
23	310	48	74	73	292	98	261	123	49	148	74	173	201	198	310
24	378	49	127	74	361	99	257	124	49	149	74	174	213	199	310

3. A move on u_x $(2 < x \leq j)$ leads to a position $\alpha + (j + 34, k) = P_{j-x} + (x-3,k) + (j+34,k)$. The second player has a winning move on vertex v_x or v_{x+34} of the second component to obtain $P_{j-x} + (x-3,k) + P_{j-x} + (x+31,k) = (x-3,k) + (x+31,k)$ or $P_{j-x} + (x-3,k) + P_{j+34-x} + (x-3,k) = P_{j-x} + P_{j+34-x}$, respectively.
We can guarantee that at least one of these positions is a $\mathcal{P}$-position if $x - 3 > t_k$ and $x + 31 < j + 34$ ($x-3$ and $x+31$ belong to $(t_k, j+34]$) or $j - x > 51$. If $x < j - 51$, then the second condition is satisfied; otherwise, if $x \geq j - 51$, $x - 3 > t_k$ (i.e., $x > t_k + 3$) is satisfied when we restrict to $j - 51 > t_k + 3$ (i.e., $j > t_k + 54$), while $x + 31 < j + 34$ is always satisfied.
4. A move on v_x $(2 < x \leq j + 34)$ leads to a position $(j,k) + \alpha' = (j,k) + P_{j+34-x} + (x-3,k)$. The second player has a winning move on vertex u_x $(x \leq j)$ or u_{x-34} $(x \geq 37)$ of the first component to obtain $P_{j-x} + (x-3,k) + P_{j+34-x} + (x-3,k) = P_{j-x} + P_{j+34-x}$ or $P_{j+34-x} + (x-37,k) + P_{j+34-x} + (x-3,k) = (x-37,k) + (x-3,k)$.
We can guarantee that at least one of these positions is a $\mathcal{P}$-position if $x \leq j$ and $j - x > 51$ (u_x is a winning move) or $x > 36$, $t_k < x - 37$, and $x - 3 < j + 34$ (u_{x-34} is a winning move). This can be guaranteed when $j > t_k + 88$. To see this, we analyze the value of x:
    a. $2 < x \leq 36$: The first condition is always satisfied since we are interested in $j > t_k + 88$. Therefore, $x \leq 36 < t_k + 88 < j$ and $j - x \geq j - 36 > t_k + 52 > 51$.
    b. $36 < x \leq j$: If $j - x > 51$, the first condition is satisfied; otherwise, we have $x \geq j - 51$, and we need to satisfy the second. Since $x \leq j$, it also holds that $x - 3 < j + 34$, so it suffices that $t_k < x - 37$, i.e., $x > t_k + 37$. Since $x \geq j - 51$, we also guarantee this inequality by bounding $j - 51 > t_k + 37$, i.e., $j > t_k + 88$.
    c. $j < x \leq j + 34$: the second condition is satisfied, because $x > j$ guarantees $x > 36$; $x > j > t_k + 88 > t_k + 37$ guarantees $t_k < x - 37$; and $x \leq j + 34$ guarantees $x - 3 < j + 34$.
5. A move on u_x^a (v_x^a) leads to a position $\alpha + (j+34,k) = (j,k') + (0,k'') + (j+34,k)$ ($(j,k) + \alpha' = (j,k) + (j+34,k') + (0,k'')$), with $k' < k$ and $k'' < k$. The second player has a winning move on the corresponding vertex of the other component (that is, if the move was played on u_x^a, there is a winning move on v_x^a, and vice versa) to obtain $(j,k') + (0,k'') + (j+34,k') + (0,k'') = (j,k') + (j+34,k')$, which is guaranteed to be a $\mathcal{P}$-position when $j > t_{k'}$. The same argument applies to a move on a leaf of u_x^a or v_x^a.
6. A move on u'' (v'') leads to a position $\alpha + (j+34,k) = (j,k') + (j+34,k)$ ($(j,k) + \alpha' = (j,k) + (j+34,k')$), with $k' < k$. The second player has a winning move on vertex v'' (u'') of the second (first) component to obtain $(j,k') + (j+34,k')$, which is guaranteed to be a $\mathcal{P}$-position when $j > t_{k'}$.

Therefore, for the game $(j,k) + (j+34,k)$ to have a winning strategy for the second player, it suffices that $j > \max\{t_k + 2, t_k + 54, t_k + 88, t_{k'}\} = \max\{t_k + 88, t_{k'}\} = t_k + 88$ (third condition). Hence, the class of caterpillars C_k is $\langle 34, t_k \rangle$-periodic. $\square$

Theorem 5.1 tells us how to determine the nimber of a caterpillar (j,k) in constant time as follows: (a) compute $nb(j',k)$ for all $j' \leq t_k$; (b) compute $nb(j',k)$ for $t_k + 1 \leq j' \leq t_k + 34$; (c) set $j' = (j + 1 - t_k) \bmod 34$ and return $nb(j',k)$. Observe that steps (a) and (b) can be executed via Algorithm 1 in $O(1)$ time overall, since k and t_k are tabulated constants, as in Table 2.

Corollary 5.2 *Let C_k be a class satisfying conditions 1, 2, and 3 of Theorem 5.1, for some positive constants k and t_k. Then, $nb(G)$ can be computed in $O(1)$ time for all $G \in C_k$.*

Computational experiments based on Theorem 5.1 showed that the sufficient condition given in the theorem applies to all classes C_k for $1 \leq k \leq 6999$, i.e., all of these classes are 34-periodic. Hence:

Corollary 5.3 *Let $G \in C_k$, for $k \in \{1, \ldots, 6999\}$. Then, $nb(G)$ can be computed in $O(1)$ time.*

The preceding discussions and results presented in this section leads to the formulation of the following conjecture:

Conjecture 5.4 *Let $k \geq 0$ be an integer. Then, there exists an integer $t_k \geq 0$ such that C_k is $\langle 34, t_k \rangle$-periodic, with transient t_k.*

6 Solving Powers of Paths and Cycles

For an integer $k > 0$, C_n^k denotes the graph obtained from a cycle C_n by adding edges between vertices at distance at most k in the cycle. The graph P_n^k is defined similarly from the path P_n. These graphs are called the *kth power of the cycle C_n* and *kth power of the path P_n*, respectively. Note that $C_0^k = P_0^k = \varnothing$, $C_n^1 = C_n$, and $P_n^1 = P_n$. In addition, $C_n^k = K_n$ if $k \geq \lfloor n/2 \rfloor$, and $P_n^k = K_n$ if $k \geq n - 1$.

In this section, we use the techniques and results of the previous sections to solve Kayles on powers of cycles and paths. First, we need the following property:

Lemma 6.1 *Let G be a graph such that $G = C_n^k$ or $G = P_n^k$, and let S be a non-empty independent set of G. Then, all the components of the graph $G' = G - N[S]$ are powers of paths.*

Proof. If G is the kth power of a path, then the components of G' are k-th powers of paths, because G' is an induced sugbgraph of G.

If G is the kth power of a cycle, let $u \in S$. The graph $G - u$ is clearly a kth power of a path. Now, by removing the remaining vertices in $S \setminus \{u\}$ one at a time, and using the fact in the previous paragraph, the result follows. $\square$.

The above lemma says that, starting from C_n^k or P_n^k, all subsequent moves are done on disjoint unions of powers of paths. From this fact, we devise a dynamic programming algorithm to solve Kayles on powers of paths.

Algorithm 2 PowerPathNimb - Computes $nb(P_n^k)$

Require: $P_n^k, 1 \leq k < n-1$
1: **if** $n \leq 0$ **then**
2:     **return** 0
3: **end if**
4: **if** $nb(P_n^k)$ is already computed **then**
5:     **return** $nb(P_n^k)$
6: **end if**
7: $S \leftarrow \emptyset$
8: $j \leftarrow \min\{k, \lfloor n/2 \rfloor\}$
9: **for** $i = 1 \ldots j$ **do**
10:     ▷ a move on v such that $|N[v]| < 2k+1$
11:     $S \leftarrow S \cup \{\text{PowerPathNimb}(P_{n-k-i}^k)\}$
12: **end for**
13: **for** $i = j+1 \ldots \lceil n/2 \rceil$ **do**
14:     ▷ a move on v such that $|N[v]| = 2k+1$
15:     $n_1 \leftarrow \text{PowerPathNimb}(P_{i-k-1}^k)$
16:     $n_2 \leftarrow \text{PowerPathNimb}(P_{n-i-k-1}^k)$
17:     $S \leftarrow S \cup \{n_1 \oplus n_2\}$
18: **end for**
19: $nb(P_n^k) \leftarrow$ mex S
20: **return** $nb(P_n^k)$

To simplify the description of Algorithm 2, negative values of n are allowed, in which case $P_n^k = \varnothing$ (lines 1 to 3). Next, in lines 4 to 6, the algorithm checks whether the nimber of the current position is already computed. If not, it calculates the nimbers of reachable positions from the current position and calculates the mex of them. It finishes by tabulating and returning the new result to be used in next computations.

Theorem 6.2 *Let $n \geq 0, k \geq 1$. Then, $nb(P_n^k)$ can be computed in $O(n^2)$ time.*

Proof. Each recursive call $\text{PowerPathNimb}(P_{n'}^k)$ corresponds to a subproblem of a dynamic programming approach. The number of subproblems is clearly $O(n)$, and each of these subproblems clearly takes $O(n)$ time. By converting the top-down strategy of Algorithm 3 into a bottom-up fashion, the result follows. □

Algorithm 3 PowerCycleNimb - Computes $nb(C_n^k)$

Require: $C_n^k, n \geq 3, 1 \leq k \leq \lfloor n/2 \rfloor$
$nimb \leftarrow \text{PowerPathNimb}(P_{n-2k-1}^k)$
return mex$\{nimb\}$

Since any game on a power of cycle leads to positions that are powers of paths, we can use Algorithm 2 as a subroutine to solve Kayles on powers of cycles. As a direct consequence of Theorem 6.2, we have:

Corollary 6.3 *Let $n \geq 3, k \geq 1$. Then, $nb(C_n^k)$ can be computed in $O(n^2)$ time.*

An interesting remark is that Algorithm 3 returns $*1$ if $\text{PowerPathNimb}(P_{n-2k-1}^k) = *0$, and $*0$ otherwise. Thus:

Corollary 6.4 *Let $n \geq 3, k \geq 1$. Then, $nb(C_n^k) \in \{*0, *1\}$.*

7 Concluding remarks

In this work, we studied the game Kayles played on caterpillars. By establishing an equivalence between caterpillars and their reduced counterparts, we showed that the periodicity of length 34 holds for a wide range of caterpillar classes, enabling the computation of their nimbers in $O(1)$ time. We also presented dynamic-programming algorithms for solving Kayles on powers of paths and cycles, improving upon the complexities achieved by prior algorithms designed for more general graph classes. Future research could explore other graph families where structural properties permit similar dynamic programming approaches or periodicity-based solutions. Whether nimber periodicity can be formally proved for powers of paths or cycles remains an interesting open question.

Impartial game vs. "fair" game. It is worth making a final observation. The concept of impartial game has nothing to do with equal chances of winning for Alice and Bob. For example, in the case of paths, Figure 2 shows that, within a periodic cycle of length 34, the nimber $*0$ occurs only 5 times. In other words, by randomly choosing a value of n within the range $0 \leq n \leq N$, where N is a large integer, Alice's winning probability is close to $\frac{29}{34}$ (it is not exactly equal to $\frac{29}{34}$ because the nimber $*0$ occurs 10 times in the transient). An interesting and little explored question in impartial games is to study Alice and Bob's winning probabilities.

Declarations

Authors' Contributions

F. Protti contributed to the conceptualization and methodology of this study. M. F. Medeiros-de-Souza and F. E. Hanada performed the experiments and formal analysis. M. F. Mederiros-de-Souza is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

This research was partially supported by the CNPq - National Council for Scientific and Technological Development.

Availability of data and materials

The datasets and softwares generated and analyzed during the current study will be made available upon request.

References

Berlekamp, E. R., Conway, J. H., and Guy, R. K. (2001). *Winning Ways for Your Mathematical Plays*, volume 1. A. K. Peters, 2nd edition. DOI: 10.2307/2323620.

Bodlaender, H. L. and Kratsch, D. (2002). Kayles and nimbers. *Journal of Algorithms*, 43:106–119. DOI: 10.1006/jagm.2002.1215.

Bodlaender, H. L., Kratsch, D., and Timmer, S. T. (2015). Exact algorithms for Kayles. *Theoretical Computer Science*, 562:165–176. DOI: 10.1016/j.tcs.2014.09.042.

Bouton, C. L. (1901). Nim, a game with a complete mathematical theory. *Annals of Mathematics*, 3(1/4):35–39. DOI: 10.2307/1967631.

DeVos, M. and Kent, D. A. (2016). *Game Theory: A Playful Introduction*. AMS - American Mathematical Society. DOI: 10.1090/stml/080.

Ferguson, T. S. (2000). *Game Theory*. Class notes for Math 167, Fall 2000, Carnegie Mellon University. DOI: 10.1017/cbo9780511609909.006.

Fleischer, R. and Trippen, G. (2006). Kayles on the Way to the Stars. In van den Herik, H. J., Björnsson, Y., and Netanyahu, N. S., editors, *Computers and Games*, pages 232–245. Springer Berlin Heidelberg. DOI: 10.1007/11674399_16.

Grundy, P. M. (1939). Mathematics and games. *Eureka*, 2:6–8. Book.

Guignard, A. and Sopena, E. (2009). Compound node–Kayles on paths. *Theoretical Computer Science*, 410(21):2033–2044. DOI: 10.1016/j.tcs.2008.12.053.

Kobayashi, Y. (2021). On structural parameterizations of node Kayles. In Akiyama, J., Marcelo, R. M., Ruiz, M.-J. P., and Uno, Y., editors, *Discrete and Computational Geometry, Graphs, and Games*, pages 96–105, Cham. Springer International Publishing. DOI: 10.1007/978-3-030-90048-9_8.

Saldanha, N. C. (2008). *Tópicos em Jogos Combinatórios*. Instituto de Matemática Pura e Aplicada, Rio de Janeiro. Book.

Schaefer, T. J. (1978). On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16(2):185–225. DOI: 10.1016/0022-0000(78)90045-4.

Sibert, W. L. and Conway, J. H. (1992). Mathematical Kayles. *International Journal of Game Theory*, 20(3):237–246. Available at:`https://link.springer.com/article/10.1007/BF01253778`.

Sprague, R. (1935). Über Mathematische Kampfspiele. *Tohoku Mathematical Journal (First Series)*, 41:438–444. Available at `https://www.jstage.jst.go.jp/article/tmj1911/41/0/41_0_438/_article`.

Zermelo, V. E. (1912). Uber eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. *Proc. Fifth Congress Mathematicians*, 5:501–504. Available at:`http://webdocs.cs.ualberta.ca/~hayward/396/asn/zermelo.pdf`.