Kayles and nimbers*

Hans L. Bodlaender† Dieter Kratsch‡

Abstract

Kayles is a combinatorial game on graphs. Two players select alternatingly a vertex from a given graph G - a chosen vertex may not be adjacent or equal to an already chosen vertex. The last player that can select a vertex wins the game. The problem to determine which player has a winning strategy is known to be PSPACE-complete. Because of certain characteristics of the Kayles game, it can be analyzed with Sprague-Grundy theory. In this way, we can show that the problem is polynomial time solvable for graphs with a bounded asteroidal number. It is shown that the problem can be solved in $O(n^3)$ time on cocomparability graphs and circular arc graphs, and in $O(n^{1+1/\log 3}) = O(n^{1.631})$ time on cographs.

1 Introduction

For various reasons, games keep attracting the interest of researchers in mathematics and computer science. Games can provide for models, for instance for human thought processes, economic behavior, fault tolerance in computer systems, and computational complexity of machine models. Also, the analysis of games can provide for entertainment, and/or beautiful theory that is interesting on its own. It may be interesting to note that one of the first books written on graph theory [8] already contained a section on the relations between graphs and games.

*An earlier version of this paper appeared as [2]

†Department of Computer and Information Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.uu.nl

‡Université de Metz, Laboratoire d'Informatique Théorique et Appliquée, Île du Saulcy, 57045 Metz Cedex 01, France. Email: kratsch@lita.sciences.univ-metz.fr

In this paper we consider a combinatorial game, that is played on graphs, called *Kayles*. In this game, two players alternately choose a vertex from a given graph. Players may not choose a vertex that has been chosen before, and may also not choose a vertex that is adjacent to a vertex that has been chosen before. The last player that is able to choose a vertex wins the game.

The game can also described as follows: when a player chooses a vertex, this vertex and all its neighbors are removed from the graph. The first player that ends his move with the empty graph wins the game.

We consider the problem: given a graph $G = (V, E)$, does there exist a winning strategy for the first player when Kayles is played on G? We denote this problem also by the name *Kayles*. Kayles has been shown to be PSPACE-complete by Schaefer [11]. Despite its intractability for general graphs, Kayles has some nice characteristics, which together allow for efficient algorithms that solve some special cases. We remark that the game Kayles is:

- a two player game.
- finite. The game always ends after a finite number of moves, and each player can choose each time from a finite number of possible moves.
- full-information. There is no information that is hidden to one or both players, like e.g. in bridge, where cards of other players are unknown.
- deterministic. Every move gives rise to a unique position; no randomization devices (like dice) are used.
- impartial. This means that positions have no preference towards players. In other words, for each position, either the player that must move has a winning strategy, or the other player has — this is regardless of whether player 1 or player 2 must move from the position. For example, chess is not impartial, as there are white and black pieces owned by the players.
- with 'last player that moves wins the game' rule.

These six characteristics of Kayles make that it can be analyzed with help of Sprague-Grundy theory. Some readers may know this theory as the theory of the game Nim. In this theory, one associates to each position a (natural) number, here called nimber after [1]. (The position has nimber i, when it

can be represented by a stack of corresponding height in the game Nim.) It is possible to do some calculations with these nimbers, and determine which player has a winning strategy. In many cases, these calculations will be intractable, but — as will be shown in this paper — in some cases, they are not.

Those basic notions and results of Sprague-Grundy theory that are needed for this paper are reviewed in Section 3. For more background, we recommend the reader to consult [1] or [4]. Some graph theoretic definitions are given in Section 2.

2 Definitions

All graphs in this paper are considered to be finite, undirected and simple. For a graph $G = (V, E)$, and a subset of the vertices $W \subseteq V$, the subgraph of G, induced by W, is denoted by $G[W] = (W, \{(v, w) \in E \mid v, w \in W\})$. We denote $|V|$ by n, and $|E|$ by e. For $v \in V$, denote the set containing v and all neighbors of v by $N[v] = \{v\} \cup \{w \in V \mid (v, w) \in E\}$. For $X \subseteq V$, write $N[X] = \cup_{v \in X} N[v]$. Given a graph $G = (V, E)$ and a set of vertices $W \subseteq V$, we denote by $G - W$ the graph $G[V - W]$. For graphs $G_1 = (V_1, E_1)$, and $G_2 = (V_2, E_2)$, with disjoint vertex sets ($V_1 \cap V_2 = \emptyset$), we define the disjoint union $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ and the join $G_1 \times G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{\{v_1, v_2\} \mid v_1 \in V_1,\ v_2 \in V_2\}$.

Definition 1 *For a graph $G = (V, E)$, a set $W \subseteq V$ of vertices is* asteroidal*, if for every vertex $v \in W$, there is a connected component of $G - N[v]$ that contains all vertices of $W - \{v\}$. The* asteroidal number *of a graph G is the maximum size of an asteroidal set of G.*

Graphs that do not have an asteroidal triple (i.e., have asteroidal number two) are said to be AT-free. They are well studied, see e.g. [5]. See [7] for a study of the asteroidal number of graphs.

Definition 2 *A graph is a cograph, if and only if it can be formed by the following rules:*

1. *Every graph with one vertex and no edges is a cograph.*
2. *If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are disjoint cographs, then $G_1 \cup G_2$ and $G_1 \times G_2$ are cographs.*

To each cograph G, one can associate a labeled rooted tree T_G, called the *cotree* of G. Each leaf node of T_G corresponds to a (unique) vertex of V. Each internal node is labeled with either a 0 or a 1. Children of nodes labeled with 1 are labeled with 0, and vice versa. Two vertices are adjacent, if and only if their lowest common ancestor in the cotree is labeled with a 1. It is possible to associate a cograph with each node of the tree. Leaf nodes correspond to the cograph with the one vertex they represent. Internal nodes labeled with 0 (1) correspond to the disjoint union (join) of the cographs, corresponding to the children of the node. G equals the cograph corresponding with the root of T_G. Cographs can be recognized in $O(n + e)$ time, and in the same time the corresponding cotree can be built [6].

A graph is a cocomparability graph, if it is the complement of a comparability graph. A graph $G = (V, E)$ is a comparability graph, if it has a transitive orientation. Cocomparability graphs can be characterized as those graphs having a cocomparability ordering, i.e. an ordering $v_1, v_2, \ldots v_n$ of its vertices such that for all $i, j, k \in \{1, 2, \ldots, n\}$, $i < j < k$: $(v_i, v_k) \in E \Rightarrow (v_i, v_j) \in E$ or $(v_j, v_k) \in E$. See e.g. [9]. It seems worth noting the following correspondence between cocomparability orderings and linear extensions: Let P be a partially ordered set and let G be the complement of the comparability graph of G. Then any cocomparability ordering of G is a linear extension of P and vice versa.

3 Sprague-Grundy theory

In this section, we review some notions and results from Sprague-Grundy theory, and give some preliminary results on how this theory can be used for Kayles. For a good introduction to Sprague-Grundy theory, the reader is referred to [4] or the less formal and very entertaining [1].

A *nimber* is an integer $\in \mathbf{N} = \{0, 1, 2, \ldots\}$. For a finite set of nimbers $S \subseteq \mathbf{N}$, define the minimum excluded nimber of S as $mex(S) = \min\{i \in \mathbf{N} \mid i \notin S\}$.

We now assume that we consider positions in a two-player game, that is finite, deterministic, full-information, impartial, with 'last player wins'-rule. (As in Kayles.)

To each position in such a game, one can associate a nimber in the following way. If no move is possible in the position (and hence the player that must move loses the game), the position gets nimber 0. Otherwise the

nimber is the minimum excluded nimber of the set of nimbers of positions that can be reached in one move.

Theorem 3 *[1, 4] There is a winning strategy for player 1 from a position, if and only if the nimber of that position is at least 1.*

Denote the nimber of a position p by $nb(p)$. We next define the sum of two games. For (finite, deterministic, impartial, ...) games $\mathcal{G}_1$, $\mathcal{G}_2$, the sum of $\mathcal{G}_1$ and $\mathcal{G}_2$, $\mathcal{G}_1 + \mathcal{G}_2$ is the game, where each player when moving first decides whether he wants to make a move in $\mathcal{G}_1$ or in $\mathcal{G}_2$, and then selects a move in that game. The player that makes the last move (whether it is in $\mathcal{G}_1$ or $\mathcal{G}_2$) wins the game $\mathcal{G}_1 + \mathcal{G}_2$.

Definition 4 *Let $i_1, i_2 \in \mathbf{N}$ be nimbers. $i_1 \oplus i_2$ is the binary sum of i_1 and i_2 without carry, i.e., $i_1 \oplus i_2 = \sum\{2^j \mid (\lfloor i_1/2^j \rfloor \text{ is odd}) \Leftrightarrow (\lfloor i_2/2^j \rfloor \text{ is even})\}$.*

In other words, write i_1 and i_2 in binary notation. For every digit, take a 1 if either i_1 has a 1 for that digit, and i_2 has a 0 for that digit, or vice versa. For example $10 \oplus 7 = (8+2) \oplus (4+2+1) = 8+4+1 = 13$.

With (p_1, p_2) we denote the position in $\mathcal{G}_1 + \mathcal{G}_2$, where the position in $\mathcal{G}_i$ is p_i $(i = 1, 2)$.

Theorem 5 *[1, 4] Let p_1 be a position in $\mathcal{G}_1$, p_2 a position in $\mathcal{G}_2$. The nimber of position (p_1, p_2) in $\mathcal{G}_1 + \mathcal{G}_2$ equals $nb((p_1, p_2)) = nb(p_1) \oplus nb(p_2)$.*

As Kayles is an impartial, deterministic, finite, full-informtion, two-player game with the rule that the last player that moves wins the game, we can apply Spraque-Grundy theory to Kayles, and we can associate with every graph G the nimber of the start position of the game Kayles, played on G. We denote this nimber $nb(G)$, and call it the nimber of G.

An important observation is the following: when $G = G_1 \cup G_2$ for disjoint graphs G_1 and G_2, then the game Kayles, played on G is the sum of the game Kayles, played on G_1, and the game Kayles, played on G_2. Hence, by Theorem 5, we have the following result.

Lemma 6 $nb(G_1 \cup G_2) = nb(G_1) \oplus nb(G_2)$.

Note that G_1 and G_2 might be disconnected graphs.

Our second observation makes that it is sufficient to know the nimbers of certain subgraphs of the input graph G. Consider Kayles, played on $G = (V, E)$, and suppose that a vertex $v \in V$ is played. Then, the nimber of the resulting position is the same as the nimber of $G - N[v]$, as the effect of playing on v is the same as the effect of removing v and its neighbors from the graph. As the nimber of a position is the minimum nimber that is not in the set of nimbers of positions that can be reached in one move, we have:

Lemma 7 *(i) If $G = (V, E)$ is the empty graph, then $nb(G) = 0$.*
(ii) If $G = (V, E)$ is not the empty graph, then $nb(G) = mex(nb(\{G - N[v] \mid v \in V\})$.

The last two lemmas give a recursive algorithm to compute the nimber of a graph G, and hence, a method to decide which player has a winning strategy for Kayles on G. Using memoisation (i.e., storing values that are computed in a table, and looking up whether a value has been computed earlier) in a suitable way gives for special graph classes a polynomial time algorithm. This is explored in the next section.

4 A polynomial time algorithm for Kayles on graphs with bounded asteroidal number

In this section, we show that if the asteroidal number of a graph is bounded by a constant k, then Kayles can be solved in $O(n^{k+2})$ time. We extensively use results for solving the independent set problem on graphs with bounded asteroidal number from Broersma et al. [3]. We may assume that G is connected; otherwise we compute the nimber of every connected component of G and then apply Lemma 6. The following notion is from [3].

Definition 8 *Let $v \in V$ be a vertex in $G = (V, E)$. A* component *of v in G is a connected component in the graph $G - N[v]$. We number the components of v $C_{v,1}, \ldots, C_{v,s_v}$.*

The following notion is also from [3]. Intuitively, a lump is the set of vertices 'between' the vertices of the corresponding asteroidal set.

Definition 9 *Let A be an asteroidal set of at least two vertices in $G = (V, E)$. The* lump of A, *denoted by $L(A)$, is the set $\{v \in V \mid$ for all $x \in A$: there is a component $C_{x,i}$ of x containing v and $A - x\}$.*

Lemma 10 ([3]) *Let $C_{v,i}$ be a component of $v \in V$ in $G = (V, E)$, and let $w \in C_{v,i}$. Then*

$$G[C_{v,i} - N[w]] = G[L(\{v, w\})] \cup \bigcup_{C_{w,j} \subseteq C_{v,i} - N[w]} G[C_{w,j}]$$

From Lemmas 10 and 6, we directly have:

Corollary 11 *Let $C_{v,i}$ be a component of $v \in V$ in $G = (V, E)$, and let $w \in C_{v,i}$. Then*

$$nb(G[C_{v,i} - N[w]]) = nb(G[L(\{v, w\})]) \oplus \bigoplus_{C_{w,j} \subseteq C_{v,i} - N[w]} nb(G[C_{w,j}])$$

and thus, by Lemma 7, we have

Corollary 12 *Let $C_{v,i}$ be a component of $v \in V$ in $G = (V, E)$. Then*

$$nb(G[C_{v,i}]) = mex\{nb(G[L(\{v, w\})]) \oplus \bigoplus_{C_{w,j} \subseteq C_{v,i} - N[w]} nb(G[C_{w,j}]) \mid w \in C_{v,i}\}$$

With Corollary 12, we obtain a computational method for the nimber that is related to Lemma 6.2 from [3], where a similar formula for the size of the maximum independent set is given; the difference is mainly in the operands that are used.

In the same way as Lemma 10.8 from [3], we can obtain the following result.

Lemma 13 *Let A be an asteroidal set in $G = (V, E)$, and let $w \in L(A)$. Then*

$$\begin{aligned} &nb(G[L(A)]) = \\ &\quad mex_{w \in L(A)} \bigoplus_{C_{w,i} \cap A = \emptyset} nb(G[C_{w,i}]) \oplus \\ &\qquad \bigoplus_{C_{w,i} \cap A \neq \emptyset \wedge L(A) \cap C_{w,i} \neq \emptyset} nb(G[L(\{w\} \cup (C_{w,i} \cap A))]). \end{aligned}$$

The same computational method can be used as in [3], which basically consists of tabulating the values of $nb(C_{v,i})$ and $nb(G[L(A)])$, over all components $C_{v,i}$ (over all $v \in V$), and all asteroidal sets A (in a dynamic programming fashion.) The main difference is that here, we also need to compute the mex of several sets, but as there are $O(n^k)$ computations of a mex of a set with $O(n)$ elements, this does not affect the asymptotic running time. For the remaining details, we refer to [3]. Thus we arrive at an algorithm that uses the same running time as the algorithms in [3] for computing the independent set, independent dominating set, or independent perfect dominating set of a graph of asteroidal number at most k.

Theorem 14 *There is an algorithm that decides in $O(n^{k+2})$ time which player has a winning strategy for Kayles, played on a given graph $G = (V, E)$ with asteroidal number at most k, $n = |V|$.*

5 Kayles on cocomparability graphs

In this section, we give an algorithm that solves Kayles in $O(n^3)$ time on cocomparability graphs. As the class of cocomparability graphs contains several interesting subclasses, in particular the permutation graphs and the interval graphs, we get the same result for these classes too. Cocomparability graphs have asteroidal number at most 2.

Given a cocomparability graph, a cocomparability ordering, i.e. an ordering $v_1, v_2, \ldots v_n$ of its vertices such that for all $i, j, k \in \{1, 2, \ldots, n\}$, $i < j < k$: $(v_i, v_k) \in E \Rightarrow (v_i, v_j) \in E$ or $(v_j, v_k) \in E$, can be computed by an $O(n + e)$ time algorithm presented by McConnell and Spinrad in [10]. On the other hand, the best known recognition algorithms for cocomparability graphs have running time $O(n^{2.376})$ and $O(ne)$, respectively.

In order to describe our algorithm more easy, we add two isolated 'fake vertices', $s = v_0$, and $t = v_{n+1}$, to the graph G and its cocomparability ordering $v_1, v_2, \ldots v_n$. For all $v, w \in V \cup \{s, t\}$ with $v = v_i$, $w = v_k$ and $i < k$, we denote the set $V_{vw} = \{x \in V \mid x = v_j \text{ with } i < j < k\} - N[\{v, w\}]$. Clearly $G = G[V_{st}]$.

To compute the nimber of a graph $G[V_{vw}]$, we have to know the nimber of all subgraphs resulting from a play of one vertex in V_{vw}. The following lemma gives an easy characterization of such a resulting graph.

Lemma 15 *Let $x \in V_{vw}$. $G[V_{vw}] - N[x]$ is the disjoint union of $G[V_{vx}]$ and $G[V_{xw}]$.*

Proof: Clearly $V_{vw} - N[x]$ is the disjoint union of V_{vx} and V_{xw}. Now let $y \in V_{vx}$ and $z \in V_{xw}$. Then $(y, z) \in E$ implies $(y, x) \in E$ or $(x, z) \in E$, contradicting the fact that both y and z are not adjacent to x. $\square$

Lemma 15 also implies that for an asteroidal set $A = \{v, w\}$ of G its lump $L(A)$ is contained in V_{vw}. Notice that $G[V_{vw}]$ is not necessarily connected and that vertices of $V_{vw} - L(A)$ belong either to a component of v or to a component of w. Furthermore by Lemma 15, for any vertex $x \in V$ the graph $G - N[x] = G[V_{st}] - N[x]$ is the disjoint union of $G[V_{sx}]$ and $G[V_{xt}]$. Notice that both graphs $G[V_{sx}]$ and $G[V_{xt}]$ might be disconnected and that one of them might be empty.

From the discussion above, we see that the following dynamic programming algorithm solves the problem in $O(n^3)$ time. We compute all values $nb(G[V_{vw}])$ for all v, w in order of increasing value of $k - i$, where $v = v_i$ and $w = v_k$. Given a v, and w, the set $V_{vw} = \{x \in V \mid x = v_j, i < j < k \wedge \{v, x\} \notin E \wedge \{w, x\} \notin E\}$ can be computed in $O(n)$ time. If $V_{vw} = \emptyset$, then $nb(G[V_{vw}]) = 0$; otherwise, we compute $mex\{nb(G[V_{vx}]) \oplus nb(G[V_{xw}]) \mid x \in V_{vw}\}$ in $O(n^2)$. (The needed values $nb(G[V_{vx}])$ and $nb(G[V_{xw}])$ have been computed earlier and can be looked up.) Thus, the overall time of the procedure is $O(n^3)$.

Theorem 16 *There is an algorithm that decides in $O(n^3)$ time which player has a winning strategy for Kayles, played on a given cocomparability graph $G = (V, E)$.*

Thus, we also have an $O(n^3)$ algorithm for Kayles on subclasses of the cocomparability graphs like interval graphs, permutation graphs, etc.

One may observe that the result can be extended to some classes of graphs that have a circular intersection model, like the circular arc graphs. A straightforward algorithm using the fact that every graph $G - N[x]$ is an interval graph (permutation graph) if the graph itself is a circular-arc graph (circular permutation graph) would have running time $O(n^4)$. With an algorithm, similar to that for cocomparability graphs, we obtain the following result.

Theorem 17 *There is an algorithm that decides in $O(n^3)$ time which player has a winning strategy for Kayles, played on a given circular arc graph or a circular permutation graph $G = (V, E)$.*

6 Kayles on cographs

The class of cographs is a subclass of the class of cocomparability graphs, hence by the results of the previous section, there is an algorithm that solves Kayles on cographs in $O(n^3)$ time. However, with a more detailed analysis, we can obtain an algorithm with a better running time. The algorithm computes for every node in the cotree T_G of the input graph G, the set of nimbers of the positions reachable in one move from the cograph associated to the node.

Definition 18 *Let $G = (V, E)$ be a graph. The nimberset of G is the set of nimbers $nbs(G) = \{nb(G[V - N[v]] \mid v \in V\}$.*

Recall that $nb(G) = mex(nbs(G))$. We use the following notation: for a set of nimbers $S \subseteq N$, and a nimber α, we denote $\alpha \oplus S = \{\alpha \oplus \beta \mid \beta \in S\}$.

Lemma 19 *Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ be two disjoint graphs.*
(i) $nbs(G_1 \cup G_2) = nb(G_2) \oplus nbs(G_1) \cup nb(G_1) \oplus nbs(G_2)$.
(ii) $nbs(G_1 \times G_2) = nbs(G_1) \cup nbs(G_2)$.

Proof: (i) $nbs(G_1 \cup G_2) = \{nb(G_1[V_1 - N[v]] \cup G_2) \mid v \in V_1\} \cup \{nb(G_2[V_2 - N[v]] \cup G_1) \mid v \in V_2\} = nb(G_2) \oplus \{nb(G_1[V_1 - N[v]] \mid v \in V_1\} \cup nb(G_1) \oplus \{nb(G_2[V_2 - N[v]] \mid v \in V_2\} = nb(G_2) \oplus nbs(G_1) \cup nb(G_1) \oplus nbs(G_2)$.

(ii) Write $G = G_1 \times G_2$. $nbs(G) = \{nb(G[V_1 \cup V_2 - N[v]]) \mid v \in V_1\} \cup \{nb(G[V_1 \cup V_2 - N[v]]) \mid v \in V_2\} = \{nb(G_1[V_1 - N[v]]) \mid v \in V_1\} \cup \{nb(G_2[V_2 - N[v]]) \mid v \in V_2\} = nbs(G_1) \cup nbs(G_2)$. □

The lemma can be generalized as follows.

Lemma 20 *Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $\ldots$, $G_r = (V_r, E_r)$ be r disjoint graphs.*
(i) $nbs(G_1 \cup G_2 \cup \ldots \cup G_r) = \bigcup_{1 \leq i \leq r} nb(G_1) \oplus nb(G_2) \oplus \ldots \oplus nb(G_{i-1}) \oplus nb(G_{i+1}) \oplus \ldots \oplus nb(G_r) \oplus nbs(G_i)$.
(ii) $nbs(G_1 \times G_2 \times \ldots \times G_r) = nbs(G_1) \cup nbs(G_2) \cup \ldots \cup nbs(G_r)$.

The idea is to use these lemmas to compute for all internal nodes in T_G, the nimber and nimberset of the corresponding cograph. It is helpful for decreasing the running time of this computation, when we know what the maximum nimber is that a cograph with n vertices can attain.

Let $s(K)$ denote the minimum number of vertices of a cograph G with nimber at least 2^K. We will show that $s(K) = 3^K$.

Lemma 21 *For all $K \geq 0$, $s(K) \leq 3^K$.*

Proof: We give a series of cographs, H_0, H_1, H_2, ..., with H_K containing exactly 3^K vertices, and $nbs(H_K) = \{0, 1, 2, \ldots, 2^K - 1\}$, and hence $nb(H_K) = 2^K$.

For H_0, take a graph with one vertex and no edges. For $K \geq 1$, take $H_K = (H_{K-1} \cup H_{K-1}) \times H_{K-1}$. With Lemma 19, one easily verifies with induction that H_K fulfills the conditions mentioned above. □

Theorem 22 *For all $K \geq 0$, $s(K) = 3^K$.*

Proof: We now show that for all $K \geq 0$, $s(K) \geq 3^K$. As it is shown in Lemma 21 that $s(K) \leq 3^K$, Theorem 22 follows.

First observe that $s(0) = 1$, as the empty graph has nimber 0, and a graph with one vertex has nimber 1. The graph with two vertices and no edges has nimber 0, and the graph with two vertices and one edge has nimber 1. Hence $s(1) \geq 3$. We will now show that for all $K \geq 1$, $s(K+1) \geq 3 \cdot s(K)$. With induction, the result then follows.

Suppose $G = (V, E)$ is a cograph with nimber $\geq 2^{K+1}$, with minimum size of $n = |V|$. Let T_G be the corresponding cotree. We write G_i for the cograph, corresponding with node i in T_G.

Observe, that if for any node i in T_G, $2^{K+1} \in nbs(G_i)$, then G_i contains a subgraph with nimber 2^{K+1}. The size of this subgraph is smaller than the size of G, hence G was not of minimum size. Thus we may assume that $nbs(G) = \{0, 1, 2, \ldots, 2^{K+1} - 1\}$, and for all $i \in I$: $nbs(G_i) \subseteq \{0, 1, 2, \ldots, 2^{K+1} - 1\}$.

We say that a node $i \in T_G$ is *K-heavy*, if $nb(G_i) \geq 2^K$ or $nbs(G_i) \cap \{2^K, 2^K + 1, \ldots, 2^{K+1} - 1\} \neq \emptyset$. A node $i \in T_G$ is *K-precise*, if $nbs(G_i) = \{0, 1, 2, \ldots, 2^K - 1$. Note that for a K-precise node i, $nb(G_i) = 2^K$.

Claim 23 *If i_0 is K-heavy, then i_0 is K-precise, or i_0 has a descendant that is K-precise.*

Proof: From Lemma 20 it easily follows that every K-heavy node either is K-precise, or has a K-heavy child. □

As the root r of T_G is K-heavy, it follows that there must be at least one K-precise node in T_G. Note that if a K-precise node i has a descendant $j \neq i$ that is also a K-precise node, then G is not minimal: use the cograph

corresponding to the cotree, obtained by replacing the subtree rooted at i in T_G by the subtree rooted at j. So assume no K-precise node has a descendant which is also K-precise.

Claim 24 *There are at least two K-precise nodes in T_G.*

Proof: Suppose that i is the only K-precise node in T_G. Then the only K-heavy nodes in T_G are the nodes on the path from T_G to root r. With induction, one can prove that for each node j on this path, $\{0, 1, 2, \ldots, 2^K - 1\} \subseteq nbs(G_j)$. (Use Lemma 20, and note that only one term contains the binary factor 2^K.) Hence, if a predecessor j_0 of i is 1-labeled, it follows that for the unique K-heavy child j_1 of j_0, $nbs(G_{j_1}) = nbs(G_{j_0})$. Hence G was not of minimum size, contradiction.

So we may assume that i has exactly one predecessor, namely r, which is labeled with a 0. Hence we can write $G = G_i \cup H$. (H is the union of all cographs, corresponding to the other childs of r.) $nbs(G_i) = \{0, 1, 2, \ldots, 2^K - 1\}$, $nbs(H) \subseteq \{0, 1, 2, \ldots, 2^K - 1\}$. If $nb(H) < 2^K$, then $2^K + nb(H) \notin nbs(G_i \cup H)$, hence $nb(G) < 2^{K+1}$. So $nb(H) = 2^K$. Applying Lemma 19 it follows that $nbs(G_i \cup H) = \{2^K, 2^K + 1, 2^K + 2, \ldots, 2^{K+1} - 1\}$, contradiction. □

If there are at least three K-precise nodes, then note that each of these must have at least $s(K)$ leaf-descendants. Hence $s(K+1) \geq 3 \cdot s(K)$. Assume now there are exactly two K-precise nodes, i_0 and i_1. Let i_2 be the lowest common ancestor of i_0 and i_1. Similar as above, we can argue that G is not of minimum size, if a node between i_0 and i_2 or a node between i_1 and i_2 has a label 1, and if i_2 has label 1, then it has exactly two children, which are both K-heavy. (Each subtree rooted at one of these two children contains exactly one of i_0 and i_1, in this case.)

We consider now two cases, namely that i_2 has label 0, and that i_2 has label 1.

Case 1. i_2 has label 0. Then G_{i_2} can be written as $G_{i_2} = G_{i_0} \cup G_{i_1} \cup G_{j_1} \cup \ldots G_{j_r}$. Write $H = G_{j_1} \cup \ldots \cup G_{j_r}$. Note that $nbs(H) \subseteq \{0, 1, 2, \ldots, 2^K - 1\}$, as none of the nodes $j_1, \ldots, j_r$ is K-heavy. If $nb(H) = 2^K$, then $nbs(G_{i_2}) = \{0, 1, \ldots, 2^K - 1\}$ and i_2 is K-precise, contradiction. So $nb(H) < 2^K$, and it follows that $nbs(G_{i_2}) = \{2^K, 2^K + 1, \ldots, 2^{K+1} - 1\} \cup nbs(H)$.

Claim 25 *Let j be a node in T_G on the path from i_2 to r. Let H_j be the subgraph of G_j, obtained by removing all leaf-descendants of i_0 and of i_1 from G_j. Then $nbs(G_j) = nbs(H_j) \cup \{2^K, 2^K + 1, \ldots, 2^{K+1} - 1\}$.*

Proof: With induction. For $j = i_2$, the claim holds, as is argued above. Suppose the claim holds for the K-heavy child j' of j. Note that none of the other children of j is K-heavy. We must have that $nb(G_{j'}) = nb(H_{j'}) < 2^K$, otherwise $nb(G_{j'}) = 2^{K+1}$, which contradicts the minimality of G.

Write $G_j = G_{j'} \cup H$, or $G_j = G_{j'} \times H$. If j is labeled with a 0, then $H_j = H_{j'} \cup H$, and $nbs(G_j) = nb(G_{j'}) \oplus nbs(H) \cup nb(H) \oplus nbs(H_{j'}) \cup nb(H) \oplus \{2^K, 2^K+1, \ldots, 2^{K+1}\} = nb(H_{j'}) \oplus nbs(H) \cup nb(H) \oplus nbs(H_{j'}) \cup \{2^K, 2^K+1, \ldots, 2^{K+1}\} = nb(H_{j'} \cup H)\{2^K, 2^K+1, \ldots, 2^{K+1}\}$. If j is labeled with a 1, then $H_j = H_{j'} \times H$ and $nbs(G_j) = nbs(G_{j'}) \cup nbs(H) = nbs(H_{j'}) \cup \{2^K, 2^K+1, \ldots, 2^{K+1}-1\} \cup nbs(H) = nb(H_{j'} \times H) \cup \{2^K, 2^K+1, \ldots, 2^{K+1}\}$. □

In particular, we have for the root r of T_G that $\{0, 1, 2, \ldots, 2^{K+1}-1\} = nbs(G_r) = nbs(H_r) \cup \{2^K, 2^K+1, \ldots, 2^{K+1}-1\}$. Hence $nb(H_r) \geq 2^K$. So, if we remove all leaf-descendants of i_0 and i_1 from G, we remain with a graph with nimber at least 2^K, hence with a graph with at least $s(K)$ vertices. As both G_{i_0} and G_{i_1} contain at least $s(K)$ vertices, G contains at least $3 \cdot s(K)$ vertices. This ends the analysis of case 1.

Case 2. i_2 has label 1. Then G_{i_2} can be written as $G_{i_2} = (G_{i_0} \cup G_{j_1} \cup \ldots G_{j_r}) \times (G_{i_1} \cup G_{k_1} \cup \ldots G_{k_s})$. Write $H = G_{j_1} \cup \ldots G_{j_r}$, and $K = G_{k_1} \cup \ldots G_{k_r}$. Calculation shows that $nbs(G_{i_2}) = \{0, 1, 2, \ldots, 2^K-1\} \cup 2^K \oplus nbs(H) \cup 2^K \oplus nbs(K)$.

Consider the graph $G' = G_{i_0} \cup (H \times K)$. There are two cases:

Case 2.1. $nb(H \times K) = 2^K$. Then G contains at least $3 \cdot s(K)$ vertices: H and K together contain at least $s(K)$ vertices and are disjoint from G_{i_0} and G_{i_1}, which both also contain at least $s(K)$ vertices.

Case 2.2. $nb(H \times K) < 2^K$. As none of the nodes $j_1, \ldots, j_r,\ k_1, \ldots, k_s$ is K-heavy, it follows that $nbs(G') = nb(H \times K) \oplus nbs(G_{i_0}) \cup nb(G_{i_0}) \oplus nbs(H \times K) = nb(H \times K) \oplus \{0, 1, 2, \ldots, 2^K-1\} \cup 2^K \oplus nbs(H) \cup 2^K \oplus nbs(K) = nbs(G_{i_2})$. Now let G'' be the cograph, that corresponds to the cotree that is obtained by replacing in T_G the subtree rooted at i_2 by the cotree $T_{G'}$ of G'. The nimberset for i_2 does not change under this replacement operation, and hence $nb(G'') = nb(G)$. But G'' has fewer vertices than G, contradiction. This ends the proof of case 2, and of Theorem 22. □

Corollary 26 *For every cograph $G = (V, E)$ with n vertices, $nb(G) < 2n^{1/\log 3}$.*

Proof: Take the largest K with $2^K \leq nb(G)$. By Theorem 22, $n \geq 3^K$, hence $nb(G) < 2 \cdot 2^K = 2 \cdot 3^{(1/\log 3)K} \leq 2 \cdot n^{1/\log 3}$. □

Note that $1/\log 3 \approx 0.63093$.

We now give an algorithm that solves Kayles on cographs. We suppose that cograph G is given together with its cotree T_G. For each node i in T_G, let G_i denote the cograph corresponding with this node, and write $nb(i) = nb(G_i)$, and $nbs(i) = nbs(G_i)$. Let $z = \lfloor 2n^{1/\log 3} \rfloor$. To store $nbs(i)$ for each node i in T_G, associate with each node i in T_G a boolean array with entries for all nimbers $0,\ 1, \ldots,\ z$. The algorithm computes $nbs(i)$ and $nb(i)$ for all nodes i in T_G. For easier presentation, we state the algorithm as a recursive procedure, which is called with compute_nimbersets(r), with r the root of T_G.

```
procedure compute_nimbersets(node i):
begin if i is a leaf of T_G
        then begin nbs(i) := {0};
                   nb(i) := 1;
             end;
        else  begin Suppose the children of i are j_1, ..., j_r;
                   helpnbs := nbs(j_1);
                   helpnb := nb(j_1);
                   if label(i) = 0
                   then for k := 2 to r do
                          begin helpnbs := nb(j_k) ⊕ helpnbs ∪ helpnb ⊕ nbs(j_k);
                                helpnb := mex(helpnbs);
                          end;
                   else begin for k := 2 to r
                                do helpnbs := helpnbs ∪ nbs(j_k);
                                helpnb := mex(helpnbs);
                        end;
                   nbs(i) := helpnbs;
                   nb(i) := helpnb;
             end;
end.
```

Correctness follows from Lemmas 19 and 20. Taking the union of two sets of nimbers, and taking the $\oplus$-sum of a nimber and a set of nimbers can be done in $O(z)$ time. As T_G has n leaves, and hence $\leq n-1$ edges, a linear number of these operations is done. Hence, the total time of the algorithm is bounded by $O(nz) = O(n^{1+1/\log 3}) = O(n^{1.631})$.

Theorem 27 *Kayles can be solved on cographs in* $O(n^{1+1/\log 3}) = O(n^{1.631})$ *time.*

7 Final remarks

In this paper, we obtained polynomial time algorithms for Kayles, when restricted to several well studied classes of graphs. For several other interesting classes of graphs, the complexity of Kayles is still open. Probably the most notable of these classes is the class of trees. Already in 1978, Schaefer mentioned as an open problem the complexity of Kayles, when restricted to trees where only one vertex has degree at least three [11]. To the authors' best knowledge, this problem is still unresolved.

References

[1] Berlekamp, E. R., Conway, J. H., and Guy, R. K. *Winning Ways for your mathematical plays, Volume 1: Games in General.* Academic Press, New York, 1982.

[2] Bodlaender, H. L. Kayles on special classes of graphs — An application of Sprague-Grundy theory. In *Proceedings of the 18th International Workshop on Graph-Theoretic Concepts in Computer Science, WG'92* (1993), E. W. Mayr, Ed., Springer Verlag, Lecture Notes in Computer Science, volume 657, pp. 90–102.

[3] Broersma, H., Kloks, T., Kratsch, D., and Müller, H. Independent sets in asteroidal triple-free graphs. *SIAM J. Disc. Math. 12* (1999), 276–287.

[4] Conway, J. H. *On Numbers and Games.* Academic Press, London, 1976.

[5] Corneil, D. G., Olariu, S., and Stewart, L. Asteroidal triple-free graphs. *SIAM J. Disc. Math. 10* (1997), 399–430.

[6] Corneil, D. G., Perl, Y., and Stewart, L. K. A linear recognition algorithm for cographs. *SIAM J. Comput. 4* (1985), 926–934.

[7] Kloks, T., Kratsch, D., and Müller, H. Asteroidal sets in graphs. In *Proceedings 23nd International Workshop on Graph-Theoretic Concepts in Computer Science WG'97* (1997), R. Möhring, Ed., Springer Verlag, Lecture Notes in Computer Science, vol. 1335, pp. 229–241.

[8] König, D. *Theorie der endlichen und unendlichen Graphen.* Akademische Verlagsgesellschaft, Leipzig, 1936.

[9] Kratsch, D., and Stewart, L. K. Domination on cocomparability graphs. *SIAM J. Disc. Math. 6* (1993), 400–417.

[10] McConnell, R. M., and Spinrad, J. P. Modular decomposition and transitive orientation. *Disc. Math. 201* (1999) 189–241 (1999).

[11] Schaefer, T. J. On the complexity of some two-person perfect-information games. *J. Comp. Syst. Sc. 16* (1978), 185–225.