Finding a winning strategy in variations of Kayles

Simon Prins

ICA-3582809
Utrecht University, The Netherlands

July 15, 2015

Abstract

Kayles is a two player game played on a graph. The game can be defined as follows: both players take turns picking a node from a graph G. They remove that node and all its neighbors. When there are no more nodes left, the player whose turn it is loses. An alternate but equivalent way of phrasing it is to say that both players take turns picking an unmarked node from graph G and marking it and its neighbors. When all nodes are marked, the player whose turn it is loses. In this paper a variation of the game Kayles is studied, where instead of removing or marking just the neighbors of the chosen node, the players remove or mark the nodes at a certain distance d. Note that in this variation there is a difference between marking and removing a node. When the players remove nodes, the distance between other nodes can change, but when the players mark nodes the distances stay the same. This paper proves that Kayles with marking is PSPACE-complete and that Kayles with removing is PSPACE-complete for $d = 2$. Furthermore an $O^*(1.5875)$ algorithm for finding a winning strategy for Kayles with marking is given for distance $d = 2$. It is also shown that both versions of Kayles can be solved in polynomial time on certain classes of graphs.

1 Introduction

This paper studies two variations of the game Kayles. Kayles is a two player game played on a graph. The game can be defined as follows: both players take turns picking a node from a graph G. They remove that node and all its neighbors. When there are no more nodes left, the player whose turn it is loses. An alternate but equivalent way of phrasing it is to say that both players take turns picking an unmarked node from graph G and marking it and its neighbors. When all nodes are marked, the player whose turn it is loses.

The game of Kayles has been well studied. Schaefer proved it to be PSPACE-complete, see [1]. The use of Sprague-Grundy theory in finding winning strategies for Kayles has also been studied, see [2]. Bodlaender and Kratsch used this to show that Kayles can be solved in polynomial time on certain classes of graphs in [5]. In [4], Bodlaender et al. give an exponential time algorithm for solving it on general graphs.

In this paper a variation of the game Kayles is studied, where instead of removing or marking just the neighbors of the chosen node, the players remove or mark the nodes at a certain distance d. Although the two definitions given for Kayles are equivalent when removing only the neighbors of a node, they become different when the players remove nodes at a certain distance. When the players remove the nodes from the graph, the distance between two nodes u and v can increase. This happens when a node on a shortest path from u to v is removed. When marking nodes, the distance between nodes will never change, since the graph itself is never actually altered. The version of Generalized Kayles with Marking is denoted GKM_d, where d is the distance at which nodes are marked. Generalized Kayles with Removal is denoted GKR_d.

Although the two versions of Kayles are not generally the same, they can be equivalent on certain graphs. Let $N_d(v)$ denote the set of all vertices at distance at most d from v.Suppose a graph G has the property that removing any vertex v and all vertices $N_d(v)$ will not increase the distance between any two vertices from being smaller then or equal to d to being larger than d. Suppose that all subgraphs $G - N_d(v)$ that can be obtained by picking any vertex v and removing it and all its neighbors at distance d or less, recursively share this property. Denote such a graph d-Kayles equivalent. The following theorem holds.

Theorem 1. *A game of GKM_d that is played on a d-Kayles equivalent graph G is equivalent to a game of GKR_d played on the same graph.*

Consider a game of GKM_d, played on a graph G. The dth powergraph of G is the graph with the same nodes as G, and where there is an edge between every pair of nodes u, v iff the distance between u and v is at most d in G. Note that the following theorem holds:

Theorem 2. *A game of GKM_d that is played on graph G is equivalent to a game of regular Kayles played on the d-th powergraph of G.*

This provides an easy way to reuse some of the results obtained for Kayles. Particularly, the algorithm described by Bodlaender et. al. in [4] for determining if a winning strategy exists for Kayles can be used in the following way. Given a game of GKM_d, played on graph G. The algorithm for regular Kayles described by Bodlaender can be used to determine if a winning strategy exists on the dth powergraph of G. If such a winning strategy exists, then there also exists a winning strategy for GKM_d on graph G, if it does not, then no winning strategy is possible for GKM_d either. Since the powergraph of a graph can be computed in polynomial time, and the algorithm for regular graphs can be run in $O^*(1.6031^n)$ time, this gives an $O^*(1.6031^n)$ algorithm for finding a winning strategy for GKM_d. In Section 4 an improved upper bound of $O^*(1.5875)$ is given for GKM_2.

In Section 2 some preliminary results for the regular version of Kayles are provided. These results are also useful for the variations of Kayles studied in this paper. In Section 3 the complexity of the generalized versions of Kayles is studied. The section provides a proof that GKM_d is PSPACE-complete. It also proves that GKR_2 is PSPACE-complete. Section 4 gives an exact algorithm for finding a winning strategy for GKM_2 on arbitrary graphs. Section 5 provides some polynomial time algorithms for solving both versions of Generalized Kayles on special classes of graphs. Finally, Section 6 provides an overview of the conclusions that can be drawn from this paper.

2 Preliminaries

2.1 PSPACE-completeness of regular Kayles

Schaefer proves Kayles to be PSPACE-complete in [1]. In this section his proof is given. In Section 3, the graph given in this proof is extended, to prove GKM_d to be PSPACE-complete. It is also used in the proof that GKR_2 is PSPACE-complete.

Schaefer gives a reduction from the game $G_\omega(CNF)$ to Kayles.

Definition 1. He defines $G_\omega(CNF)$ as follows: $G_\omega(CNF)$ is a two player game, where the input is a pair (A, ξ) where A is a CNF-formula and $\xi = (\xi_1, \xi_2, ..., \xi_n)$ is a list of distinct variables, including all those occurring in A. Move i consists of assigning to ξ_i a value of 0 (false) or 1 (true). After n moves, [the first player] wins iff the assignment which has been produced makes A true.

$G_\omega(CNF)$ is known to be PSPACE-complete, even when restricting it so that n must be odd and the first clause of A must be $(x_1 \vee \neg x_1)$.

Next is shown that this restricted version of $G_\omega(CNF)$ can be reduced to an instance of a game of Generalized Kayles with Marking, for any even d. An example of this reduction for the formula $(x_1 \vee \neg x_1) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$ is shown in Figure 1.

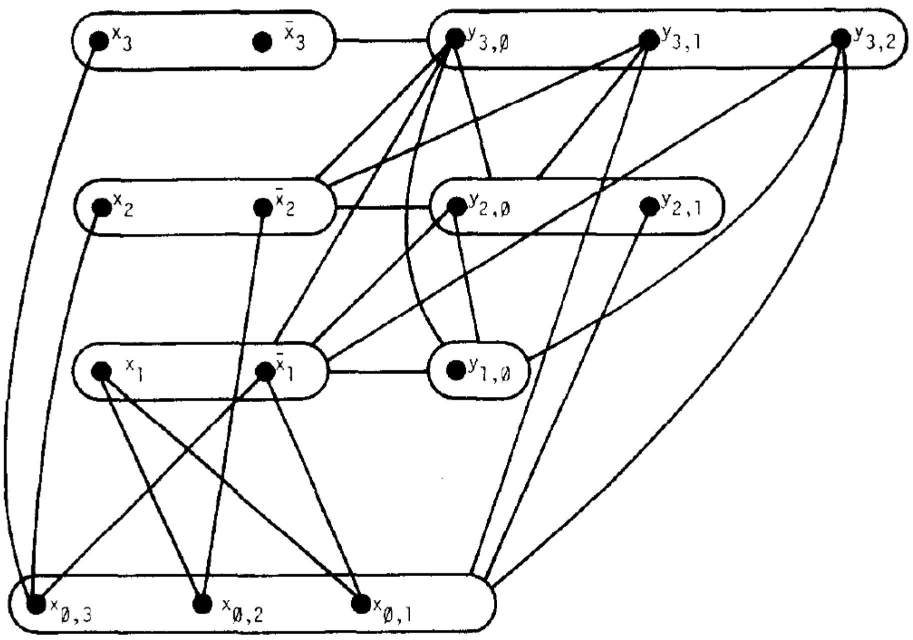

Figure 1: A reduction from $G_\omega(CNF)$ to Kayles for the formula $(x_1 \vee \neg x_1) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$. Source: *On the Complexity of Some Two-Person Perfect-Information Games*, by Schaefer [1].

First the graph G is constructed. This is the same graph that Schaefer constructs is his proof. Let $A = B_1 \wedge B_2 \wedge ... \wedge B_m$, with each B_i a disjunction of literals. For every B_i a node $x_{0,i}$ is added to the graph. For each variable ξ_i a node x_i and a node $\bar{x}_i$ are added to the graph. Finally nodes $y_{i,0}, y_{i,1}, ..., y_{i,i-1}$ are added to the graph for each variable ξ_i. There is an edge between a node x_i and a node $x_{0,j}$ if B_j contains the non-negated literal ξ_i. There is an edge between a node $\bar{x}_i$ and a node $x_{0,j}$ if B_j contains the negated literal ξ_i. Call the nodes $x_{0,1}$ to $x_{0,m} row0$. Call the nodes x_i and $\bar{x}_i$ and all the nodes $y_{i,0}, y_{i,1}, ..., y_{i,i-1}$ *row i*. Additional edges are added to each row to form a clique. There is an edge between a node $y_{i,j}$ and every node in the rows 0 to $i-1$ except for row j.

There exists a winning strategy for regular Kayles on this graph iff there exists a winning strategy for the $G_\omega(CNF)$ game. The idea is as follows. On turn i, the player whose turn it is, is supposed to pick either node x_{n-i+1} or node $\bar{x}_{n-i+1}$. This is called a legitimate move. This corresponds to assigning the value true or the value false respectively to variable ξ_i. Note that the rules of Kayles allow the player to play on any other node as well. This will be addressed later. Note that since each row is a clique, the other nodes in the row will disappear as well. After n turns, if the corresponding assignment of values to the variables makes A true, then there are no more nodes in row 0 and, since n is odd, it is the second players turn and the first player wins. If the assignment of variables does not make A true, then some nodes will remain. The second player can pick any of these nodes. Since only nodes on row 0 remain, all of them will be removed. This means the second player wins the game.

If a player does not play legitimately, i.e. on turn i, he picks some node other than x_{n-i+1} or $\bar{x}_{n-i+1}$, then the other player can make a move in such a way that all remaining nodes are removed and he wins. When a player picks a node in row j with $j < n - i + 1$, then node $y_{i,j}$ will remain. It is connected to all nodes except for those on row j. Therefore if one player plays illegitimately on row j, then the other player can pick this node and win. If the player played illegitimately on a node in row $n - i + 1$, i.e. he picked a node $y_{n-i+1,j}$, then all nodes will be removed except those on row j. The other player can pick any of these node to win. This means that if there is a winning strategy for Kayles on this graph, it can only contain legitimate moves, thus there must be a winning strategy in the $G_\omega(CNF)$ game.

2.2 Sprague-Grundy theory

Sprague-Grundy theory provides results that can help analyzing positions in two player games that are finite, deterministic, full-information, impartial, and with the 'last player wins rule'. Kayles and the variations of Kayles discussed in this paper all fall into this category of games. Sprague-Grundy theory allows us to assign a nimber to any position in such a game. A nimber is an integer belonging to the set $\mathbb{N} = \{0, 1, 2, ...\}$. A nimber is defined as 0 if there is no move possible. Otherwise it is defined as the minimum excluded nimber of the set of nimbers of positions reachable in one move. The minimum excluded nimber of a set S is defined as $mex(S) = min\{i \in \mathbb{N} | i \notin S\}$. The following theorem helps us determine whether a player has a winning strategy:

Theorem 3. *[2] [3] There is a winning strategy for player 1 from a position,* *if and only if the nimber of that position is at least 1.*

Another theorem allows us to combine two (finite, impartial, deterministic...) games and get a new nimber for the resulting game. Denote the nimber of a position p by $nb(p)$. Given two games G_1 and G_2, the sum of G_1 and G_2, denoted $G_1 + G_2$ is the game where a move consists of choosing G_1 or G_2 and making a move in that game. A player that cannot make a move in either game looses the game $G_1 + G_2$. (p_1, p_2) denotes the position in $G_1 + G_2$ where the position in G_i is p_i.

The binary XOR operation is denoted by $\oplus$, i.e., $i_1 \oplus i_2 = \sum\{2^j | (\lfloor i_1/2^j \rfloor$ is odd$) \leftrightarrow (\lfloor i_2/2^j \rfloor$ is even$)\}$ for nimbers i_1, i_2.

Theorem 4. *[2] [3] Let p_1 be a position in G_1, p_2 be a position in G_2. The nimber of position (p_1, p_2) in $G_1 + G_2$ equals $nb((p_1, p_2)) = nb(p_1) \oplus nb(p_2)$.*

Both Kayles with Marking and Kayles with Removal are impartial, deterministic, finite, full-information, two player games with the rule that the last player that moves wins the game. This means that Sprague-Grundy theory can be applied to them and a nimber can be associated with every position in either game. For Kayles with Removal this works as follows. A nimber can be associated with every graph G, the nimber of the start position of the game Kayles with Removal played on G. This nimber is denoted $nb(G)$, and it is called the nimber of G. Suppose G_1 and G_2 are two disjoint graphs and $G = G_1 \cup G_2$. The game of Kayles with Removal played on G is the sum of the same game played on G_1 and G_2. Hence the next lemma follows from Theorem 4:

Lemma 1. *$nb(G_1 \cup G_2) = nb(G_1) \oplus nb(G_2)$.*

Consider a game of Kayles with Removal played on graph $G = (V, E)$. Suppose a node $v \in V$ is played. Then the nimber of the resulting position is the same as the nimber of $G - N_d[v]$, as its effect is the same as removing v, and all nodes at distance at most d from v. 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, the following can be concluded:

Lemma 2. *(i) If $G = (V, E)$ is the empty graph, then $nb(G) = 0$.*

(ii) If $G = (V, E)$ is not the empty graph, then $nb(G) = mex(nb(\{G - N_d[v] | v \in V\}))$.

For Kayles with Marking, determining the nimber works slightly different. If two unmarked regions of the graph become disconnected, there may still be a node in one region that is at distance at most d from another region.

In this case the two regions cannot be considered as separate games, since a move in one region could influence the other region. However, a position $p = (G, X)$ can be defined, where G is the original graph on which the game is played, and X is the set of vertices which remain unmarked at position p. Now $p_1 = (G, X_1)$ and $p_2 = (G, X_2)$ are two positions on the same graph. Suppose that X_1 and X_2 are distinct and no node from one set has a neighbor in the other set in G^d. A move on a node in X_1 can no longer influence the nodes in X_2. The position $p_3 = (G, X_1 \cup X_2)$ is therefore the sum of the positions p_1 and p_2. Hence by Theorem 4, the following lemma is obtained:

Lemma 3. *$nb(p_3) = nb(p_1) \oplus nb(p_2)$.*

Consider a position (G, X) in the game of Kayles with Marking, with $G = (V, E)$. Suppose a node $v \in X$ is played. Then the nimber of the resulting position is the same as the nimber of $(G, X - N_d[v])$, as its effect is the same as removing v, and all nodes at distance at most d from v. 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, the next lemma follows:

Lemma 4. *(i) If $p = (G, \emptyset)$ then $nb(p) = 0$.*

(ii) If $p = (G, X)$ and X is not the empty set, then $nb(p) = mex(nb(\{G - N_d[v] | v \in X\}))$.

3 Complexity of Generalized Kayles

Regular Kayles has been proven PSPACE-complete by Schaefer in [1]. In this section it is proven that Generalized Kayles with Marking is PSPACE-complete as well. It is also proven that Generalized Kayles with Removal is PSPACE-complete for $d = 2$.

3.1 Generalized Kayles with marking

In this section the following theorem is proven, based on the proof for regular Kayles by Schaefer in [1].

Theorem 5. *Kayles with Marking is PSPACE-complete for any d.*

Proof. In his proof that regular Kayles is PSPACE-complete, Schaefer gives a reduction from the game $G_\omega(CNF)$ to Kayles. This proof is the basis for the following proof.

Here, also a reduction from $G_\omega(CNF)$ is used. First, construct the graph G from $G_\omega(CNF)$, as described in Section 2.1.

Graph G' is constructed by replacing each node in G with three nodes in G'. These three nodes form a clique and whenever there is an edge between two nodes v and w in G, there is an edge between every copy of v in G' and every copy of w in G'.

The proof that regular Kayles is PSPACE-complete would still work the same on this graph, only now whenever a player makes a move, he can choose any of the three copied nodes.

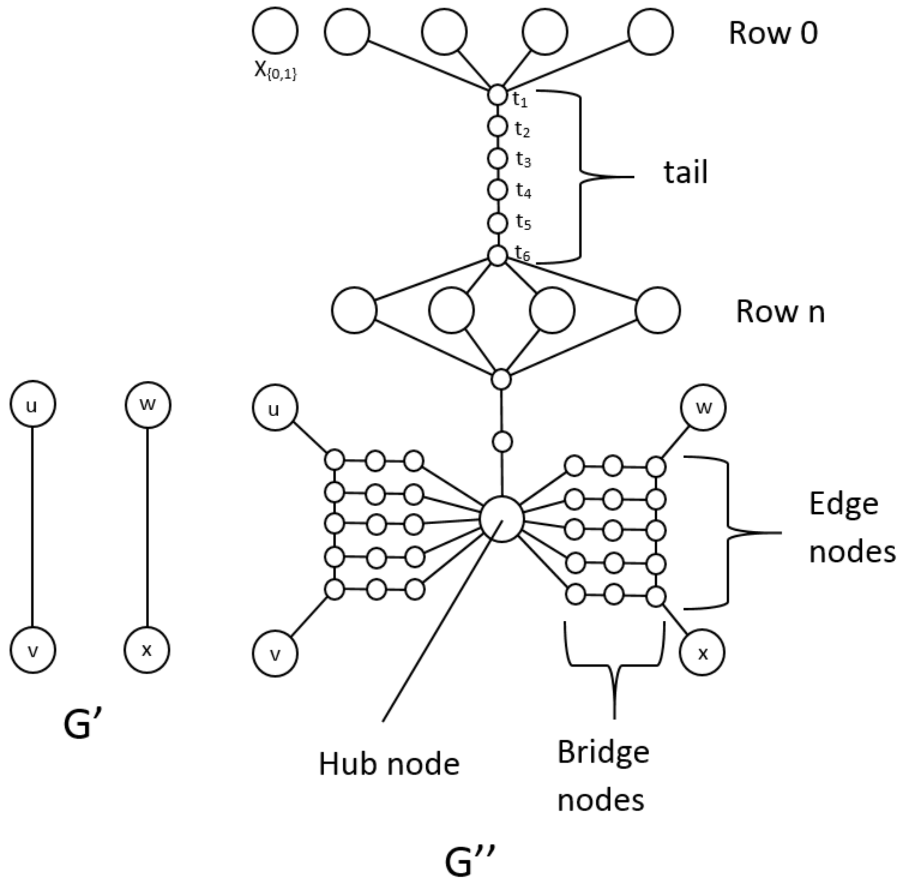

Figure 2: The transformation of a part of graph G' to G'' for $d = 6$.

Now the graph G'' is constructed from the graph G'. Figure 2 shows a part of a graph G' and the resulting part of the graph G'' for $d = 6$.

Each edge vw in G' is replaced with a string of $d-1$ nodes. Each of these nodes is connected to the next and the first and last nodes are connected to v and w respectively. The distance between neighboring nodes from the original graph is now exactly d. These added nodes are called edge nodes.

Each edge node is connected to a single hub node by a string of $\lfloor d/2 \rfloor - 1$ nodes, called bridge nodes. For $d = 2$ and $d = 3$, each edge node is connected directly to the hub node. The hub node has another string of $\lceil d/2 \rceil - 1$ nodes attached to it. The last of these is attached to every node in row n. For $d = 2$, the nodes in row n are directly connected to the hub node. There is also a string of d nodes, labeled t_1 to t_d. t_d is connected to each of the nodes in row n, and t_1 is connected to each node in row 0 except for $x_{0,1}$. These nodes are called the tail.

The idea is that the first player, on his first turn, will pick one of the nodes on the first row. This node is at distance d or less from every other node that was added in G''. This means that after the first turn, only nodes from G' will remain unmarked and the game will continue like the original game on graph G.

Suppose the first move is on one of the original nodes of Graph G, but not in the first row. The opponent can now pick the corresponding counter move in the first row. This marks all remaining nodes.

Suppose the first move is to pick a node at distance $\lceil d/2 \rceil - 1$ or less from the hub node. All original nodes are at distance $\lfloor d/2 \rfloor + 1$ from the hub, so they are marked. Only one or more nodes of the tail will remain unmarked. Picking one of these will mark all the remaining nodes, so the first player loses.

Suppose the first move is to pick an edge node. If d is odd, then the edge node is at distance $\lceil d/2 \rceil - 1$ from the hub node and the previous case applies. If d is even, then these nodes are at distance $d/2$ from the hub. This means that all edge nodes are marked. Also, the entire nth row is marked. The original nodes will remain, except for row n and the two nodes linked by the picked edge node. Since there are three of each node, the game played on the regular nodes still works the same. The second player can now pick the tail node adjacent to row 0. This will mark row 0, except for node $x_{0,1}$. It will also remove all the nodes that were added, so that the remaining game works the same as the game on G. However, in this new game, the players have swapped turns. The second player is trying to make the bottom row empty and the first player is trying to prevent this. The only node remaining in the bottom row is node $x_{0,1}$ which is connected to both x_1 and $\bar{x}_1$. The second player will always win the game if he plays it out as normal, so the first player has no winning strategy.

Suppose that the first move is to pick t_1. The distance from this node the the nodes of row n is exactly d. The distance from this node to the nodes of row 0 is 1. The distance to the hub node is $\lfloor d/2 \rfloor + 2$. Since there are $\lceil d/2 \rceil - 1$ nodes connecting the hub to row n, exactly one of those will remain. Picking this node will mark every remaining node. The only exception is when $d = 2$. In this case the hub node is not marked. It can be picked to mark all remaining nodes.

If the first move is to pick one of the nodes t_2 to $t_{\lceil d/2\rceil -1}$ there is always a bridge node that can be picked to mark all remaining nodes.

If the first move is to pick node $t_{\lceil d/2\rceil}$, then the hub will remain and picking this marks all remaining nodes.

If the first move is to pick on of the nodes $t_{\lceil d/2\rceil +1}$ to t_{d-1}, picking any bridge node will remove all remaining nodes.

If the first move is to pick node t_d, then the second player can pick any edge node. If d is odd, then all nodes are now marked. If d is even, then the game will continue again with the players having swapped turns. Row n is removed and row 0 is removed, except for node $x_{0,1}$. The second player can win this by playing the game out normally.

In conclusion, any illegitimate first move will result in a loss for the first player. A winning strategy is only possible if the game is played out as described by Schaefer. This means that if a winning strategy can be found for the first player, there also exists a winning strategy in $G_\omega(CNF)$. □

3.2 Generalized Kayles with Removal

In this section a proof is given for the following theorem, based on the proof for regular Kayles by Schaefer.:

Theorem 6. *Kayles with Removal is PSPACE-complete for $d = 2$.*

Proof. First, construct the graph G from $G_\omega(CNF)$, as described in Section 2.1.

The set of nodes for graph G' as defined as follows:

$V' = \bigcup_{i=0}^{n} X'_i$

$X'_0 = \{x_{0,k,1} | 1 \leq k \leq m\} \cup \{x_{0,k,2} | 1 \leq k \leq m\}$

$X'_i = \{x_i, \bar{x}_i\} \cup \{y_{i,j,1} | 0 \leq j \leq i-1\} \cup \{y_{i,j,2} | 0 \leq j \leq i-1 \wedge \neg(i=1)\}$

The edges are constructed as follows:

$E' = \bigcup_{i=0}^{n} [X'_i]^2 \cup D' \cup \bigcup_{i=1}^{n} \bigcup_{j=0}^{i-1} C'_{i,j}$

$D' = \{\{x_i, x_{0,k,j}\} | x_i$ occurs unnegated in $B_k, j \in \{1,2\}\} \cup \{\{\bar{x}_i, x_{0,k,j}\} | x_i$ occurs negated in $B_k, j \in \{1,2\}\}$

$C'_{ij} = \{\{y_{i,j,1}, w\}, \{y_{i,j,2}, w\} | w \in \bigcup_{\substack{0<=k<i \\ k \neq j}} X_k\}$

This has the effect of duplicating all nodes $y_{i,j}$ except for $y_{1,0}$ and all nodes $x_{0,i}$. The proof that Kayles is PSPACE-complete would still work on this graph, only now when a player makes an illegitimate move, the other player can choose which of the two y nodes to take. Duplicating the nodes from X_0 can be viewed as duplicating the clauses of the formula, which has no effect on the existence of a winning strategy. Adding these nodes ensures that there are always an odd number of nodes in the graph, which will be used later.

Now the graph G'' is constructed from the graph G'. This is done by replacing each edge uv by the edges $u - x_{uv1}$, $u - x_{uv2}$, $v - x_{uv1}$, $v - x_{uv2}$. Note that we have also added two nodes x_{uv1} and x_{uv2} per edge. These nodes are referred to as edge nodes. A node z_i is also added for every variable i of the CNF game. These nodes are all connected to each other. They are also connected to all edge nodes between nodes from X_0 to X_i. They are also connected to edge nodes between each pair $y_{j,k,1}$, $y_{j,k,2}$ with k unequal to zero. Finally nodes $z_{0,1}$ and $z_{0,2}$ are added. These are connected to all the nodes of X_1. $z_{0,2}$ is also connected to all edge nodes connected to two nodes in X_1.

To show that this reduces an instance of a CNF game to an instance of a 2-Kayles game, it is first shown that if each player only plays legal moves, i.e. on turn i, he only plays either x_{n-i+1} or $\bar{x}_{n-i+1}$, a winning play in the instance of 2-Kayles is equivalent to a winning play in the CNF game. It is also shown that if a player plays any illegitimate move, he automatically loses.

When only legitimate moves are played, on each move i node z_{n-i+1} will be removed, since that node is connected to the edge nodes between x_{n-i+1} and $\bar{x}_{n-i+1}$ and therefore has distance 2 to each of those nodes. On the nth turn, $z_{0,1}$ and $z_{0,2}$ will also be removed. Finally, all nodes that have been added will get removed, except possibly for some edge nodes. However these come in pairs. From Lemma 1 it follows that pairs of identical graph components sum to zero, so they do not influence the existence of a winning strategy, and can be ignored.

Case 1: A player plays illegitimately on one of the nodes of graph G If a player plays illegitimately on one of the nodes from graph G', this move can be countered by playing the corresponding move given in the proof by Schaefer. If the player picks a node in X_0, then z_1 to z_n will be removed, since they are connected to all edge nodes between nodes in X_0. If the player picks a node from X_1 to X_{n-1} the corresponding move is to pick a certain y node. The edge nodes between y nodes are connected to z_1 to z_n, so again they will be removed. If the player picks an illegitimate node in X_n, the next player picks a node from the corresponding set X_i, which is the reverse of one of the previously mentioned cases, so again all nodes z_1 to z_n will be removed. If one of the nodes in X_1 gets picked, then the nodes $z_{0,1}$ and $z_{0,2}$ will both be removed, otherwise they will both remain.

The final result is that all nodes from the graph G' are gone and that an even number of disconnected nodes remain. This means that the player who played illegitimately will lose.

Case 2: A player plays illegitimately on one of the edge nodes Suppose now that a player plays illegitimately on one of the edge nodes. Note: it is possible that some edge nodes are no longer connected to one of their endpoints, because it was previously removed. Such an edge node will not be connected to any of the z nodes.

Case 2.1: A player plays illegitimately on one of the edge nodes still connected to both its endpoints First assume that the edge node that was picked is still connected to both its endpoints. If this is turn i, then node z_{n-i+1} is connected to all edge nodes, which means that picking an edge node will remove all other edge nodes. Since all nodes z_1 to z_n are connected, they also get removed, as well as the endpoints of the edge node that was picked. This means that, except for the endpoints of the edge, all original nodes still remain, but are disconnected. Some edge nodes may also remain, but they are all disconnected and appear in pairs. If one of the endpoints was a node in X_1 then $z_{0,1}$ and $z_{0,2}$ are also removed. Since the graph G' was manipulated to have an odd number of nodes, there are now an odd number of disconnected nodes left. This means that the player who picked the edge node will lose. Suppose that neither of the endpoints of the chosen edge were in X_1. Then $z_{0,1}$ and $z_{0,2}$ remain. Picking either of these will remove both of them and also the set X_1. Since there are an odd number of nodes in X_1, there is an even number of disconnected nodes left and the player who played illegitimately will lose.

Case 2.2: A player plays illegitimately on one of the edge nodes of which one of its endpoints has been removed Suppose that one (but not both) of the endpoints of the edge node has been removed. This means that selecting one of these nodes will make the total number of nodes in the graph odd. If the next player now takes an edge node connected to two nodes in X_1, there are an even number of disconnected nodes left and the player who played illegitimately loses.

Case 3.1: A player plays illegitimately on node z_i on the $n-i+1$th turn Suppose that a player illegitimately picks node z_i on the $n-i+1$th turn. This node is connected to all edge nodes, so all edge nodes are removed. All nodes from G' are directly connected to at least one edge node, so they also get removed. $z_{0,2}$ is also connected to some edge nodes, so it also gets removed. $z_{0,1}$, however is not removed. When the next player picks node $z_{0,1}$, no nodes remain except possibly some pairs of edge nodes. This means that the player who played illegitimately loses.

Case 3.2: A player plays illegitimately on node z_j, not on the

$n - i + 1$th turn, where $i \neq j$ Suppose now, that a player picks a node z_j and it is not the $n - j + 1$th turn. All nodes from X_1 to X_j and the edge nodes between them are removed. All nodes z_1 to z_i are removed as well, where i is the number of the current turn. Note that also the edge nodes in X_j to X_i are removed, since they are connected to z_i. $z_{0,2}$ is also removed, however, $z_{0,1}$ is not. This means that an odd number of nodes remains and the player who played illegitimately loses.

Case 4: A player plays illegitimately on node $z_{0,1}$ or $z_{0,2}$ Suppose the player illegitimately picks $z_{0,1}$ or $z_{0,2}$, then they are both removed. Also, X_1 is removed. This means that picking $y_{i,1}$ will remove all remaining nodes. If the nodes z_1 to z_i are not removed because $z_{0,1}$ was picked and not $z_{0,2}$, then they are removed now. Only some edge node pairs may remain, which means that the player who played illegitimately loses. □

4 An algorithm for solving Kayles with Marking on Arbitrary Graphs

In the previous section it was shown that Generalized Kayles with Marking is PSPACE-complete. This means that it is unlikely that there exists a polynomial time algorithm for determining a winning strategy. In this section an exponential time algorithm is given when $d = 2$. First consider the following algorithm: compute a nimber for every position $p = (G, X)$, where $G = (V, E)$ is the original graph and X is some subset of V. If these nimbers are computed in increasing order of the size of X, then the nimber for each position can be computed in polynomial time using Lemmas 3 and 4. There are $O(2^n)$ subsets of V, and therefore this algorithm runs in $O^*(2^n)$ time.

Now an algorithm with a better running time is given.

```
compute_nimber($G$, $X$, $d$)
if $nb(X)$ already computed then
    return nb($X$);
else
    M := $\emptyset$;
    for All $v \in W$ do
        let $Z_1, Z_2, ..., Z_r (r \geq 1)$ be the sets of vertices in the
        components of $G[X]^d - N^d(v)$;
        nim := 0;
        for $i \leftarrow 1$ to $r$ do
            nim := nim $\oplus$ compute_nimber($G$, $Z_i$, $d$);
        end
        M := M $\cup$ nim;
    end
    return mex(M);
end
```

Algorithm 1: Procedure compute_nimber

Theorem 7. *The above algorithm runs in $O^*(1.5875)$ when $d = 2$.*

Proof. The algorithm computes a nimber for every position that can be reached while playing the game. When the nimber of a position can be computed from smaller components of the graph using Lemma 3, it does so. The algorithm computes a nimber only once for every reachable position and it does this in polynomial time. The following theorem gives an upper bound on the number of such reachable positions. Theorem 7 therefore follows from the next theorem.

Theorem 8. *The number of reachable positions in Kayles with Marking is bounded by $O^*(1.5875)$ for $d = 2$.*

The proof for this theorem is algorithmic: we give a branching procedure that generates all reachable positions. By distinguishing different types of vertices and assigning these different weights, and considering the different branching vectors, we obtain a set of recurrences, whose solution gives the desired upper bound. For information on branching algorithms and their analysis, in particular branching vectors and the corresponding recurrences we refer to [6].

We say that a set of nodes X is nontrivial if $|X| \geq 3$; otherwise we call it trivial. There are at most $O(n^2)$ trivial sets. During our branching process, we decide at some points to put some vertices in a set X and forbid for some vertices to put them in X. Placing a vertex in X means we consider all positions that can be reached when either player at some point picks that vertex. When placing a vertex in X, we say we select the vertex. The vertices in G are of six types:

- **White** or free vertices. Originally all vertices in G are white. We have not made any decision yet for a white vertex. All white vertices have weight one.
- **Red** vertices. Red vertices may not be placed in the independent set X: i.e..we already decided this during the branching. It is still possible that a red vertex becomes deleted later, however. Red vertices have to be neighbors of white vertices. Red vertices have a weight α.
- **Blue** vertices. Blue vertices may not be placed in the independent set X: i.e..we already decided this during the branching. It is still possible that a blue vertex becomes deleted later, however. A blue vertex may not be the neighbor of a white vertex. blue vertices have a weight β.
- **Green** vertices. A green vertex is 'safe': it never will be removed. I.e., we cannot place the green vertex nor any vertex at distance at most d from the green vertex in the independent set X. Green vertices have weight zero.
- **Black vertices**: when a vertex is placed in the independent set, any vertex at a distance of exactly two will become black. This is a vertex that will be deleted when the algorithm is done, however it is left as a black vertex since it may still influence other vertices. In particular, when a white neighbor of a black vertex is placed in X, any other neighbor of the black vertex is also at distance two and will become black as well. Black vertices have a weight of γ.
- **Removed vertices**: these are vertices that will never become green and have no influence on the rest of the game. They either have been placed in the independent set or are the neighbor of a vertex in the independent set. Vertices at distance 2 from a vertex in the independent set may also be removed if they have no white neighbors, since they no longer influence the remainder of the game. All removed vertices have weight zero. Removed vertices are considered not existing, i.e., when discussing the neighbors of a vertex, these neighbors will be white, red, blue, green or black.

The semantics of the colors imply that we can do the following actions:

- **Rule 1**: a red vertex v has no white neighbors, we can color it blue.

- **Rule 2**: if a blue vertex is not connected to a black vertex, we can color it green. This is valid as it can no longer be removed.
- **Rule 3**: if a green vertex v is connected, either directly or through a black vertex, to a white vertex w, we can color w red. This is valid since placing w in X would remove v.
- **Rule 4**: if a black vertex has no white neighbors, we can remove it. This is valid, as the only purpose of a black vertex is to ensure that its neighbors remain connected to its white neighbors.

Case 1: There is a white vertex v with at least three white neighbors Branching on v gives branching vector $(4, 1 - \alpha)$.

Case 2: There is a white vertex v with two white neighbors

Case 2.1: The subgraph induced by white vertices contains a cycle of length r, with $r \geq 5$. We number the vertices v_1 to v_r. We branch on v_1. If we pick v_1 we decrease the measure by $5 - 2 * \gamma$. If we do not pick v_1 we decrease the measure by $1 - \alpha$. We get a branching vector of $(5 - 2 * \gamma, 1 - \alpha)$.

Case 2.2: The subgraph induced by white vertices contains a cycle of length r, with $r == 4$. We number the vertices v_1 to v_r. We branch on v_1. If we pick v_1 we decrease the measure by 4. Note that we can completely remove v_3 since it has no white neighbors. If we do not pick v_1 we decrease the measure by $1 - \alpha$. We get a branching vector of $(4, 1 - \alpha)$.

Case 2.3: The subgraph induced by white vertices contains a path of length r, with $r \geq 4$. We number the vertices v_1 to v_r. We branch on v_3. If we pick v_3 we decrease the measure by 4. Note that we can completely remove v_1 since it has no white neighbors. If we do not pick v_3 we decrease the measure by $1 - \alpha$. We get a branching vector of $(4, 1 - \alpha)$.

Case 2.4: There is a white vertex v with two white neighbors, neither of which has a white neighbor, except possibly one another

Case 2.4.1: One of the neighbors of v has a red neighbor r_1 We consider each case of placing v or one of its neighbors in x:

- We pick v, the measure decreases by $3 + \alpha - \gamma$.
- We pick the neighbor of v adjacent to r_1. The measure decreases by $3 + \alpha$.
- We pick the other neighbor of v, the measure decreases by 3.
- We pick neither v, nor any of its neighbors. v and its neighbors are colored blue. We decrease the measure by $3 - 3 * \beta$.

Case 2.4.2: One of the neighbors of v has a black neighbor b_1 We consider each case of placing v or one of its neighbors in x:

- We pick v, the measure decreases by 3.
- We pick the neighbor of v adjacent to b_1. The measure decreases by $3 + \gamma$.
- We pick the other neighbor of v, the measure decreases by 3.
- We pick neither v, nor any of its neighbors. v and its neighbors are colored blue. We decrease the measure by $3 - 3 * \beta$.

Case 2.4.3: Neither of the white neighbors of v has any neighbor We consider each case of placing v or one of its neighbors in x:

- We pick v, the measure decreases by 3.
- We pick one of the neighbors of v, the measure decreases by 3.
- We pick the other neighbor of v, the measure decreases by 3.
- We pick neither v, nor any of its neighbors. v can be colored blue and its neighbors are colored green. We decrease the measure by $3 - \beta$.

Case 3: There is a white vertex v with one white neighbor w
Case 3.1: v also has a red neighbor r_1
Case 3.1.1: r_1 has another white neighbor x We consider each case of placing v and its neighbor in x:

- We pick v, the measure decreases by $3 + \alpha - \gamma$.
- We pick w, the measure decreases by $2 + \alpha - \gamma$.
- We pick neither v nor w. v and w are colored blue. We decrease the measure by $2 - 2 * \beta$.

Case 3.1.2: r_1 has no white neighbor x We consider each case of placing v and its neighbor in x:

- We pick v, the measure decreases by $2 + \alpha$.
- We pick w, the measure decreases by $2 + \alpha$.
- We pick neither v nor w. v, w and r_1 are colored blue. We decrease the measure by $2 + \alpha - 3 * \beta$.

Case 3.2: v also has a black neighbor b_1, which has another white neighbor x

Case 3.2.1: b_1 has a third white neighbor We branch on v, giving branching vector $(4 - \gamma, 1 - \alpha)$.

Case 3.2.2: b_1 has exactly two white neighbors and x has another white neighbor We consider each of the following cases of placing the white nodes in X:

- We pick x. The measure decreases by 3.
- We pick v, the measure decreases by 3.
- We pick w, but not x, the measure decreases by $3 - \alpha + \gamma$.
- We do not pick any of v, w or x. v and w are colored blue, x is colored red. b_1 can be removed. We decrease the measure by $3 - \alpha - 2 * \beta + \gamma$.

Case 3.2.3: b_1 has exactly two white neighbors and x has no white neighbor We consider each of the following cases of placing v or w in X:

- We pick v. The measure decreases by $3 + \gamma$.
- We pick w, the measure decreases by 2.
- We do not pick any of v or w. v and w are colored blue. We decrease the measure by $2 - 2 * \beta$.

We consider each case of placing v and its neighbor in x:

- We pick v. The measure decreases by 3.
- We pick w, the measure decreases by 2.
- We pick neither v nor w. v and w are colored blue. We decrease the measure by $2 - 2 * \beta$.

Case 3.3: v also has a black neighbor b_1, which has a red neighbor We consider each case of placing v and its neighbor in x:

- We pick v. The measure decreases by $2 + \alpha$.
- We pick w, the measure decreases by $2 + \gamma$.
- We pick neither v nor w. v and w are colored blue. We decrease the measure by $2 - 2 * \beta + \gamma$.

Case 3.4: v also has a black neighbor b_1, which has a blue neighbor We consider each case of placing v and w in x:

- We pick v. The blue node can be removed The measure decreases by $2 + \beta + 2 * \gamma$.
- We pick w. We can remove b_1 as it is no longer connected to a white node. The measure decreases by $2 + \gamma$.
- We pick neither v nor w. v and w are colored blue. b_1 can be removed. We decrease the measure by $2 - 2 * \beta + \gamma$.

Case 4: There is a white vertex v without white neighbors

Case 4.1: v has a black neighbor b_1, which has a white neighbor w

Case 4.1.1: There are three white vertices v_1, v_2 and v_3, v_1 and v_2 are connected by a black node and v_2 and v_3 are connected by a black node. Note that the black node connecting v_1 and v_2 could be the same as the one connecting v_2 and v_3. We branch on v_2. This gives branching vector $(3 - \gamma, 1 - \beta)$.

Case 4.1.2: There are two white vertices v_1 and v_2, connected by a black node. We branch on v_1. This gives branching vector $(2 + \gamma, 1 - \beta + \gamma)$.

Case 4.2: v has a red neighbor r_1, which has a white neighbor w We branch on v. This gives branching vector $(2 + \alpha, 1)$.

Case 4.3: v has a red neighbor r_1, which has no white neighbors We branch on v. This gives branching vector $(1 + \alpha, 1 + \alpha - \beta)$.

Case 4.4: v has two black neighbors b_1 and b_2, which are connected to two blue nodes, b_3 and b_4, respectively We branch on v. This gives branching vector $(1 + 2 * \beta + 2 * \gamma, 1 + 2 * \gamma)$.

Case 4.5: v has a single black neighbor b_1, which has a blue neighbor b_2, which has a black neighbor b_3, which has a white neighbor w We branch on v. This gives branching vector $(1 + \beta + \gamma, 1 + \gamma)$.

Case 4.6: v has a black neighbor b_1, which has a blue neighbor b_2 We branch on v. This gives branching vector $(1 + \beta + \gamma, 1 + \beta + \gamma)$.

If no case applies, then there are no white vertices left. This means we have found a set of vertices that represents a position that can be reached in a game of GKM_2 on graph G. Choosing $\alpha = 0.5001$, $\beta = 0$ and $\gamma = 0.5001$ gives the claimed upper bound of 1.5875^n solutions. The tight branching vectors are $(3, 3, 3, 3 - \beta)$ from Case 2.4.3, $(3 + \alpha - \gamma, 3 + \alpha - \gamma, 2 - \alpha + \beta)$ from case 3.1.1.1, and $(3, 3, 3 - \alpha + \gamma, 3 - \alpha - 2 * \beta + \gamma)$ from Case 3.2.2. From the above, it follows that there are $O(1.5875^n)$ nontrivial reachable positions.

As the value 1.5875 is obtained by rounding, and there are at most $n + m$ trivial positions, the result follows. $\square$

5 Polynomial time algorithms for special classes of graphs

Section 3 it is shown that Kayles with Marking is PSPACE-complete. It is also shown that Kayles with Removal is PSPACE-complete for $d = 2$. This means that it is unlikely that there is a polynomial time algorithm for solving those problems on arbitrary graphs. In this section it is shown that there exist polynomial time algorithms for certain classes of graphs.

5.1 Interval Graphs

In [5], Bodlaender and Kratsch have shown that regular Kayles can be solved in $O(n^3)$ time on interval graphs. Using Theorem 2 we have that if a game of Kayles with Marking with distance d is played on a graph G, and the d-th powergraph of G is an interval graph, then the game of Kayles with Marking can be solved in $O(n^3)$ time. With the following theorem we have that a game of Kayles with Marking with distance d played on an interval graph G can be solved in $O(n^3)$ time.

Theorem 9. *The power graph of an interval graph is still an interval graph.*

Proof. Take an interval graph G. The set S is a set of intervals which represent graph G. Construct the set of intervals S' by taking, for each interval v in S, an interval v', which has the same left side as v, but whose right side is equal to the maximum of the left sides of all intervals at distance at most d in G. Let G' be the interval graph corresponding to S'. Because the intervals were extended in S', all nodes that were at distance d or less from each other in S, are neighbors in G'.

Suppose there exist a node v and w at distance greater than d in G, and v is to the left of w in S. Now suppose that these nodes are neighbors in G'. Let l be the left bound of w in S' and let r be the right bound of v in S'. Since v and w are connected, it follows that $r \leq l$. There exists a node u for which the left side in S' is equal to r, which is at distance at most d from v in G. Take a path from v to u in G of distance at most d. Take the intervals representing these nodes in S. l must be contained in one of these intervals and therefore a neighbor of w. Since the interval containing l is at distance less than d from v, the distance from v to w is at most d. This is in contradiction with the assumption that v and w are at a distance greater than d in G. It follows that there exist no two nodes at distance greater than d in G, which are neighbors in G'.

This means that the powergraph of an interval graph is itself powergraph. □

5.2 Kayles with Marking on Cocomparability graphs

In [5], Bodlaender and Kratsch have also shown that regular Kayles can be solved in $O(n^3)$ time on cocomparability graphs. In fact the interval graphs from the last section are a special class of cocomparability graphs. In this section the following theorem is proven:

Theorem 10. *The powergraph of a cocomparability graph is also a cocomparability graph.*

Proof. From this theorem it follows that a game of Kayles with Marking with distance d played on a cocomparability graph G can be solved in $O(n^3)$ time. This works similar as in the previous section. The proof of the theorem follows:

Suppose that the k-th powergraph of cocomparability graph G is itself not a cocomparability graph. Then there must be some triple i, j, k, $i < j < k$, with and edge between v_i and v_k, but no edge between v_i and v_j and no edge between v_j and v_k. If the edge between v_i and v_k was already there in the original graph, then this is in contradiction with the original graph being a cocomparability graph. If the edge was added when taking the powergraph of G, then the distance between v_i and v_k must be at most k. Consider a path of length at most k from v_i to v_k in G. Since $i < j < k$, this path must pass node v_i at some point. There must be some consecutive nodes v_s and v_t on the path such that $s < j < t$. Since there is an edge between v_s and v_t, there must also be an edge from one of these nodes to v_j. The distance from v_i to v_s is at most $k - 1$ and the distance from v_k to v_t is also at most $k - 1$. Therefore in the powergraph, there must either be an edge from v_i to v_j or an edge from v_k to v_j. This is in contradiction with the assumption that the powergraph of a cocomparability graph is not a cocomparability graph. □

5.3 Kayles with Removal on Cocomparability graphs

In this section the following theorem is proven:

Theorem 11. *Kayles with removal can be solved in $O(n^3)$ time on cocomparability graphs.*

Proof. This is shown by using Theorem 1. Since Kayles with Marking can be solved in $O(n^3)$ time on cocomparability graphs, if a game of Kayles with Removal on a cocomparability graph is equivalent to a game of Kayles with Marking, the above theorem follows. To prove this, it can be shown that removing a vertex v and $N_d(v)$ does not increase the distance between two vertices from smaller than or equal to d to larger than d. This can be shown as follows:

In a game of GKR on a Cocomparability graph, if a player makes a move on a node v_i, in the resulting graph, all nodes with higher cocomparability order, that are at distance more than d from that node, will fall in one section of the graph All nodes with lower cocomparability order, that are at distance more than d from that node, will fall in another section. There will be no edges between the sections. This can be proven as follows.

Suppose there is an edge from a node with a lower cocomparability order to a node with a higher cocomparability than node v_i. At least one of these nodes must have an edge to v_i. Removing v_i must have removed that node and therefore the edge. □

6 Conclusion

In this section an overview is given of the results from this paper. It also gives an overview of the problems that remain open.

This paper shows that GKM_d is PSPACE-complete. It has also shown that GKR_2 is PSPACE-complete. Whether GKR_d is PSPACE-complete for values of d greater than two remains an open problem.

This paper gives an $O^*(1.5875^n)$ algorithm for finding a winning strategy in GKM_2. We suspect that GKM_d for larger values of d will result in even better upper bounds, but this remains an open question, as only an upper bound of $O^*(1.6031^n)$ has been proven. For GKR_d on graph G, the best known algorithm is to compute a nimber for every subgraph of G. This gives an upper bound of $O*(2^n)$. We strongly suspect that a better upper bound for an algorithm is possible, however we were unable to prove this.

This paper also shows that it is possible to determine if a winning strategy exists on a cocomparability graph in $O(n^3)$ time for both GKR_d and GKM_d.

References

[1] Thomas J. Schaefer, *On the Complexity of Some Two-Person Perfect-Information Games*, Journal of Computer and System Sciences, Volume

16 (1978): 185-225.

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

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

[4] Hans L. Bodlaender, Dieter Kratsch, Sjoerd T. Timmer, *Exact Algorithms for Kayles*, Theoretical Computer Science, Volume 562 (2015): 165-176.

[5] Hans L. Bodlaender, Dieter Kratsch, *Kayles and nimbers*, Journal of Algorithms, Volume 43, Number 1 (2002): 106-119.

[6] F.V. Fomin and D. Kratsch, *Exact Exponential Algorithms*, Springer, 2010.

[7] Rudolf Fleischer, Gerhard Trippen, *Kayles on the Way to the Stars*, Computers and Games (2004): 232-245.

[8] Adrien Guignard, Eric Sopena, *Compound Node-Kayles on Paths*, Theoretical Computer Science, Volume 410, Numbers 21-23 (2009): 2033-2044.