ENGLISH SUMMARY OF THE THESIS OF
Antoine Gianni Philippe DAILLY

Criticality, identification and vertex deletion games on graphs

Content

This document is an extended summary of Antoine Dailly's thesis (written in French). It contains most of the literature review and main results, as well as all the proofs that are not in submitted and published papers [12, 21, 32, 33, 34, 35]. The reader is invited to refer to the full thesis and the papers for more details on the covered topics.

Abstract

In this thesis, we study both graphs and combinatorial games. There are several links between those two domains: games are useful for modeling an opponent in optimization problems on graphs, and in the other direction several classical games are played on graphs. We will study two graph problems and adapt some classical combinatorial games to be played on graphs.

In a first chapter, we study a criticality problem. A graph that verifies some property, and such that any modification (vertex or edge addition or deletion) breaks the property is called *critical* for this property. We focus on the critical graphs for the property "having diameter 2", called D2C graphs. The Murty-Simon conjecture gives an upper bound on the number of edges in a D2C graph with a given number of vertices. However, recent research suggests that this bound can be improved for non-bipartite D2C graphs. We show the validity of this approach by proving a smaller upper bound for a subfamily of non-bipartite D2C graphs.

In a second chapter, we consider an *identification* problem. Identification consists in assigning some data to every edge or vertex of a graph, such that this assignment induces a label to every vertex with the added condition that two distinct vertices must have a different label. We define an edge-coloring using sets of integers inducing an identification of the vertices, and prove that this coloring requires at most a logarithmic number of integers (with respect to the order of the graph) in order to successfully identify the vertices. This result is compared with other identifying colorings, for which the number of colors required to successfully identify the vertices can be linear with respect to the order of the graph.

In order to show the link between graphs and games, we adapt a well-known family of games to be played on graphs. We propose a general framework for the study of many *vertex deletion games* (which are games in which the players delete vertices from a graph under predefined rules) such as Arc-Kayles. This framework is a generalization of subtraction and octal games on graphs. In their classical definition, those games exhibit a high regularity: all finite subtraction games are ultimately periodic, and Guy conjectured that this is also true for all finite octal games.

We specifically study the connected subtraction games CSG(S) (with S being a finite set). In those games, the players can remove k vertices from a graph if and only if they induce a connected subgraph, the graph remains connected after their deletion, and $k \in S$. We prove that those games are all ultimately periodic, in the sense that for a given graph and vertex, a path attached to this vertex can be reduced (after a certain preperiod) without changing the Grundy value of the graph for the game. We also prove pure periodicity results, mostly on subdivided stars: for some sets S, the paths of a subdivided star can be reduced to their length modulo a certain period without changing the outcome of the game.

Finally, we define a weighted version of Arc-Kayles, called Weighted Arc-Kayles (WAK for short). In this game, the players select an edge and reduce the weight of its endpoints. Vertices with weight 0 are removed from the graph. We show a reduction between WAK and Arc-Kayles, then we prove that the Grundy values of WAK are unbounded, which answers an open question on Arc-Kayles. We also prove that the Grundy values of WAK are ultimately periodic if we fix all but one of the weights in the graph.

Keywords : Criticality, identification, coloring, graph theory, vertex deletion games, subtraction games, combinatorial game theory, combinatorics.

Contents

Introduction

Introduction

I started studying graphs and games during my research internship in 2015, withing the GAG[1] ANR project, under the supervision of Éric Duchêne and Aline Parreau. I immediately knew that this would stir my interest: two domains that were initially very theoretical before becoming necessary to many practical applications. Graphs are a very powerful mathematical model that are used everywhere in computer science and beyond (mathematics, industrial sciences...). Combinatorial games are two-player games with perfect information and no chance in which the winner is entirely decided by the last move. Combinatorial game theory allows the study of well-known games such as Go, Chess, Hex, and is invaluable in our modern understanding of number theory and optimization paremeters.

In the vast field theoretical computer science, graph theory and combinatorial games theory are close neighbours, but have few interactions. Graphs are a useful model for games, and several games are defined on graphs, but it is only recently that new links between the two domains have grown in importance. The GAG ANR project is focused on bringing closer together the fields of graphs and combinatorial games, and this thesis is in the same spirit. In this manuscript, we will study graph problems and games on graphs.

In Part I, we will study two different graph problems. First, in Chapter 1, we will work on critical graph theory, which is the study of the graphs that have a certain property and such that any small modification creates a graph without this property. The graphs that we study are called *diameter 2 critical* (or D2C): they have diameter 2, but removing any edge increases the diameter. In the 1960s, several authors have studied this family and conjectured an upper bound on the number of edges in a D2C graphs of a given order. However, recent studies indicate that this upper bound is not tight for most of those graphs, which is why we propose a strengthening of this conjecture. We then prove a stronger bound for a subfamily of D2C graphs.

In Chapter 2, we study a variant on the *vertex-distinguishing* problem, which consists in assigning a unique label to every vertex of a graph. This variant defines a parameter, which is the smallest quantity of information necessary to distinguish every vertex. We show that this quantity can only take three values, which is very different from most of the other known ways to distinguish vertices.

In Chapter 3, we present several links between games and graphs. In particular, we show how to change graph properties and parameters into games and the new problems that arise with those definitions. Indeed, game problems are generally way more computationnally difficult than the graph problems they are derived from.

Part II is focused on combinatorial games on graphs. First, in Chapter 4, we give a summary of the theory of *impartial* games (which are combinatorial games in which the only difference between the two players is who begins). In the 1930s, it was proved that all impartial games are actually equivalent to Nim, the first combinatorial game in history (that was studied in 1901), which allows a better study of this family. We focus in particular on taking-breaking games, a subfamily of impartial games where players remove counters from heaps, and may break the heaps. We extend the definition of these games to play them on graphs, which includes several well-known games from the literature such as Arc-Kayles (where the players remove two adjacent vertices until the graph has no edge left).

In Chapter 4, we study the family of connected subtraction games, which are the extension on graphs of the subtraction games. Those games played on heaps of counters have a regular behaviour. We prove that their extension on graphs also have a regular behaviour, then we study several subfamilies on some graph classes. We prove pure periodicity results: in some families (such as subdivided stars), the paths can be reduced without changing the game. We also study a variant of Arc-Kayles where the players are not allowed to disconnect the graph.

Finally, in Chapter 6, we study a variant of Arc-Kayles played on weighted graphs: the players select an edge and reduce the weights of its endpoints. The vertices with weight 0 are then removed from the graph. We prove several results on this game, such as a periodicity theorem when all weights but one are fixed, and use one of our results to answer an open question on Arc-Kayles, which is a very difficult game.

[1] Graphs and Games : `https://projet.liris.cnrs.fr/gag/`

Preliminaries

We give in this section basic definitions of graphs and complexity that we will use in this manuscript. More specific definitions will be given in the chapters where they are used.

Graphs

A graph $G(V,E)$ is a vertex set V and an edge set $E \subseteq V^2$, an edge linking two vertices together. If the elements of E are ordered, then G is *directed*; otherwise it is *undirected.* An edge between two vertices u and v will be denoted indefferently by uv or vu if G is undirected; if G is directed it will be denoted by $\overrightarrow{uv}$ if the ordered pair (u,v) is in E. Most of the graphs that we will study will be undirected. The only directed graphs that we will consider will be identified as such.

In the whole manuscript except Chapter 6, we will consider *simple* graphs: an edge will always link two distinct vertices, and there can only be at most one edge between two vertices. Except in Chapters 2 and 6, graphs will be *unlabeled*: neither vertices nor edges will have a label.

The *order* of a graph G is the size of its vertex set, and will be indifferently denoted by $|V(G)|$ or n if no confusion is possible. The *size* of a graph G is the size of its edge set, and will be indifferently denoted by $|E(G)|$ or m if no confusion is possible. if $n=0$ we way that G is *empty*, if $n=1$ we say that G is *trivial.*

Two vertices u and v are *adjacents*, or *neighbours*, if uv is an edge. Otherwise we denote by $\overline{uv}$ the non-edge between u and v. Let $e=uv$ be an edge, u and v are called the *endpoints* of e and we say that they are *incident* with e. Given a vertex u, we define its *neighbourhood*, denoted by $N(u)$, as the set of its neighbours: $N(u)=\{v\in V | uv\in E\}$. The *closed neighbourhood* of u, denoted by $N[u]$, is the neighbourhood plus u itself: $N[u]=N(u)\cup\{u\}$. The *degree* of a vertex u in the graph G, denoted by $d_G(u)$ or $d(u)$ when the context is clear, is the size of its neighbourhood: $d(u)=|N(u)|$. The *maximal degree* of G, denoted by $\Delta(G)$, is the highest degree of a vertex in G: $\Delta(G)=\max_{u\in V}(d(u))$. The *minimal degree* of G, denoted by $\delta(G)$, is the smallest degree of a vertex in G: $\delta(G)=\min_{u\in V}(d(u))$.

A *path* between two vertices u and v is a sequence of vertices $u=u_0,u_1,\ldots,u_k=v$ such that u_iu_{i+1} is an edge for all $i\in\{0,\ldots,k-1\}$. A graph is *connected* if there is a path between every pair of vertices. Given a graph $G(V,E)$, we can partition V into one or several connected subsets of vertices $V_1,\ldots,V_k$ such that if there is an edge uv with $u\in V_i$ and $v\in V_j$ then $i=j$. The graphs $G_1(V_1,E_1),\ldots,G_k(V_k,E_k)$ (where E_i is the subset of the edges of E incident with a vertex of V_i) are called the *connected components* of G.

Given a simple undirected graph $G(V,E)$, its *complement* is the graph $\overline{G}(V,\overline{E})$ with the same vertices as G, and which edges are exactly the non-edges of G: $\overline{E}=\{uv\in V^2 | u\neq v \text{ and } uv\notin E\}$. Note that $\overline{G}$ has $\binom{n}{2}-m$ edges, and we have $\Delta(\overline{G})=n-1-\delta(G)$ and $\delta(\overline{G})=n-1-\Delta(G)$.

Let $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$ be two simple undirected graphs. A function $\Phi: V_1\to V_2$ is a *morphism* between G_1 and G_2 if for any edge $uv\in E_1$, $\Phi(u)\Phi(v)\in E_2$. The graphs G_1 and G_2 are *isomorphic* if there is a bijective morphism between G_1 and G_2 and if the inverse function is also a morphism. We will consider that isomorphic graphs are equal, and we will write $G_1=G_2$.

If $V_1\subseteq V_2$ and $E_1\subseteq E_2$, we call G_1 a *subgraph* of G_2. Furthermore, if there is no edge from $E_2\setminus E_1$ having its two endpoints in V_1, then G_1 is a *induced subgraph* of G_2. We say that G_2 is G_1-*free* if no induced subgraph of G_2 is isomorphic to G_1.

If $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$ are two graphs on disjoint vertex sets (such that $V_1\cap V_2=\emptyset$ and $E_1\cap E_2=\emptyset$), we define their *disjoint union*, cenoted by $G_1\cup G_2$, as the graph with vertex set $V_1\cup V_2$ and edge set $E_1\cup E_2$.

Some graph classes will be met several times in this manuscript, so we define them here.

Given an integer n, the *path* P_n is the graph with vertex set $\{u_1,\ldots,u_n\}$ and with edges u_iu_{i+1} for $i\in\{1,\ldots,n-1\}$. The *cycle* C_n is the graph with vertex set $\{u_1,\ldots,u_n\}$ and with edges u_iu_{i+1} for $i\in\{1,\ldots,n-1\}$ as well as the edge u_nu_1.

A graph with no cycle is called a *forest.* If a forest has only one connected component, it is a *tree.* The vertices of a forest with degree 1 are called *leaves.* Sometimes, a specific vertex is called the *root*, which helps us define the *depth* of a vertex in a tree as the number of edges in the path from the vertex to the root. The depth of a tree is the highest depth of all its vertices.

Given an integer n, the *complete graph* K_n is the graph with n vertices and all possible edges. A complete graph is sometimes called a *clique*, especially when it is induced by a subset of vertices in a bigger graph. The most well-known complete graph is the *triangle* K_3. The number of vertices in the biggest clique in a graph

G is denoted by $\omega(G)$. The complement of a complete graph is called an *independent set*, or *stable set*: a vertex is independent in a part of a graph if it has no neighbour in this part. A *split graph* is a graph in which vertices can be partition into one clique and one independent set.

Given two integers m and n, the *complete bipartite graph* $K_{m,n}$ is the graph comprised of two independent sets of respectively m and n vertices such that every vertex in one part is adjacent to every vertex in the other part. When $m = 1$, the graph is called a *star*. Note that a star is a tree.

Given an edge $e = uv$ of a graph $G(V, E)$, we can *subdivide* e by adding to V a vertex w, removing from E the edge uv and adding to E the edges uw and wv. A graph obtained by subdividing edges from G is called a *subdivision* of G. For example, we can see that every nontrivial path is a recursive subdivision of P_2. We say that such paths are *subdivided* P_2.

Complexity

Complexity is one of the basis of computer science. Indeed, most of the problems in computer science are to find computationally efficient solutions to several issues. The complexity theory aims to classify those problems into categories that represent their inherent difficulty. We will quickly present the basis of complexity that will be used in this manuscript. Most of the definitions are based on classical books [6, 49, 97].

A problem is a *decision problem* if it asks a question on its input, to which the answer is either Yes or No. The classical problem of 3-Coloring, presented below, is such a problem. A first division of decision problems is between decidable problems, for which there is an algorithm answering Yes or No in a finite number of steps for any input, and undecidable problems, for which such an algorithm cannot exist. For example, 3-Coloring is decidable, since the enumeration of all possible coloring functions c and test them allows us to answer in finite time (since there are $3^{|V|}$ different functions, which is a lot but still a finite number).

Decision Problem.

3-Coloring

INPUT: A graph $G(V, E)$.

QUESTION: Is there a function $c : V \to \{1, 2, 3\}$ such that $uv \in E \Rightarrow c(u) \neq c(v)$?

In this manuscript, we will only work on decidable problems. From this, a new division is created by the number of steps that are necessary to answer the question. The problems are classified by using this number of steps, which is expressed in function of the *size* of the input, which is the memory space necessary to encode the input. For example, for 3-Coloring, the input is a graph $G(V, E)$. So we need to encode the number of vertices, which requires $\log_2(|V|)$ cases (by writing in binary), and the edges, which requires $|E| \log_2(|V|)$ (by writing the two endpoints in binary). This induces the definition of several classes: P (which is the family of problems such that an algorithm answers in a number of steps that is polynomial in function of the input size) or PSPACE (which is the family of problems such that an algorithm answers by using a memory space that is polynomial in function of the input size) for example.

However, this classification is not necessarily precise enough: for several decision problems, deciding whether the answer is Yes or No is difficult, but verifying the validity of an answer can be done very efficiently. For example, for 3-Coloring, it is necessary to verify all possible functions to answer No (as we told above, an exponential number). However, if the answer if Yes, then it is enough to exhibit a coloring function. More generally, given an input of a decision problem, a *certificate* is an information that allows the efficient (in a number of steps polynomial in the size of the input) verification that the answer to the problem with this input is either Yes or No. Thus, a coloring function is a certificate for 3-Coloring. This creates new complexity classes, such as NP (which is the family of problems such that there is an algorithm such that the answer is Yes (resp. No) on an input if and only if there is a certificate verifying if the answer is Yes (or No) that the algorithm can verify in a number of steps polynomial in the size of the input).

Those complexity classes are not necessarily disjoint. While it is easy to see that P $\subseteq$ NP, the question of the existence of problems that are in NP but not in P is one of the main problems in mathematics and computer science. Similarly, we know that NP $\subseteq$ PSPACE but we do not know whether this inclusion is strict or not. One of the ways we use to characterize a complexity class is to identifiy its more difficult problems.

Given two problems P and Q, a *polynomial reduction* from P to Q is a function f that transforms any input E of P into an input $f(E)$ of Q, such that the size of $f(E)$ is polynomial in the size of E, and we answer

Yes to P on E if and only if we answer Yes to Q on $f(E)$. Given a complexity class $\mathcal{C}$, a decision problem is $\mathcal{C}$-*complete* if it is in $\mathcal{C}$ and there exists a polynomial reduction from any problem in $\mathcal{C}$ to this problem. In other words, these are the hardest problems in $\mathcal{C}$.

The most classical decision problems are the boolean satisfiability problems, called SAT.

Decision Problem.

SAT

INPUT: A boolean formula F on variables x_i.

QUESTION: Is there an assignment of the x_i to `True` or `False` such that F is `True`?

According to the kind of boolean formula, the problem can belong to different complexity classes. For example, if the formula is in conjunctive normal form with two variables in each clause, then the problem (called 2-SAT) is in P. However, with the same setting but three variables in each clause, the problem (called 3-SAT) is NP-complete. Worse, QBF-SAT, in which the formula is quantified, is PSPACE-complete. This is due to the fact that the certificate for a boolean formula written in conjunctive normal form is a simple assignment of the variables, while a certiciate for a quantified boolean formula is an assignment strategy.

Decision Problem.

QBF-SAT

INPUT: A quantified boolean formula[2] F on variables x_i.

QUESTION: Is F `True`?

Generally, when we study a decision problem, we want to prove that it is in P. When it is not the case, there are two possibilities: finding the best complexity class, or restraining the input size and finding a variant that is in P. This is what is done with SAT: the general problem is very difficult (the variant QBF-SAT, which is already a restriction, is PSPACE-complete), but the variant 2-SAT is in P.

[2] A boolean formula is quantified if it is written $Q_1x_1Q_2x_2\dots Q_nx_n \mid F(x_1,\dots,x_n)$ with $Q_i \in \{\forall,\exists\}$. For example, $\forall x_1\exists x_2\exists x_3 \mid (x_1 \vee \neg x_2) \wedge x_3$ is a quantified boolean formula.

Part I

Graphs

Chapter 1

Diameter-2-critical graphs

The work in this chapter was mostly realized during a research stay at the UNAM Juriquilla, in Mexico, under the supervision of **Adriana Hansberg** and with **Florent Foucaud**. A paper is currently being written [33], and the work was presented during the International Colloquium on Graph Theory and Combinatorics 2018.

Chapter abstract

In this chapter, we study the diameter-2-critical graphs (also called D2C graphs). Those are graphs with diameter 2 and such that the deletion of any edge increases the diameter. Such graphs have been extensively studied since Murty and Simon independently conjectured that a D2C graph of order n has at most $\lfloor \frac{n^2}{4} \rfloor$ edges, and that the extremal graph is the balanced complete bipartite graph. In the Section 1.1, we will present the background on the Murty-Simon Conjecture. In Section 1.2, we will talk about how this conjecture can be strengthened. Finally, in Section 1.3, we will prove a stronger bound on the class of D2C graphs with a dominating edge.

1.1 Introduction

1.1.1 Definitions

In this chapter, we will consider that, given a graph $G(V, E)$, the *edge deletion* is a graph operation consisting in removing an edge from the edge-set E. We will note the graph resulting from the deletion of the edge e by $G - e$. The *distance* between two vertices u and v, noted $\mathrm{dist}(u, v)$, is the number of edges in the shortest path from u to v. The *diameter* of a graph G, noted $\mathrm{diam}(G)$, is the biggest distance between two vertices of G. We now define diameter-critical graphs:

Definition 1.1. *Let d be a positive integer. A graph G is* diameter-d-critical*, noted DdC, if and only if:*

1. $\mathrm{diam}(G) = d$;
2. *For every edge e in G,* $\mathrm{diam}(G - e) > d$.

For example, a complete bipartite graph (with two nontrivial parts) is D2C: the diameter is 2 since two vertices in the same part are at distance 2 and two vertices in different parts are at distance 1, and by deleting an edge uv the vertices u and v are now at distance 3 from each other, so the diameter is increased by the deletion of any edge.

While DdC graphs have been extensively studied in the literature, the case of D2C graphs has attracted even more attention. Indeed, many well-known graphs are D2C: complete bipartite graphs, the Petersen Graph, the Grötzch Graph or Moore graphs for instance. Several well-known D2C graphs are depicted on Figure 1.1.

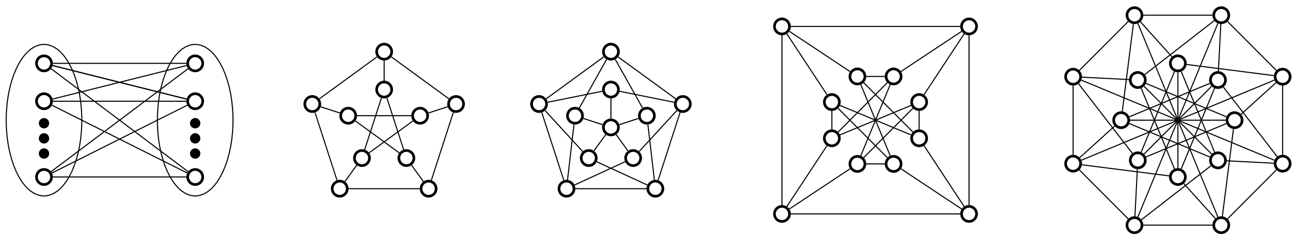


Figure 1.1: Five examples of D2C graphs: a complete bipartite graph, the Petersen Graph, the Grötzsch Graph, the Chvátal Graph and the Clebsch Graph.

1.1.2 The Murty-Simon Conjecture

Plesník [87] noticed that all the known D2C graphs had at most as many edges as a complete bipartite graph of the same order. Murty [20] and Simon independently formulated the following conjecture (which, according to Erdös, was also stated by Ore in the 1960s):

Conjecture 1.2 (Murty-Simon Conjecture, *cf.* [25])**.** *Let G be a D2C graph of order n and size m. Then, $m \leq \lfloor \frac{n^2}{4} \rfloor$ and we have equality if and only if $G = K_{\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil}$.*

The Mantel Theorem [81] states that a triangle-free graph has at most $\lfloor \frac{n^2}{4} \rfloor$ edges, and that the only graph reaching this bound is the balanced complete bipartite graph. As a consequence, the Murty-Simon Conjecture is valid for triangle-free D2C graphs[1].

The Murty-Simon Conjecture has been extensively studied, and several partial results have been found. Plesník [87] proved that $m < \frac{3n(n-1)}{8} = 0.375(n^2 - n)$. Cacceta and Häggkvist [25] proved that $m < (\frac{1+\sqrt{5}}{12})n^2 < 0.27n^2$. Fan [40] proved the validity of the Murty-Simon Conjecture for D2C graph of small order ($n \leq 24$ and $n = 26$), and found that $m < 0.2532n^2$ for other D2C graphs.

The most important result in the study of the Murty-Simon Conjecture is due to Füredi [47], who proved the validity of the conjecture for graphs of very large order: $n \geq n_0$ where n_0 is a tower of powers of 2 of size 10^{14}. Note that, according to him, his result is the first to use Szemerédi's Regularity Lemma to yield an exact bound.

1.1.3 Total domination and the Murty-Simon Conjecture

In 2003, Hanson and Wang [57] found a link between the Murty-Simon Conjecture and total domination. Tis gave way to new research on Conjecture 1.2.

In a graph $G(V, E)$, a *dominating set* is a set $S \subseteq V$ such that every vertex in V is either in S or a neighbour of a vertex in S. The smallest size of a dominating set in a graph G is called its *domination number* and is noted $\gamma(G)$. If two adjacent vertices form a dominating set, then they are called a *dominating edge*.

In 1980, Cockayne *et al.* introduced the concept of *total domination*. In a graph $G(V, E)$, a *total dominating set* is a set $S \subseteq V$ such that every vertex in V has a neighbour in S. The smallest size of a total dominating set in a graph G is called its *total domination number* and is noted $\gamma_t(G)$. Note that a dominating edge is a total dominating set of size 2.

The difference between a dominating set and a total dominating set is that elements in the total dominating set have to be dominated too. Since a total dominating set is also a dominating set, we have $\gamma(G) \leq \gamma_t(G)$. Total domination was initially introduced to modelize the queens' problem or to find a way to form a committee from a group (each member has to know another member, and each non-member has to know a member), and since then has been extensively studied. However, we are only going to show how it is related to the Murty-Simon Conjecture.

In [103] was introduced the concept of k-total dominated edge-critical graphs:

[1] Note that triangle-free graphs of diameter 2 are D2C.

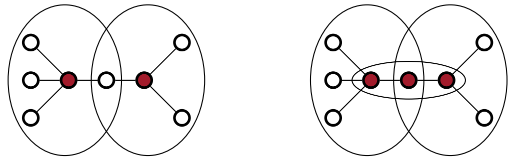


Figure 1.2: The difference between domination and total domination: on the left, the red vertices are a dominating set, but not a total dominating set since neither of them is dominated. On the right, however, the three red vertices are a total dominating set.

Definition 1.3. *Let k be a positive integer. A graph G is* k-total dominated edge-critical*, noted kTC, if and only if:*

1. $\gamma_t(G) = k$;
2. *For every non-edge $\overline{e}$, $\gamma_t(G + \overline{e}) < k$.*

For example, the union of two disjoint complete graphs (each of them being of order at least 2) is 4TC: a total dominating set is necessarily formed by taking two vertices from each clique (thus we have a total dominating set of size 4), and if two vertices u and v in each clique are linked by an edge, then uv is a dominating edge, and thus we have a total dominating set of size 2.

The authors of [103] proved that the total dominating number can be reduced by at most 2 when adding an edge. The graphs for which $\gamma_t(G) = k$ and $\gamma_t(G + \overline{e}) = k - 2$ for every non-edge $\overline{e}$ are called *k-supercritical*, and are exactly the disjoint unions of at least two complete bipartite graphs of order at least 2. Note that the union of two cliques is the complement of a complete bipartite graph.

Hanson and Wang [57] found the following link between total domination and the Murty-Simon Conjecture:

Theorem 1.4 (Hanson and Wang [57])**.** *A graph is D2C if and only if its complement is either 3TC or 4-supercritical.*

Since, by our previous remark, the only 4-supercritical graphs are the complement of the complete bipartite graphs, the Murty-Simon Conjecture can be stated in the complement:

Conjecture 1.5. *Let G be a 3TC graph of order n and size m. Then, $m > \lceil \frac{n(n-2)}{4} \rceil$.*

This reformulation has led to several studies of the Murty-Simon Conjecture, focusing on the total domination aspect in the complement. A series of paper [57, 64, 104] proved that Conjecture 1.5 is valid for 3TC graphs of diameter 3[2]. This proves the validity of the Murty-Simon Conjecture for D2C graphs with a dominating edge.

Other works focusing on this reformulated conjecture include work on small minimum degree 3TC graphs [61, 73]. Haynes, Henning and several co-authors then studied several classes of 3TC graphs, proving Conjecture 1.5 for claw-free, C_4-free, house-free and diamond-free 3TC graphs [60, 62, 63, 65].

As we have seen in this section, while the Murty-Simon Conjecture is still open, there have been many advances and it is generally agreed that the conjecture is valid. Furthermore, recent research seems to suggest that the bound is not tight and can be strengthened, which is what we will present in the next section.

1.2 Strengthening the Murty-Simon Conjecture

1.2.1 From Füredi's claim to triangle-free D2C graphs

In his paper [47], Füredi claims that the bound of the Murty-Simon Conjecture can be strengthened: by excluding the complete bipartite graphs and the graph formed by subdividing an edge from a balanced complete

[2]The 3TC graphs have diameter either 2 or 3, as was shown in [103].

bipartite graph, he claims that the size of a D2C graph of order n is less than $\lfloor\frac{(n-1)^2}{4}\rfloor+1$. This bound is better than the Murty-Simon bound by a linear factor.

However, in [8], Balbuena *et al.* proved that Füredi's claim was inexact. Indeed, they proved that a non-bipartite triangle-free D2C graph of order n has at most $\lfloor\frac{(n-1)^2}{4}\rfloor+1$ edges, with equality if and only if the graph is a specific inflation of C_5. An *inflation of* C_5, noted $C_5(n_0,\dots,n_4)$, is a graph formed by replacing each vertex x_i from C_5 by an independent set X_i of order n_i. In particular, we define the following family of D2C graph of order n:

$$\mathscr{F}_n=\left\{C_5(n_0,n_1,n_2,n_3,n_4)|n_0=n_1=1,n_3=\left\lfloor\frac{n-3}{2}\right\rfloor \text{ ou } n_3=\left\lceil\frac{n-3}{2}\right\rceil,n=2+n_2+n_3+n_4\right\}$$

The authors of [8] then prove the following:

Theorem 1.6 (Balbuena, Hansberg, Haynes, Henning [8])**.** *Let G be a non-bipartite triangle-free D2C graph of order n and size m. Then, $m\le\lfloor\frac{(n-1)^2}{4}\rfloor+1$, with equality if and only if $G\in\mathscr{F}_n$.*

Thus, while Füredi's claim was inexact (the extremal graphs for his strengthened bound are more numerous than what he claimed), the improved bound is true for triangle-free D2C graphs. However, among all the known D2C graphs, only one does not follow the improved bound: the graph called H_5, which is depicted on Figure 1.3. In the next section, we will discuss two possible strengthenings of the Murty-Simon Conjecture.

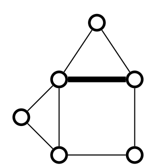


Figure 1.3: The graph H_5, a D2C graph with a dominating edge (in bold), does not follow the improved bound.

1.2.2 The strengthened conjecture

Balbuena *et al.* [8] proposed a strengthening of the Murty-Simon Conjecture. They conjectured that Füredi's claimed strengthened bound was true for D2C graphs without a dominating edge, and that the graphs reaching the bound were, beyond a certain rank, the graphs of $\mathscr{F}_n$.

We build upon this, and propose a linear strengthening of the Murty-Simon Conjecture. A computer search on all D2C graphs of order up to 11 yielded exactly 13 non-bipartite D2C graphs that are not in $\mathscr{F}_n$ and reach the improved bound of $\lfloor(n-1)^2/4\rfloor+1$ edges. Those graphs are depicted on Figure 1.4. Note that they have up to 9 vertices: no D2C graph of order 10 or 11 that are not in $\mathscr{F}_n$ reach this bound. Thus, we propose the following conjecture:

Conjecture 1.7 (Linear strengthening of the Murty-Simon Conjecture)**.** *Let G be a non-bipartite D2C graph of order n and size m. If G is not H_5, then $m\le\lfloor\frac{(n-1)^2}{4}\rfloor+1$, with equality if and only if G belongs to $\mathscr{F}_n$ or is one of the thirteen graphs from Figure 1.4.*

In the next section, we are going to present a first step in the direction of Conjecture 1.7, and show that the bound of the Murty-Simon Conjecture is not tight for the D2C graphs with a dominating edge by proving the following:

Theorem 1.8. *Let G be a non-bipartite D2C graph with a dominating edge of order n and size m. If G is not H_5, then $m\le\lfloor\frac{n^2}{4}\rfloor-2$.*

1.3 Study of D2C graphs with a dominating edge

In this section, we will first describe the structure of the D2C graphs with a dominating edge, before proving Theorem 1.8.

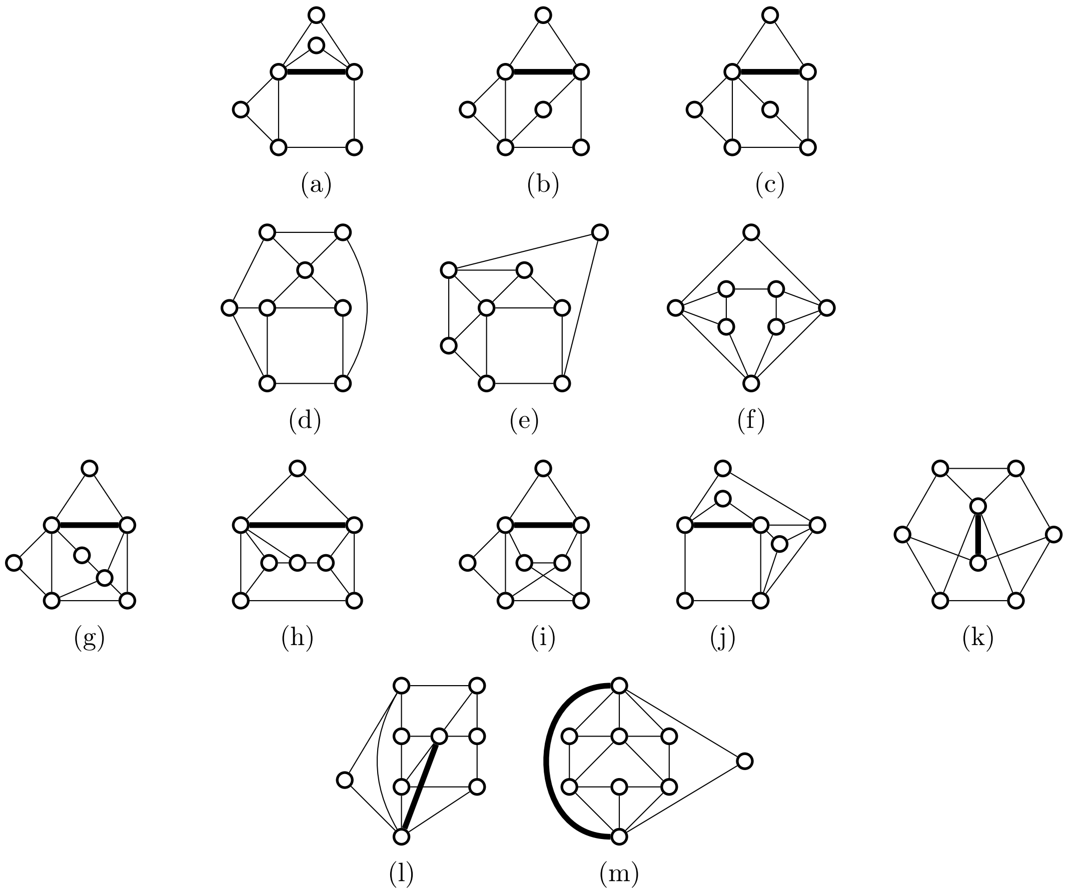


Figure 1.4: The thirteen non-bipartite D2C graphs of order $n \leq 11$ with $\lfloor (n-1)^2/4 \rfloor + 1$ edges that are not in the family $\mathscr{F}_n$. Bold edges are dominating. Only the graphs (d), (e) and (f) have no dominating edge.

1.3.1 Structure of D2C graphs with a dominating edge

All the notations that we introduce in this section will be used in the next section. Recall that a non-edge between x and y is denoted $\overline{xy}$. We start with the following definition, which is fundamental to our study:

Definition 1.9. *Let G be a D2C graph. An edge $uv \in E(G)$ is* critical *for a pair of vertices $\{x, y\}$ if the only path of length 1 or 2 from x to y uses the edge uv.*

Note that, in a D2C graph, an edge xy is critical for either the pair $\{x, y\}$ itself (in which case $N(x) \cap N(y) = \backslash$) or for a pair $\{x, z\}$ with $z \in N(y)$ or $\{y, z\}$ with $z \in N(x)$.

We now describe the structure of a D2C graph with a dominating edge. Call $G(V, E)$ the D2C graph and uv its dominating edge. We partition the other vertices of G in four sets as follows, and as illustrated on Figure 1.5:

1. $P_{uv} = \{x \mid uv$ is critical for the pair $\{x, v\}$ or $\{x, u\}\}$
2. $S_{uv} = \{x \mid x \in N(u)$ and $x \in N(v)\}$
3. $S_u = \{x \mid x \in N(u) \setminus (P_{uv} \cup S_{uv})\}$
4. $S_v = \{x \mid x \in N(v) \setminus (P_{uv} \cup S_{uv})\}$

Furthermore, we can prove the following lemma:

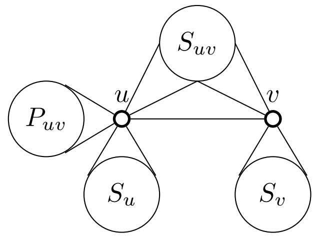


Figure 1.5: The structure of a D2C graph with a dominating edge.

Lemma 1.10. *The following statements hold:*

1. *If $P_{uv} \neq \emptyset$, then either $P_{uv} \cap N(u) = \emptyset$ or $P_{uv} \cap N(v) = \emptyset$ (in the following statements, we will assume that $P_{uv} \cap N(v) = \emptyset$);*
2. *There is no edge between P_{uv} and $N(v) \setminus \{u\}$;*
3. *If $P_{uv} = \emptyset$, then $S_{uv} = \emptyset$;*
4. *If $S_{uv} = \emptyset$, then every vertex in S_u (resp. S_v) has a neighbour in S_v (resp. S_u);*
5. *If $P_{uv} = \emptyset$, then every vertex in S_u (resp. S_v) that has at least one neighbour in S_u (resp. S_v) has a non-neighbour in S_v (resp. S_u).*

Now that we have those properties, we will partition the vertices of G into two parts X and Y, and prove that every edge within X or Y can be injectively associated to a non-edge between X and Y. This will prove that G has at most as many edges as a complete bipartite graph with parts X and Y. The partition is defined as follows:

1. $X := \{v\} \cup S_u \cup P_{uv} \cup S_{uv}$
2. $Y := \{u\} \cup S_v$

Notice that an edge yz within Y cannot be critical for the pair $\{y, z\}$ since y and z share v as a common neighbour. Thus it is necessarily critical for a pair $\{x, y\}$ or $\{y, z\}$ with $x \in X$. Assume without loss of generality that it is critical for a pair $\{x, y\}$, then we assign the edge yz to the non-edge $\overline{xy}$. This assignment function is called f, and we note $f(yz) = \overline{xy}$. The same reasoning applies for an edge in X. This construction is depicted on Figure 1.6.

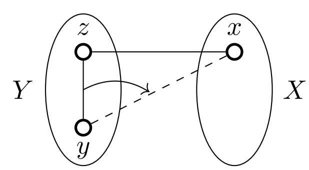


Figure 1.6: The function f assigning every edge within a part to a non-edge between the parts.

We can prove that the function f is injective. Furthermore, some non-edges between X and Y may not be assigned by the function f. We call such non-edges *f-free non-edges*, and we denote by $\mathrm{free}(f)$ the number of f-free non-edges. This gives us the exact size of G in function of its order and the number of f-free non-edges:

Lemma 1.11. *We have* $m = \left\lfloor \frac{n^2 - ||X|-|Y||^2}{4} \right\rfloor - \mathrm{free}(f) \leq \lfloor \frac{n^2}{4} \rfloor - \mathrm{free}(f)$.

In the next section, we are going to show that $\mathrm{free}(f) \geq 2$, which will prove Theorem 1.8.

1.3.2 A stronger bound on D2C graphs with a dominating edge

We will begin by stating a stronger result:

Theorem 1.12. *Let G be a non-bipartite D2C graph of order n and size m with a dominating edge uv such that $P_{uv} = \emptyset$, and let f be the associated injective function. Let c_x be the number of nontrivial connected components of diameter at least 3 in S_x (with $x \in \{u, v\}$).*

Then, $m \leq \lfloor \frac{n^2}{4} \rfloor - 2 \times \max(c_u, c_v)$.

Sketch of proof. To prove this result, we will use the function f to define an orientation on the edges of S_u and S_v (since we assume that $P_{uv} = \emptyset$, we also have $S_{uv} = \emptyset$). The orientation is defined as follows: for every edge xy within S_u or S_v such that $f(xy) = \overline{yz}$, we orient x towards y and denote the resulting arc by $\overrightarrow{xy}$. This construction is depicted on Figure 1.7. Since f is injective, each edge of S_u and S_v receives exactly one direction. In the rest of the proof, all arcs we consider are those of this f-orientation.

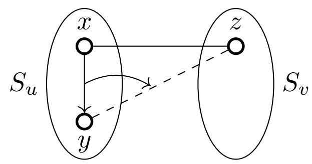


Figure 1.7: The f-orientation is defined with the function f: if $f(xy) = \overline{yz}$ then we orient x towards y.

We recall a few definitions from the theory of directed graphs. A *directed graph* is a graph where edges have been given an orientation; directed edges are called *arcs*. If x is directed towards y, we denote the arc from x to y by $\overrightarrow{xy}$. In a directed graph, we denote by $N^+(x)$, $N^+[x]$, $N^-(x)$ and $N^-[x]$ the out-neighbourhood, closed out-neighbourhood, in-neighbourhood, and closed in-neighbourhood of vertex x. A *directed cycle* is a cycle such that all arcs are directed in the same cyclic direction. A *source* is a vertex s with $N^-(s) = \emptyset$ and $N^+(s) \neq \emptyset$ while a *sink* is a vertex t with $N^+(t) = \emptyset$ and $N^-(t) \neq \emptyset$ (in other words, we consider that an isolated vertex is neither a source nor a sink). A triangle in an directed graph is *transitive* if it induces a subgraph with a source and a sink.

We then study the properties of the f-orientation. First, we prove that directed cycles yield many f-free non-edges:

Lemma 1.13. *Let $\overrightarrow{C}$ be a cycle of G that is directed with respect to the f-orientation. Then, there are at least $|\overrightarrow{C}|$ f-free non-edges incident with the vertices of $\overrightarrow{C}$.*

Thus, if a component of S_u or S_v contains a directed cycle, then its vertices are incident with at least 3 f-free non-edges. We then prove that at least 2 f-free non-edges are incident with vertices in an acyclic component of diameter at least 3.

The next lemma is key to many of the following properties of the f-orientation: if there is an arc $\overrightarrow{xy}$ such that neither x nor y is incident with an f-free non-edge, then the neighbourhood of x in the other part is exactly the neighbourhood of y in the other part plus one vertex.

Lemma 1.14. *Let $x, y \in S_u$ (resp. S_v) be two vertices such that $\overrightarrow{xy}$ is an arc of the f-orientation. If neither x nor y is incident with an f-free non-edge, then there exists a vertex $t \in S_v$ (resp. S_u) such that $N(x) \cap S_v = (N(y) \cap S_v) \cup \{t\}$ (resp. $N(x) \cap S_u = (N(y) \cap S_u) \cup \{t\}$).*

This allows us to prove that sources and sinks have f-free non-edges in their closed neighbourhood:

Lemma 1.15. *Let x be a source (resp. a sink) of the f-orientation. Then, there is at least one f-free non-edge incident with the vertices of $N^+[x]$ (resp. of $N^-[x]$).*

Thus, if there is a source and a sink with disjoint neighbourhoods in the same component, then this component is incident with at least 2 f-free non-edges. Now, assume by contradiction that there is a nontrivial component C of diameter at least 3 in S_u (without loss of generality) such that exactly one f-free non-edge is incident with C (since C is acyclic, it contains at least one source and as such is incident with at least 1

f-free non-edge). The remainder of the proof is about finding more properties of the f-orientation to find a contradiction.

Let $\mathcal{T}$ be the set of sinks in C, and r the vertex incident with the f-free non-edge. We prove the following statement, which constrain even more the structure of C:

Lemma 1.16. *The following statements hold:*

1. *Either $r \in \mathcal{T}$, or each sink in $\mathcal{T}$ has r as its unique in-neighbour;*
2. *There is exactly one source in C.*

With this structure, we can now find a contradiction. To do this, we exhibit a vertex y in S_v such that no successor of the source s and predecessor of r can be adjacent to y[3]. However, the structure of G described in Lemmas 1.10 and 1.14 and the fact that C has diameter at least 3 let us find a vertex x in C that is adjacent to y. However, since s is the only source and r the only in-neighbour of all sinks, we have a contradiction. □

Theorem 1.12 allows us to prove Theorem 1.8:

Sketch of the proof of Theorem 1.8. First, we assume that P_{uv} is empty. If there is a component of diameter at least 3 in S_u or S_v, then the result directly follows from Theorem 1.12. If all components in S_u or S_v are trivial, then the graph is bipartite and as such it is complete bipartite, a contradiction. If there are two nontrivial components (of any diameter) in S_u or S_v, then the result directly follows from Lemmas 1.13 and 1.15. If there is exactly one nontrivial component of diameter at most 2 in S_u or S_v, then a simple study allows to conclude that there are at least 2 f-free non-edges.

Then, we assume that P_{uv} is nonempty. A thorough case study allows us to find that there are always at least 2 f-free non-edges in G. □

1.4 Conclusion and perspectives

Theorem 1.8 is a first step in the study of Conjecture 1.7, the linear strengthening of the Murty-Simon Conjecture. In order to improve the result, we may try to further the method used for proving Theorem 1.12 and obtain more constraints on the f-orientation. However, this would also require to manage separately all the limit cases (components of diameter at most 2, P_{uv} nonempty). Thus, if we want to reuse this method, we will have to find more general properties on the structure of D2C graphs with a dominating edge.

The method we used to prove Theorem 1.8 is to measure the difference between a D2C graphs with a dominating edge and a complete bipartite graph. This could be generalized to other families of D2C graphs.

Another interesting question that is raised is to study whether the improved bound of Conjecture 1.7 is tight for D2C graphs with a dominating edge of order greater than 9 (for smaller order, Figure 1.4 shows several such graphs reaching this bound). As a first step in this direction, we found a D2C graphs with a dominating edge that has $\frac{(n-2)^2+15}{4}$ edges. Said graph has many twins[4], so another question is the study of twin-free D2C graphs. Indeed, many D2C graphs having a high number of edges also have many twins (such as the graphs from $\mathscr{F}_n$), so maybe the bound can be improved even further by focusing on twin-free D2C graphs.

In this chapter, we have shown that the study of the strengthened Murty-Simon Conjecture is of high interest, and that many possibilities exist to improve the bound of the Murty-Simon Conjecture. We have presented one method, that allowed us to improve the bound on D2C graphs with a dominating edge. We hope that this method can be improved and generalized to work on more families of D2C graphs. Furthermore, many interesting open questions on the D2C graphs remain yet unexplored.

[3] A *predecessor* of a vertex a is a vertex b such that there is a directed path from b to a. Similarly, a *successor* of a is a vertex b such that there is a directed path from a to b.

[4] True (resp. false) twins are vertices with the same closed (resp. open) neighbourhood. In the context of D2C graphs, there can only be false twins.

Chapter 2

Union vertex-distinguishing edge coloring

The work in this chapter was realized in collaboration with **Nicolas Bousquet**, **Éric Duchêne**, **Hamamache Kheddouci** and **Aline Parreau**. A paper has been published [21] and the work has been presented during the Bordeaux Graph Workshop 2016 and the Journés Graphes et Algorithmes 2016.

Chapter abstract

In this chapter, we study a vertex-distinguishing edge coloring: given a graph G, we use an edge coloring to identify each vertex of G with a unique label. The coloring we use assign sets of integers from 1 to k to edges, and each vertex receives the union of its incident edges' sets as a label. The parameter we study is the smallest k such that such an edge coloring can produce unique labels to each vertex. In Section 2.1, we are going to give a background on vertex-distinguishing colorings and to give a few properties of the variant we study. Then, in Section 2.2, we study the parameter on classical graph classes. Finally, in Section 2.3, we prove that the parameter can take only three values (while the parameters induced by vertex-distinguishing edge colorings usually have a logarithmic lower bound and a linear upper bound).

2.1 Introduction

2.1.1 Vertex-distingushing colorings

Distinguishing (or *identifying*) the vertices of a graph is a classical graph problem. It consists in assigning a label to each vertex of a graph. A vertex is *distinguished* (resp. *locally distinguished*) if its label is unique among all the vertices (resp. among all its closed neighbourhood). Such a labeling is *globally distinguishing* if all vertices are distinguished, and *locally distinguishing* is all vertices are locally distinguished. A natural question that arises is to find methods to generate such labelings for any graph.

A classical way to generate such labelings is to use vertex or edge colorings. The colorings are not necessarily proper[1], and indeed many vertex-distinguishing colorings have been studied under both improper and proper variants. Each of these colorings will induce a graph parameter, which is defined as the smallest integer k such that there exists such a vertex-distinguishing coloring using k colors. Furthermore, there are four possible variations (proper or improper, globally or locally distinguishing) and thus four such parameters for each coloring and way to generate the labels. We will present a few of the most well-known vertex-distinguishing colorings. Note that such colorings have been sometimes been studied under different names and notations in the literature. For more information, the interested reader can refer to Chapter 3 in [36].

The first vertex-distinguishing coloring that has been introduced is called the *point-distinguishing edge coloring* [58]. In this variant, edges are colored by integers, and each vertex receives as a label the union of its incident edges' colors. This is depicted on Figure 2.1. The associated parameter is denoted χ_S and has been extensively studied, especially on complete bipartite graphs [68, 69, 70, 88, 89] but also on paths,

[1] Recall that a vertex coloring is proper if no adjacent vertices get the same color, and an edge coloring is proper if no incident edges get the same color.

cycles, cubes, complete graphs [58] and unions of paths [28]. Its proper variant (also called *observability*) has also been studied [10, 11, 26, 24], as well as the locally distinguishing [54, 55] and the proper locally distinguishing [9, 107] variants.

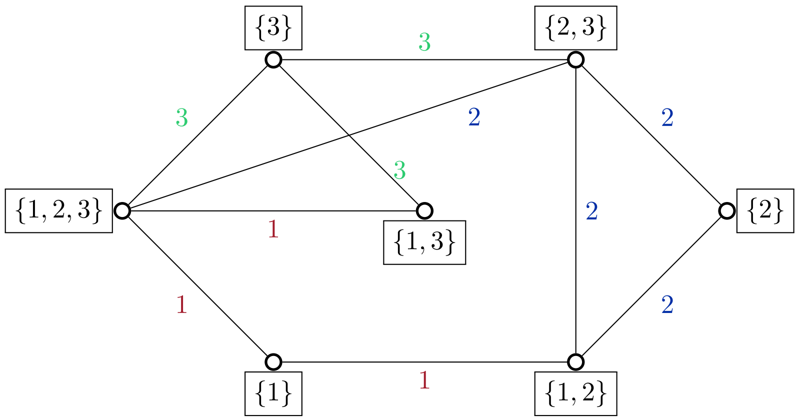


Figure 2.1: A point-distinguishing edge coloring. Vertex labels are boxed, unlike the colors assigned to edges. Three colors are used to distinguish all vertices, and we can show that no such coloring using two colors exist, thus $\chi_S(G) = 3$.

One of the most well-known vertex-distinguishing edge colorings is the *sum-distinguishing edge coloring*, first introduced in [27]. In this coloring, edges are colored by integers and each vertex receives as a label the sum of its incident edges' colors. If this coloring is well-known, it is due to its locally distinguishing variant [74], where we only want to distinguish adjacent vertices. This variant, depicted on Figure 2.2, induces a parameter denoted χ^e_Σ. It is the subject of the well-known *1-2-3 Conjecture*, which states that $\chi^e_\Sigma(G) \leq 3$ for any graph G. The work on this conjecture has been compiled in a survey [93], which also presents results on locally distinguishing colorings.

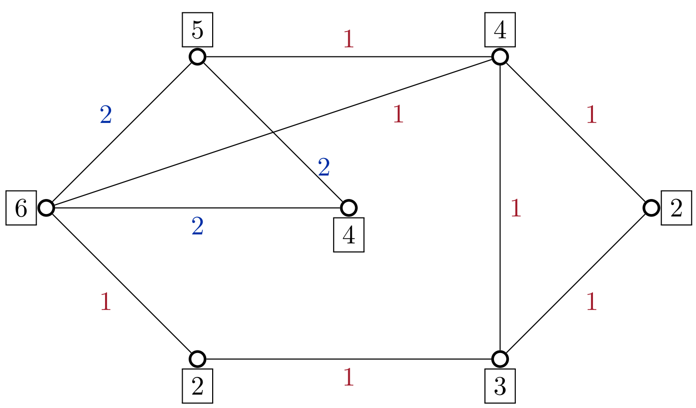


Figure 2.2: A locally sum-distinguishing edge coloring. Two colors are used, and it is conjectured that only three colors are sufficient for any graph.

Many other vertex-distinguishing edge colorings have been studied in the literature. Table 2.1 presents all the variants that have been introduced so far. As we can see, many ways to generate labels from edge colorings have been imagined, often as a way to study the 1-2-3 Conjecture.

Vertex-distinguishing vertex colorings have also been studied, although more recently [85]. In what is called an *identifying coloring*, vertices are colored by integers, and each vertex receives as a label the union of

Set distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	1985 [58]	1997 [24]
Locally distinguishing	2008 [54]	2002 [107]
Sum distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	1988 [27]	1985 [80]
Locally distinguishing	2004 [74]	2013 [43]
Multiset distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	1992 [4]	1997 [24]
Locally distinguishing	2005 [2]	2002 [107]
Sequence distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	Undefined	Undefined
Locally distinguishing	2012 [94]	Undefined
Product distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	Undefined	Undefined
Locally distinguishing	2008 [98]	2017 [79]
Set's diameter distinguishing		
	Improper coloring	**Proper coloring**
Globally distinguishing	2012 [102]	Undefined
Locally distinguishing	2012 [91]	Undefined

Table 2.1: Summary of the various vertex-distinguishing edge colorings in the literature, along with the year they were introduced. We can see that the subject has attracted a lot of attention, especially in the early 2000s.

its closed neighbourhood's colors. This is depicted on Figure 2.3. The associated parameter is denoted χ_{id}. Locally distinguishing and proper variants have also been studied [3, 39, 45, 50, 82].

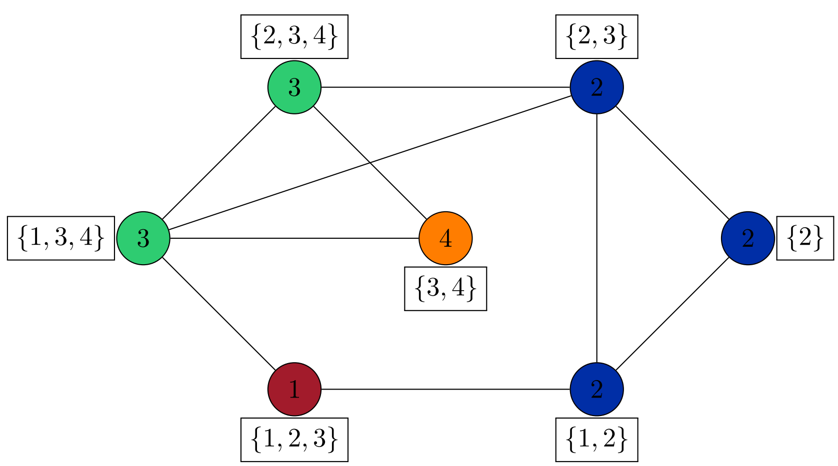


Figure 2.3: An identifying coloring using four colors.

As we can see, the vertex-distinguishing colorings have been extensively studied. In this chapter, we will study a new vertex-distinguishing edge coloring, which has links with other distinguishing colorings we mentioned earlier. An interesting facet of this new variant and the associated parameter is that the gap between its lower and upper bounds is constant, unlike most other globally distinguishing colorings for which the lower bound of the parameter is logarithmic while the upper bound is linear.

2.1.2 Union vertex-distinguishing edge coloring

We will use an edge coloring to distinguish the vertices of a graph. The edges are colored by sets of integers, and a vertex receives as a label the union of its incident edges' sets. More formally, for a graph $G(V, E)$, the edge coloring is a function $c : E \to 2^{\{1,\dots,k\}}$ assigning to each edge a nonempty subset of $\{1, \dots, k\}$ (we say that c uses k colors). The label of a vertex u is defined as follows:

$$\mathrm{id}_c(u) = \bigcup_{v \ t.q. \ uv \in E} c(uv)$$

The edge coloring c is *union vertex-distinguishing* if, for all distinct vertices u and v, $\mathrm{id}_c(u) \neq \mathrm{id}_c(v)$. An example of such a coloring is depicted on Figure 2.4. Note that a union vertex-distinguishing edge coloring can only be defined on graphs with no connected component of size 1 or 2 (since otherwise those vertices could not be distinguished). Thus, all the graphs we will consider in this chapter will have connected components of size at least 3.

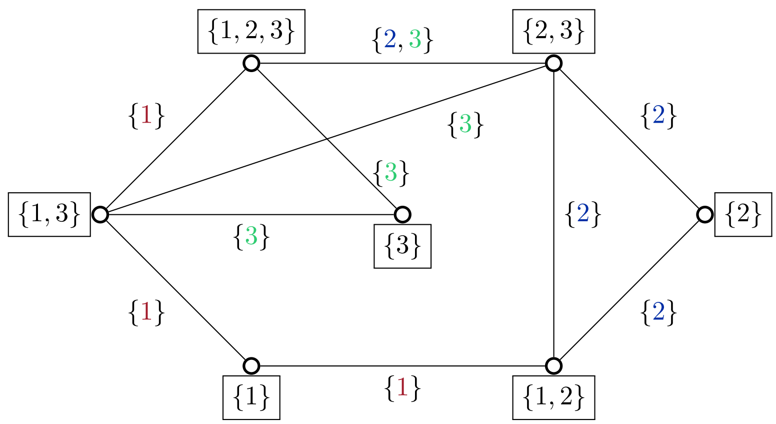


Figure 2.4: A union vertex-distinguishing edge coloring using three colors.

The associated parameter, called the *union vertex-distinguishing number*, is defined as the smallest integer k such that a union vertex-distinguishing edge coloring exists, and is denoted by $\chi_\cup$. There are links between $\chi_\cup$ and the parameters χ_S and χ_{id} that were defined in the previous section:

Proposition 2.1. *For any graph G, $\chi_\cup(G) \leq \min(\chi_S(G), \chi_{id}(G))$.*

Furthermore, we have a first natural lower bound for $\chi_\cup$:

Proposition 2.2. *For any graph $G(V, E)$, $\chi_\cup(G) \geq \lceil \log_2(|V| + 1) \rceil$.*

If a graph $G(V, E)$ verifies $\chi_\cup(G) = \lceil \log_2(|V| + 1) \rceil$, then we say it can be *optimally colored*. A natural question is to study if there are any graphs that can be optimally colored, and if it is the case then which ones. Furthermore, we want to have an insight on the gap between the optimal and the actual value of $\chi_\cup$ for graph classes for which the value of $\chi_\cup$ is the highest possible. As a first step in this study, we prove that if we know the value of $\chi_\cup$ for a graph, then adding as many edges as we want only costs one color. We call a graph obtained by adding edges to G an *edge-supergraph* of G. The idea of the proof of this lemma is presented on Figure 2.5.

Lemma 2.3. *Let G be a graph. If H is an edge-supergraph of G, then $\chi_\cup(H) \leq \chi_\cup(G) + 1$.*

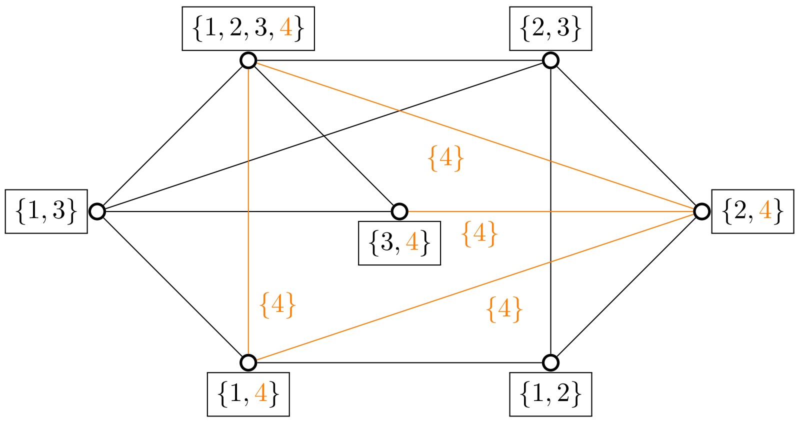


Figure 2.5: This graph is an edge-supergraph of the graph shown on Figure 2.4 (the colors on the edges of the original edges have been removed to make the drawing clearer). Coloring the new edges with a new color does not change the fact that the coloring was union vertex-distinguishing.

2.2 Graph classes

In this section, we study the value of $\chi_\cup$ for several classical graph classes. The main interest in studying this value is that, thanks to Lemma 2.3, if we know the exact value of $\chi_\cup$ for a graph G, then we can bound the value of $\chi_\cup$ for any graph admitting G as a subgraph.

The first class that we study are paths:

Theorem 2.4. *A path can be optimally colored.*

Sketch of proof. We actually prove a stronger result: a path P_n with vertices $u_1, \ldots, u_n$ such that $m = \lceil \log_2(n+1) \rceil$ has a union vertex-distinguishing edge coloring c using m colors such that:

(i) $\mathrm{id}_c(u_1) = \{1\}$;

(ii) $\mathrm{id}_c(u_n) = \{m\}$;

(iii) If there exists j such that $\mathrm{id}_c(u_j) = \{1, m\}$, then $j = n-1$.

This is proved by induction on n. The case $n = 3$ is trivial. For the other values of n, we let $n = 2^k + \ell$ (with $0 \leq \ell < 2^k$) and use c_k the optimal coloring of P_{2^k-1} and c_ℓ the optimal coloring of P_ℓ to construct c. The constructions vary depending on the value of ℓ and are depicted on Figure 2.6. $\square$

Lemma 2.3 then gives us the following result (recall that a path or cycle is hamiltonian if it passes through every vertex exactly once, and that a graph is hamiltonian if has a hamiltonian cycle and semi-hamiltonian if it has a hamiltonian path and no hamiltonian cycle):

Corollary 2.5. *Let $G(V, E)$ be a hamiltonian or semi-hamiltonian graph. Then $\chi_\cup(G) \leq \lceil \log_2(|V|+1) \rceil$.*

However, some hamiltonian graphs can be optimally colored:

Theorem 2.6. *A cycle can be optimally colored, except C_3 and C_7 which need one more color.*

Other graph classes can be optimally colored:

Theorem 2.7. *A complete binary tree can be optimally colored.*

Theorem 2.8. *A star subdivided at most once can be optimally colored.*

At this point, it may seem like most graphs can be optimally colored. However, we find an infinite family of graphs that cannot be optimally colored:

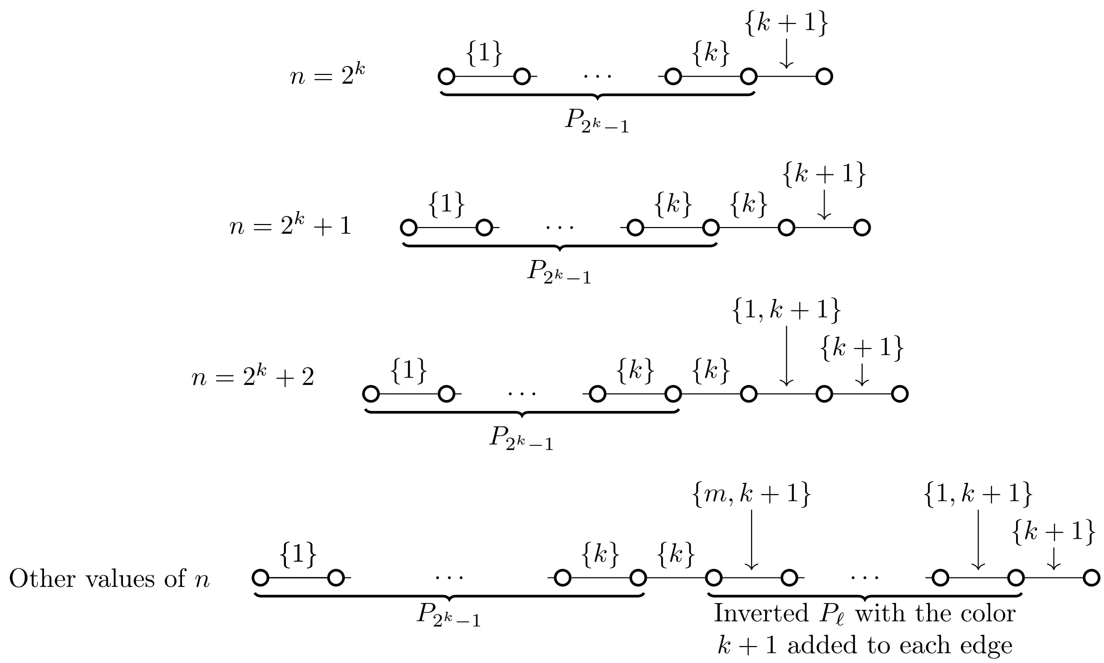


Figure 2.6: Construction of the optimal coloring c of P_n from the optimal colorings of P_{2^k-1} and P_ℓ.

Theorem 2.9. *Let $n \geq 3$ be an integer. If n is a power of 2, then the complete graph K_n can be optimally colored. Otherwise, $\chi_\cup(K_n) = \lceil \log_2(n+1) \rceil + 1$.*

Sketch of proof. Let $n \geq 3$ be an integer, and $k = \lceil \log_2(n+1) \rceil$. If K_n has a union vertex-distinguishing edge coloring c using k colors, then for any two vertices u and v of K_n we have $\mathrm{id}_c(u) \cap \mathrm{id}_c(v) \neq \emptyset$ since u and v are necessarily neighbours. Since vertices are labeled by nonempty subsets of $\{1, \ldots, k\}$, this implies that n cannot be larger that the maximal number of nonempty subsets of $\{1, \ldots, k\}$ having a nonempty intersection. However, such a number is 2^{k-1}. This means that if K_n can be optimally colored then n is a power of 2. We now need to prove that this graph can indeed be optimally colored, and that we only need one more color for the other values of n.

If n is a power of 2 then we lexicographically order all nonempty subsets of $\{1, \ldots, k\}$ containing 1 by cardinality and sum, and call them $v_1, \ldots, v_n$. Thus, we have $v_1 = \{1\}$, $v_2 = \{1, 2\}$, $\ldots$, $v_n = \{1, \ldots, k\}$. Furthermore, we denote by $u_1, \ldots, u_n$ the vertices of K_n. Now, we define c as follows: $c(u_n u_i) = v_i$ for $1 \leq i \leq n-1$ and $c(u_i u_j) = \{1\}$ for $1 \leq i, j \leq n-1$ with $i \neq j$. This is depicted on Figure 2.7 for $n = 8$. It follows that $\mathrm{id}_c(u_i) = v_i$, and as such the vertices of K_n are distinguished.

Otherwise, K_n cannot be optimally colored and is an edge-supergraph of a star. Since a star can be optimally colored by Theorem 2.8, the result holds by Lemma 2.3 and by the discussion above. $\square$

2.3 A general upper bound for the union vertex distinguishing number

The graphs that we studied in the previous section all had a union vertex-distinguishing edge coloring using the optimal number of colors or the optimal number of colors plus one. This raises the question of the upper bound on $\chi_\cup$. We prove the following:

Theorem 2.10. *For any graph $G(V, E)$, $\chi_\cup(G) \leq \lceil \log_2(|V| + 1) \rceil + 2$.*

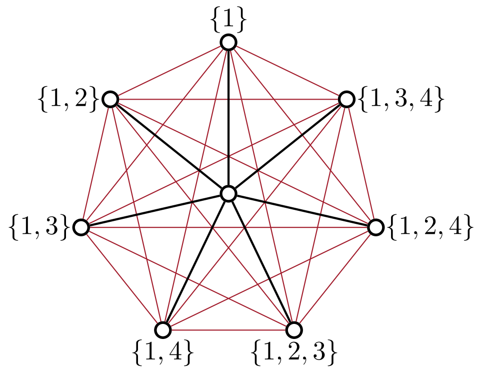


Figure 2.7: A union vertex-distinguishing edge coloring of K_8 using 4 colors. The red edges are colored by the singleton $\{1\}$. The other edges are colored by the set at their endpoint. Each vertex is distinguished by a nonempty subset of $\{1, 2, 3, 4\}$ containing 1, the central vertex being distinguished by $\{1, 2, 3, 4\}$.

Sketch of proof. The idea of the proof is to extract a graph H of which G is an edge-supergraph, to find a union vertex-distinguishing edge coloring of H using the optimal number of colors plus one, and then to add back the edges at the cost of one color with Lemma 2.3.

The graph H that we choose is a forest of stars subdivided at most once. We prove that extracting such a forest is possible for any graph:

Lemma 2.11. *For any graph G, there is a forest of stars subdivided at most once H such that G is an edge-supergraph of H.*

Theorem 2.8 ensures that each of the individual stars can be optimally colored. We now need to see how many colors are necessary for the forest, which is the disjoint union of the individual stars. We call a graph with order between 2^k and $2^{k+1} - 1$ a *k-graph*. We prove that the disjoint union of two k-graphs (which is a $(k+1)$-graph) can be optimally colored:

Lemma 2.12. *Let H_1 and H_2 be two k-graphs that can be optimally colored. Then their disjoint union $H_1 \cup H_2$ can be optimally colored.*

This helps us to prove that the disjoint union of ℓ graphs that can be optimally colored admits a union vertex-distinguishing edge coloring using the optimal number of colors plus one:

Lemma 2.13. *Let $H_1, \ldots, H_\ell$ be graphs that can be optimally colored. We call H the disjoint union of $H_1, \ldots, H_\ell$. Then, we have $\chi_\cup(H) \leq \lceil \log_2(|V(H)| + 1) \rceil + 1$.*

This proves Theorem 2.10. The method is depicted on Figure 2.8. □

2.4 Conclusion and perspectives

Theorem 2.10 and Proposition 2.2 allow us to prove the following:

Theorem 2.14. *For any graph $G(V, E)$, we have:*

$$\lceil \log_2(|V| + 1) \rceil \leq \chi_\cup(G) \leq \lceil \log_2(|V| + 1) \rceil + 2.$$

This result is interesting, since generally parameters induced by vertex-distinguishing colorings have a logarithmic lower bound and a linear upper bound, both of which are reached by specific graphs. The different results that are known on the parameter $\chi_\cup$ are displayed on Table 2.2. As we can see, we do not know any graph for which the value of $\chi_\cup$ reaches the upper bound, and indeed we conjecture that no such graph exists:

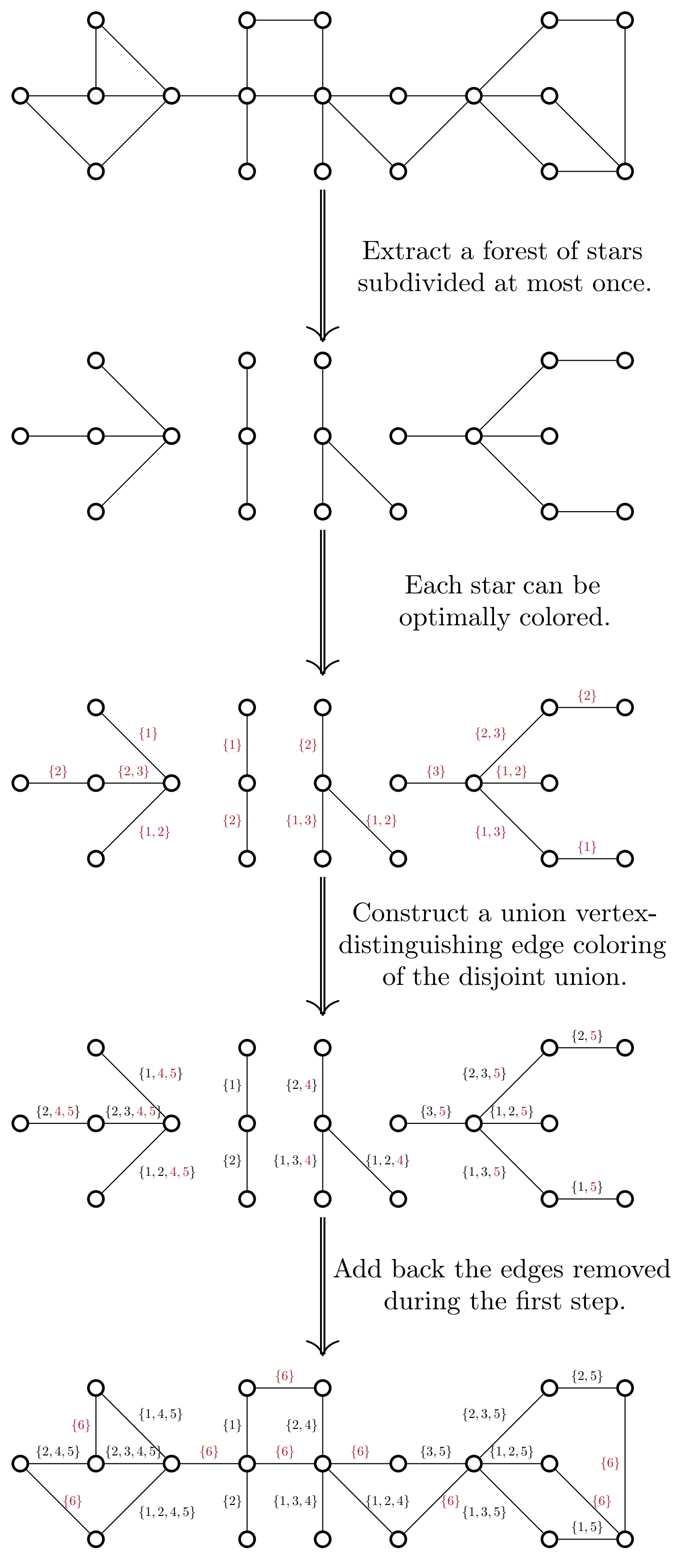


Figure 2.8: Illustration of the method we use to find a union vertex-distinguishing edge coloring of G using $\lceil \log_2(|V(G)|+1) \rceil + 2$ colors (at each step, the colors we add are in red).

Conjecture 2.15. *For any graph $G(V, E)$, we have:*

$$\lceil \log_2(|V|+1) \rceil \leq \chi_\cup(G) \leq \lceil \log_2(|V|+1) \rceil + 1.$$

A way to tackle this conjecture would be to prove that any tree can be optimally colored. Since all graphs have a spanning tree, Lemma 2.3 would give us the result. Another way to study it would be to improve the bound on the forests of stars subdivided at most once.

Finally, locally distinguishing or proper variants could also be considered, and it would be interesting to know if the small gap between the upper and lower bound would be maintained in the proper version. Furthermore, the complexity of Decision Problems 2.16 and 2.17 is still open on many graph classes, so furthering the study in this direction could yield interesting results.

Decision Problem 2.16.

Optimally colored

INPUT: A graph $G(V, E)$.

QUESTION: Do we have $\chi_\cup(G) = \lceil \log_2(|V(G)|+1) \rceil$?

Decision Problem 2.17.

Union vertex-distinguishing number

INPUT: A graph $G(V, E)$ and an integer $k \in \{0, 1, 2\}$.

QUESTION: Do we have $\chi_\cup(G) = \lceil \log_2(|V(G)|+1) \rceil + k$?

Family of G	Value of $\chi_\cup(G)$	Reference
Paths P_n (for $n \geq 3$)	$\lceil \log_2(\lvert V(G)\rvert+1) \rceil$	Theorem 2.4
Cycles C_n (for $n \geq 4, n \neq 7$)		Theorem 2.6
Complete binary trees		Theorem 2.7
Stars subdivided at most once		Theorem 2.8
Complete graphs K_{2^k}		Theorem 2.9
Cycles C_3, C_7	$\lceil \log_2(\lvert V(G)\rvert+1) \rceil + 1$	Theorem 2.6
Complte graphs K_n (for $n \neq 2^k$)		Theorem 2.9
Hamiltonian graphs	$\leq \lceil \log_2(\lvert V(G)\rvert+1) \rceil + 1$	Corollary 2.5
Forest of stars subdivided at most once		Lemma 2.12
Any graph	$\leq \lceil \log_2(\lvert V(G)\rvert+1) \rceil + 2$	Theorem 2.10

Table 2.2: The parameter $\chi_\cup$ can only take three values for any graph G. This table presents, for each of those three values, graph classes for which $\chi_\cup$ takes this value or for which $\chi_\cup$ is bounded by this value. Note that we do not know any graph G for which $\chi_\cup(G) = \lceil \log_2(|V(G)|+1) \rceil + 2$.

Intermission

Chapter 3

From graphs to games

Chapter abstract

In this chapter, we will present some articulations between graphs and games. In particular, we will see how one can derive a game problem from a graph problem. First, in Section 3.1, we will talk about Maker/Breaker games. Then, Section 3.2 will see us introduce the construction and partition games as well as the parametes that these games induce. In Section 3.3 we will briefly mention the achievement and avoidance games as well as the combinatorial games that will be detailed in Chapter 4, before further explaining the main problems that are studied in games in Section 3.4. Finally, we will illustrate some of those problems in Section 3.5, where we will study the Grundy coloring game.

3.1 From resilience to games

In Chapter 1, we studied a problem of *criticality*: a graph is critical in relation to some property (or parameter) if it verifies the property (or if the parameter has a given value) but any small operation done to the graph generates a graph that does not verify the property (or for which the value of the parameter is different). A related notion is the *resilience* of a property (or parameter), which is a measure of how this property (or parameter) behaves when a graph is modified by small operations. A property or parameter is said to be resilient to a graph operation if applying this operation to a graph verifying the property (or for which the parameter has a given value) generates a graph which still verifies the property (or for which the value of the parameter is unchanged).

An interesting way to study the resilience is to simulate an opposition by modeling the property or parameter with a two-player, perfect information game. In such games, we will assume that the two players play perfectly, in the sense that they have an infinite computing power and since the games we define are deterministic, they can always determine the best possible play.

The first family of games that we can define this way are the *positional Maker/Breaker games*. Those games, which are a very efficient way to study the resilience of a graph property, are formally defined the following way:

Definition 3.1 (Positional Maker/Breaker game [66])**.** *Let X be a finite set and $\mathcal{F} \subseteq 2^X$ a family of subsets of X. In a* positional Maker/Breaker game *on the hypergraph $(X, \mathcal{F})$*[1]*:*

- *The set X is called the* board*; the elements of $\mathcal{F}$ are called the* winning sets*;*
- *The players are called Alice and Bob (or Maker and Breaker) and they play alternatively, by definition and if not mentioned otherwise Alice begins;*
- *A move consists in owning a non-owned element of X;*

[1] A hypergraph $H(V, \mathcal{F})$ is defined by a vertex set V and a hyperedge set $\mathcal{F} \subseteq 2^V$.

- *Alice wins if, by the end of the game, she owns the totality of a winning set, Bob wins if he owns at least one element in each winning set.*

In summary, Alice and Bob select vertices of a hypergraph. The goal of Alice is to own all the vertices in one hyperedge, while the goal of Bob is to own a subset S of vertices such that every hyperedge contains at least one vertex from S. Note that the Maker/Breaker games can never end in a draw: either Alice owns a winning set and she wins, or she does not and Bob wins.

Definition 3.1 gives us a general frame for Maker/Breaker games. This allows us to easily adapt a property of graph objects into a positional Maker/Breaker game by using the following construction:

Construction 3.2. *Let $\mathcal{P}$ be a graph property. The positional Maker/Breaker game derived from $\mathcal{P}$ is the positional Maker/Breaker game on $(X, \mathcal{F})$ where:*

- *X is the set of the graph elements (vertices, edges, ...) that are affected by the property;*
- *$\mathcal{F}$ is the family of sets of elements such that $\mathcal{P}$ is verified.*

Construction 3.2 is illustrated by Examples 3.3 and 3.4, which are based respectively on the properties "being a dominating set" and "being a hamiltonian cycle". In the first example, we have $X = V$ and $\mathcal{F} = \{S \subseteq V \mid S \text{ dominates } V\}$. In the second example, $X = E$ and $\mathcal{F} = \{C \subseteq E \mid C \text{ is a hamiltonian cycle}\}$.

Example 3.3. *In the positional Maker/Breaker domination game [37] is played on a graph $G(V, E)$. Alice and Bob alternate selecting vertices from V. The game ends when all vertices are selected. Alice wins if she owns a dominating set, otherwise Bob wins.*

Example 3.4. *The positional Maker/Breaker hamiltonicity game [30] is played on a complete graph K_n. Alice and Bob alternate selecting edges from E. The game ends when all edges are selected. Alice wins if she can create a hamiltonian cycle (a cycle that goes through every vertex exactly once) with some of her edges, otherwise Bob wins.*

The main problem on Maker/Breaker games is Decision Problem 3.5, which asks whether Alice wins whatever Bob's strategy. Note that asking the same question for Bob is equivalent, since one wins if the other loses. An other problem is the determination of Alice's or Bob's strategy. We can see that, while related, those two problems are completely distinct. While a winning strategy for Alice or Bob is a certificate for Decision Problem 3.5, it is possible to know whether Alice or Bob wins without knowing their strategy.

Decision Problem 3.5.

POSITIONAL MAKER/BREAKER GAME
INPUT: A graph G.
QUESTION: Does Alice win the game on G, whatever Bob does?

While positional Maker/Breaker games have been studied as early as the 1960's and 1970's [56, 38, 30, 14], their study has recently seen a rise since their ideas have been used to study the probability that Alice can select a set of vertices or edges from a complete or random graph such that this set verifies some property [101]. However, as mentioned in several books on the topic [66], Maker/Breaker games are in general heavily biased towards one player. For example, in the Maker/Breaker domination game presented in Example 3.3, Alice has a huge advantage since Bob needs the graph to have an order exponential in the minimum degree in order to win [37]. A first way to rebalance those games is to define a bias:

Definition 3.6 ([66])**.** *Let p, q be two positive integers. The* positional Maker/Breaker game with bias $(p{:}q)$ on the hypergraph $(X, \mathcal{F})$ *is the positional Maker/Breaker game on the hypergraph $(X, \mathcal{F})$ in which Alice selects p elements from X during her turn, and Bob selects q elements from X during his turn.*

The study of biased Maker/Breaker games is generally a probabilistic one: one looks for the worst bias such that Alice has a high probability of winning the game. However, we are going to detail a deterministic way to rebalance Maker/Breaker games. In this specification, Alice and Bob will build the same sets, and Bob will try to block or stall Alice. Such games will be detailed in the next section.

3.2 From graph parameters to game parameters

In the games that we are going to define in this section, Alice and Bob will construct the same sets. The difference between them will then be the winning condition. Such games are derived from graph properties and parameters. Schmidt [92] proposed a way to define games from graph properties and parameters, which is based on the following partition of graph problems, in which many of them fall:

Definition 3.7 ([92])**.** *In a* construction problem*, we try to find a subset of graph elements verifying some property. The associated* construction parameter *is the smallest (or biggest) integer k such that a set of size k verifies the property.*

In a partition problem*, we try to partition the elements of the graph into k sets verifying some property. The associated* partition parameter *is the smallest (or biggest) integer k such that this partition is possible.*

Example 3.8 illustrates this definition with three problems.

Example 3.8. *In the domination problem, we look for the smallest integer k such that there is a set of k vertices that dominate the graph. Thus, the domination number is a construction parameter.*

In the maximum independent set problem, we look for the biggest integer k such that there is a set of k independent vertices. Thus, the independence number is a construction parameter.

In the coloring problem, we partition the vertices of the graph in k sets such that each set is independent, and we look for the smallest k such that this partition is possible. Thus, the coloring number is a partition parameter.

This partition of graph problems allow us to give a general frame for two categories of games where Alice and Bob work on the same sets but have conflicting objectives. Those games are derived from graph parameters:

Construction 3.9 (Construction game [92])**.** *Let $\mathcal{P}$ be a construction parameter and k be a positive integer. The* construction game *associated to $\mathcal{P}$ is the game where:*

- *The game is played on a graph G;*
- *At their turn, Alice and Bob add to a set X elements from G such that each move must "improve" the construction (in the sense that no move can be useless);*
- *The game ends when there is no move left, or when k moves are played if $\mathcal{P}$ is a parameter when we try to minimize the size of the set;*
- *Alice wins if, at the end, X verifies the property associated to $\mathcal{P}$. Furthermore, if $\mathcal{P}$ is a parameter when we try to maximize the size of the set, then she wins if and only if $|X| \geq k$. Otherwise, Bob wins.*

Construction 3.10 (Partition game [92])**.** *Let $\mathcal{P}$ be a partition parameter and k be a positive integer. The* partition game *associated to $\mathcal{P}$ is the game where:*

- *The game is played on a graph G;*
- *At their turn, Alice and Bob add to a set among k sets $X_1, \ldots, X_k$ elements from G under some conditions;*
- *The game ends when there is no move left;*
- *Alice wins if, at the end, the X_is verify the property associated to $\mathcal{P}$. Otherwise, Bob wins.*

Thus, there are two subfamilies of construction games. In the first, which is when we try to minimize the construction parameter, Alice and Bob alternate adding elements to a set (the condition that every move has to "improve" the set exists so that Bob cannot play useless moves). Alice tries to have the set satisfy the property in k moves or less, while Bob tries to stall. This is illustrated with Example 3.11, which applies Construction 3.9 to the domination number. In the second, which is when we try to maximize the construction parameter, Alice and Bob again add elements to a set, but this time Alice will try to stall and make the set as big as possible, while Bob will try to end the game before the set is of size k. However, he is not allowed to "break" the property. This is illustrated with Example 3.12, which applies Construction 3.9 to the independence number. Partition games are more floating, since moves may affect several elements at the same time, but generally Alice's goal is to cover the whole graph with the sets, or to have them verify some property. This is illustrated with Example 3.13, which applies Construction 3.10 to the coloring number. Note that, like Maker/Breaker games, there can be no draw in construction and partition games: either Alice reaches her goal and she wins, or Bob wins.

Example 3.11. *In the Dominator/Staller domination game [23], Alice and Bob take turns adding to a set X a vertex that is not dominated yet. The game ends when the graph is dominated, or when k moves have been played. Alice wins if the graph is dominated, otherwise Bob wins.*

Example 3.12. *In the independence game [86], Alice and Bob take turns adding to a set X a vertex that is not adjacent to a vertex in X. The game ends when X is a maximal independent set. Alice wins if X is of size at least k, otherwise Bob wins.*

Example 3.13. *In the coloring game [48], Alice and Bob take turns assigning vertices in a set X_i, in such a way that each set is an independent set. The game ends when no move is available. Alice wins if the X_i's cover the whole graph, otherwise Bob wins.*

Decision Problem 3.14 is the decision problem associated with construction and partition games. As with Decision Problem 3.5, the question is to ask whether Alice or Bob wins.

Decision Problem 3.14.

Construction or partition game

INPUT: A graph G and an integer k.

QUESTION: Does Alice win the game on G, whatever Bob does?

Furthermore, this decision problem induces an optimisation problem: at which point the value of k can be small or big while ensuring Alice's victory. This allows us to define game parameters (also called optimisation parameters):

Construction 3.15. *Let $\mathcal{P}$ be a construction or partition parameter. The associated* game parameter *$\mathcal{P}_g$ is the smallest or biggest value of k (depending on whether we want to minimize or maximize the value of $\mathcal{P}$) such that Alice wins the construction or partition game associated to $\mathcal{P}$.*

For example, we can derive the coloring number χ to define the game coloring number χ_g, which is the smallest number of colors that Alice needs to win the coloring game described in Example 3.13.

The main problems in the study of game parameters is to determine the value of the parameter, and to find the complexity of the associated decision problem described by Decision Problem 3.16.

Decision Problem 3.16.

Game parameter

INPUT: A graph G, an integer k.

QUESTION: Do we have $\mathcal{P}_g(G) \geq k$ (or $\mathcal{P}_g \leq k$)?

However, as mentioned above, those games always end with Alice's or Bob' victory. In the next section, we will see how we can use graph properties or parameters to create games that can end in a draw.

3.3 Further into the games' territory

A third way to create a game from a graph property or parameter is to have Alice and Bob follow the same goal, the winner being the first one reaching this goal. This goal can be either positive (they want to realize some construction) or negative (they want to avoid some construction). The games are called respectively achievement and avoidance games, but they can also be considered as positional Maker/Maker or Breaker/Breaker games. Example 3.17 is an example of an avoidance game derived from the edge coloring problem.

Example 3.17. *In* SIM *[96], Alice and Bob alternate coloring, each with their own color, uncolored edges from the complete graph K_6. The first one to get a triangle of their color loses.*

In the original definition of SIM, the Turán Theorem ensures that either Alice or Bob wins, since it is played on K_6. However, playing this game on other graphs may result in a draw. Indeed, if neither player can realize the construction, or force their opponent to realize it, then none of them wins. This makes such games harder to study, so it is interesting to add another win condition. For example, by stating that the first player unable to play a move loses, we obtain a defining feature of combinatorial games.

Combinatorial games will be fully defined in Chapter 4, so for now we will simply say that these are two-player games where the winner is fully determined by the last play: depending on the convention, the player who plays the last move either wins or loses. Thus, those games are achievement or avoidance games where the goal is either to play the last move or to force the opponent to play the last move. Example 3.18 is a combinatorial game defined from the coloring number.

Example 3.18. *The version of* Col *[31] introduced in [16]) is played on a graph with a set of k colors. Alice and Bob properly color vertices from the graph, the first player unable to color a vertex loses the game.*

As was the case for Maker/Breaker, construction and partition games, the main questions that are studied in achievement, avoidance and combinatorial games are the decision problem (does Alice win?), the winning strategy and the complexity of determining those. Those will be summed up in the next section.

3.4 General problems in games

As we saw in the previous sections, the main problem when studying games is to answer to the decision problem: does Alice win the game, whatever Bob does (see Decision Problems 3.5 and 3.14). A problem that is linked is the question of the winning strategy. Furthermore, in the case of construction and partition games, the problem of studying the game parameter is also very important (see Decision Problem 3.16).

In the general case, a certificate validating that Alice wins a game on a given graph (whether it is positional, combinatorial, or a construction or partition game) is a strategy. In other words, we have to prove that *there exists* a move such that, *for all* Bob's moves, Alice is still in a winning position. By repeating this, we have a certificate of the form $\exists x_1 \forall x_2 \exists x_3 \dots \forall x_{n-1} \exists x_n$ followed by a boolean formula with the x_i's corresponding to the moves as variables. Thus, certificates for the decision problem in games often are quantified boolean formulas, so those problems are as hard as QBF-SAT[2] and as such are at most in PSPACE [41]. This is a contrast with the decision problems in graphs, for which the certificates often are basic boolean formulas and as such are as hard as the variants of SAT that are in NP.

Thus, there is a complexity gap between graph decision problems and game decision problems. This explains why game problems are generally considered as very hard. This is illustrated by Table 3.1, which depicts the increasing hardness of decision problems for variants of the coloring problem in graphs and games.

Decision problem	Chromatic number (decide if $\chi(G) \leq k$)	Col (decide if Alice wins)	Game chromatic number (decide if $\chi_g(G) \leq k$)
Complexity	NP-complete [49, 75]	PSPACE-complete [16, 90]	Open

Table 3.1: Evolution of the complexity while deriving games from a graph parameter.

The question of the value of game parameters is similar: while deciding the value of a graph parameter is often NP-hard, the same problem for game parameters is often PSPACE-hard. Another question is often considered: does the value of the parameter change whether Alice or Bob plays the first move?

Another problem, which is surprisingly difficult to tackle, is the question of monotonicity. For example, in the coloring game, if Alice has a winning strategy with k colors, does she still have one with $k+1$ colors? While seemingly easy, this question is still open. Even the relaxation, where Alice is allowed $f(k)$ colors for

[2]It is actually a variant of QBF-SAT where the quantifiers $\exists$ and $\forall$ alternate. However, this does not change the complexity of the problem, which is still PSPACE-complete.

some function f, is open. Again, there is a huge complexity gap between graph and game problems: if there exists a k-coloring of a graph, then there exists a $(k+1)$-coloring of the same graph.

Other hard questions are to determine extremal or critical families for a given game parameter, as well as studying the variation of the parameters under basic graph operations. Those problems are beginning to arise, and they generally are more difficult than the associated graph questions.

In the next section, those questions will be illustrated on a specific example, the Grundy coloring game.

3.5 An example: the game Grundy number

The coloring game defined in Example 3.13 is known to be difficult to study. Indeed, one of the only known ways to study the chromatic number is to study a colorblind variant. In this variant, known as the marking game and parameterized by an integer k, Alice and Bob alternate selecting a vertex that has at most k already selected neighbours. Alice aims to have the whole graph selected. The smallest integer k such that she has a winning strategy is called the coloring number, and is denoted $\mathrm{col}_g(G)$. It is easy to see that if Alice has a winning strategy in the marking game with integer k, then she can replicate her strategy by greedily coloring vertices and win the coloring game with k colors, and as such $\chi_g(G) \leq \mathrm{col}_g(G)$. The study of this variant is the main way to study the coloring game.

When facing this kind of difficulty, it is often useful to consider simplifications. In the coloring game, Alice and Bob select vertices and then color them. Thus, two simpler variants appear: one where they only select vertices, which are then colored with a given rule, and one where the order of the vertices is fixed and the players only choose the color. We will focus on the first simplification, and greedily color the vertices as they are selected by Alice and Bob. The game that we will study is the partition game derived from the Grundy number.

The greedy coloring of a graph consists in defining an order on the vertices, and then following the order and coloring the vertices with the first available color (colors are integers). The Grundy number[3] [29] of a graph G, denoted $\Gamma(G)$, is the biggest number of colors used in a greedy coloring of G. In other words, studying the Grundy number is equivalent to finding the worst possible order to greedily color the vertices of a graph. The gap between the coloring number and the Grundy number is a measure of the difference between an optimal coloring and an algorithmically simple coloring.

By using Constructions 3.10 and 3.15, we can define the Grundy coloring game and its associated parameter, the game Grundy number Γ_g (there are two variants: Γ_g^A is the game Grundy number when Alice begins, and Γ_g^B is the game Grundy number when Bob begins). In the Grundy coloring game, Alice and Bob alternate selecting vertices, which are then greedily colored. The game ends when no move is available, and Alice wins if the graph is fully colored. The game Grundy number is the smallest number of colors such that Alice wins.

Note that, for this parameter, the question of the monotonicity that we mentioned earlier is easily solved: the game Grundy number being the biggest number of colors that Bob can enforce in a greedy coloring, adding more colors does not increase the number of options.

The Grundy coloring game has been introduced by Havet and Zhu [59], who studied the parameter on several graph classes (note that if $\mathcal{H}$ is a graph family, then $\Gamma_g(\mathcal{H}) = \max_{G\in\mathcal{H}} \Gamma_g(G)$). They first give a condition for which the values of Γ_g^A and Γ_g^B are equal, then note several trivial bounds: the chromatic number χ (which measures the best possible coloring) is a lower bound for Γ_g while the Grundy number Γ (which measures the worst possible greedy coloring) is an upper bound for Γ_g. Furthermore, we also have $\Gamma_g(G) \leq \mathrm{col}_g(G)$. However, a more thorough study of some graph classes has given a better bound for the game Grundy number compared to the coloring number (the bounds are lowered by 1):

Theorem 3.19 (Havet, Zhu [59])**.** *For all forest F, $\Gamma_g(F) \leq 3$.*
For all partial 2-tree P, $\Gamma_g(P) \leq 7$.

Building upon a result from Gyárfás and Lehel [53] on the coloring number, we find a better bound of $\omega(G)$ for the game Grundy number of split graphs[4], which is also an improvement of 1:

Proposition 3.20 (D.)**.** *Let G be a split graph with clique number ω. Then, $\Gamma_g^A(G) = \Gamma_g^B(G) = \omega$.*

[3] Also called first-fit coloring number, or greedy coloring number, or ochromatic number.

[4] Recall that a split graph is a graph where vertices can be partitioned into a clique and an independent set.

Proof. Alice will maintain the following invariant: "*when a color k is played for the first time, either it is on a vertex of the clique or a vertex of the clique will be colored with k with the next move*". Since a vertex may only be colored by k if it has neighbours already colored by the colors 1 to $k-1$, this means that the vertices from the clique will be colored with colors $1, 2, \ldots, \omega$ and thus that no bigger color will be used. To ensure this, Alice will always play on the clique as long as it is not fully colored.

The invariant is trivially true for the first color: if Alice begins then she only has to play on the clique. Otherwise, if Bob wants to break it then he has to play on the independent set, and at least one vertex from the clique will not be adjacent to the vertex he played on. Alice only has to select this vertex, which will be colored by 1.

Assume now that the invariant was maintained for the first $k-1$ colors, and that the current move assigns the color k to a vertex x. If x is in the clique, then the invariant is maintained. Note that if Alice played the current move, then she played on the clique. Thus, if x is in the independent set, then Bob colored x with the color k, and this vertex is necessarily adjacent to the $k-1$ vertices colored by colors $1, \ldots, k-1$. Furthermore, x has at least one non-neighbour y_k in the clique. Alice then plays on y_k, which will be colored by k. Thus, the invariant is maintained.

This strategy is depicted on Figure 3.1. □

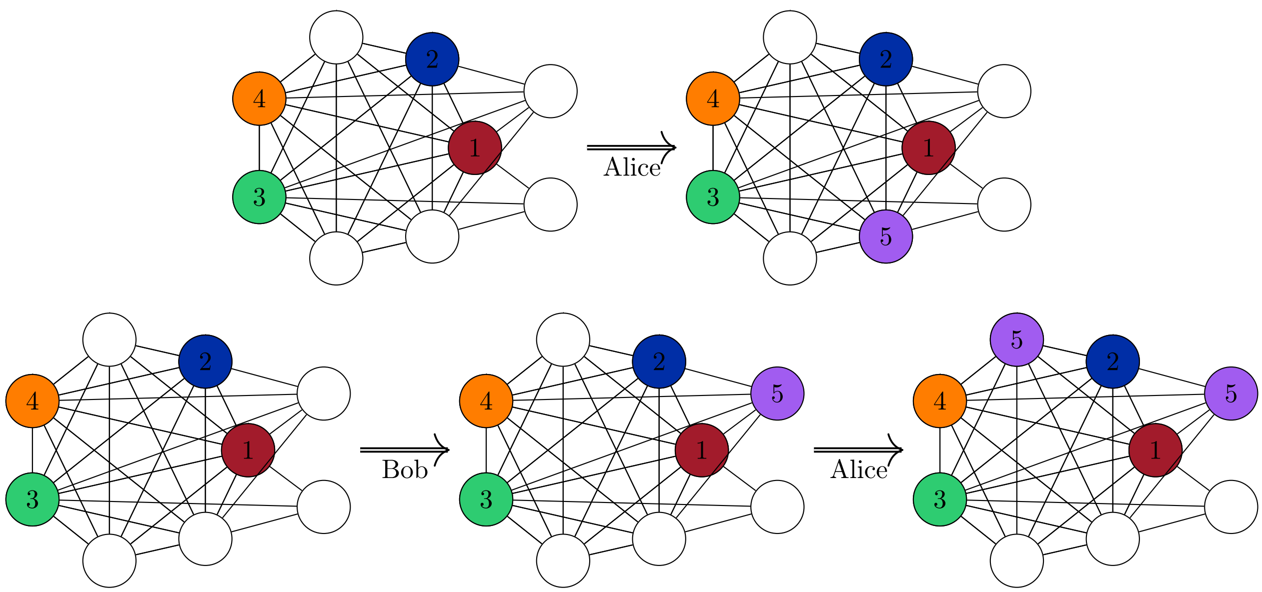


Figure 3.1: How Alice maintains the invariant: above, Alice easily does it if she is the first one to play a new color. Below, if Bob first plays a new color on the independent set, then Alice can find a vertex in the clique that can be colored with the same color.

Another natural question linking games and extremal graph theory is to characterize the graphs for which the parameter reaches a given value. We will characterize the graphs for which the game Grundy number has value 2, $n-1$ and n. First, we need the following lemma. Recall that two edges or non-edges are *disjoint* if they do not share an endpoint.

Lemma 3.21. *Let G be a graph of order n with k disjoint non-edges. Then, $\Gamma_g^A(G), \Gamma_g^B(G) \leq n-k$.*

Proof. We study G once it is fully colored. Let S be the set of vertices of G whose color has been assigned several times during the game. Let C be the set of colors used several times during the game. In particular, exactly $n - |C| + |S|$ colors have been used. For any disjoint non-edge $\overline{uv}$, there are two possibilities:

1. If u and v have the same color k, then $u, v \in S$ and $k \in C$.

2. If u and v have two distinct colors k_u and k_v, assume without loss of generality that $k_u > k_v$. Then, there is a vertex $w \in N(u)$ such that w has color k_v. Thus, $v, w \in S$ and $k_v \in C$.

This implies that $|S| - |C| \geq k$: if a vertex from S is counted for several non-edges then its color is counted only once in C. Thus, at most $n-k$ colors have been used. □

This lemma allows us to study several extremal families for Γ_g^A and Γ_g^B:

Proposition 3.22 (D.)**.** *Let G be a connected graph of order n. Then:*

1. *$\Gamma_g^A(G) = n$ or $\Gamma_g^B(G) = n$ if and only if G is a complete graph;*
2. *$\Gamma_g^A(G) = n-1$ or $\Gamma_g^B(G) = n-1$ if and only if G is a split graph with clique number $n-1$;*
3. *$\Gamma_g^A(G) = 2$ if and only if G is bipartite and there is a vertex in one part that is adjacent to every vertex in the other part;*
4. *$\Gamma_g^B(G) = 2$ if and only if G is bipartite and for every vertex in a part, there is another vertex of the same part such that those two vertices cover the other part.*

Proof.

1. If G is a complete graph then $\Gamma_g^A(G) = \Gamma_g^B(G) = n$. Otherwise, G has at least one non-edge, and Lemma 3.21 implies that $\Gamma_g^A(G), \Gamma_g^B(G) \leq n-1$.
2. G is not a complete graph by the above point. There are three possibilities for the non-edges in G. If there are two disjoint non-edges, then Lemma 3.21 implies that at most $n-2$ colors will be used. If the non-edges of G induce a star, then G is a split graph with clique number $n-1$ and we have $\Gamma_g(G) = n-1$ by Proposition 3.20. If the non-edges of G induce a triangle, then G is a split graph with clique number $n-2$ and we have $\Gamma_g(G) = n-2$ by Proposition 3.20. Together, those three cases prove the statement.
3. If G is bipartite with parts X and Y, and (without loss of generality) there is a vertex $x \in X$ such that $N(x) = Y$, then Alice can play on x which will be colored by the color 1. No vertex in Y can have the color 1, and as such no vertex in X will have a neighbour colored with 1. Thus, all vertices in X are colored with the color 1, and all vertices in Y are colored with the color 2.

   Conversely, assume that no such x exists. Then, for any vertex u that Alice can color with the color 1, Bob can always play on a vertex v which is at distance 3 from u. The path from u to v induces a P_4, and its endpoints are colored with 1, so three colors will be necessary.
4. If G is bipartite with parts X and Y, and for all vertex $x \in X$ (resp. Y) there is a vertex $y \in X$ (resp. Y) such that $N(x) \cup N(y) = Y$ (resp. X), then whatever vertex x Bob plays on, it will be colored by the color 1. Alice then colors the associated vertex y with color 1. The pair $\{x, y\}$ then acts as a vertex from the above point, and the same argument holds. Thus, only 2 colors are necessary.

   Conversely, assume there exists a vertex $x \in X$ such that no vertex $y \in X$ verifies $N(x) \cup N(y) = Y$ (without loss of generality). Then Bob colors x with the color 1. Wherever Alice plays, Bob can then find a vertex y at distance 3 from x that he can color with the color 1. The path from x to y induces a P_4, and its endpoints are colored with 1, so three colors will be necessary.

□

Since the introduction of the Grundy coloring game, several variants have been introduced: an edge coloring variant [106] and a biased variant [76]. The last one allowed the authors to prove that there is no function f such that $\chi_g(G) = f(\Gamma_g(G))$ for every graph G.

While seeming easier than the coloring game, the Grundy coloring game is actually still very difficult to study. The only families that have been studied are very constrained, and the only proof that is not based on the proofs for col_g is the case of the forests. Furthermore, the proof uses the fact that coloring a vertex "cuts" a tree in several trees where the leaves are the only colored vertices, a method that cannot be generalized to other interesting graphs like grids. Furthermore, the complexity of deciding the game Grundy number is still open.

Thus, the Grundy coloring game seems as hard as the coloring game. This is also the case of the other variant that we talked about at the beginning of this section: in the sequential coloring game [16], an order on the vertices is defined, and Alice and Bob color the vertices in this order, only choosing the color. Deciding whether Alice has a winning strategy if there are three colors or more available is PSPACE-complete, even on some very constrained graphs such as bipartite graphs [17].

3.6 Conclusion

As we saw in this chapter, there are many ways to derive games from graph properties or parameters: Maker/Breaker games, construction games, partition games, achievement games, avoidance games, combinatorial games... Those games, and the parameters that they may induce, are an interesting way of studying the resilience of graph properties and parameters.

In the next part, we are going to take this path in the other direction: we will extend the definition of some games in order to play them on graphs. Our goal is to find whether some specific structures that appear in those games (in their classical definition) can be found again in graphs. We will focus on combinatorial games.

Part II

Games

Chapter 4

Combinatorial games

Chapter abstract

This chapter is divided in two parts. In Section 4.1, we remind the reader of the definitions and most important results in the theory of impartial combinatorial games, called the Sprague-Grundy Theory. In Section 4.2, we will recall the literature on a specific family of impartial games played on heaps of counters, the taking-breaking games. We will then explain how we can extend their definitions to play them on graphs.

4.1 The Sprague-Grundy Theory

The definitions and results in this section are all based on the books [5, 15, 31, 95]. As such, we redirected the reader to those books to have more context and examples.

4.1.1 Games and outcome

In the previous chapter, we explained several ways to derive games from graph parameters. In one of them, both players had the same objective: playing the last move (or forcing the other to play it). This is one of the constitutive elements of combinatorial games, which are defined as follows:

Definition 4.1. *A game is* combinatorial *if and only if it meets the following conditions:*

1. *It is a two-player games;*
2. *The players alternate playing, and cannot pass;*
3. *The information is perfect in the sense that both players have a perfect knowledge of the game and the possible options of their opponent and themselves;*
4. *The game is finite: the number of available moves is finite and there can be no loop;*
5. *There is no chance;*
6. *There can be no draw;*
7. *The last move fully determines the winner.*

We will generally use the term of *position* to design the state of the game at a specific moment. Note that some of the conditions can be relaxed, which allows to define related families of games for which the tools of combinatorial games theory can be used with some success. For example, allowing the game to loop through positions gives the loopy games theory (which include Chess for example).

The last condition in Definition 4.1 gives us two winning conventions. In *normal play*, the player who plays the last move wins the game. In *misere play*, the player who plays the last move loses the game. **In the rest of this manuscript, we will only consider games under the normal convention.**

The first combinatorial game that was studied was NIM [22]. This game is played on heaps of counters. At their turn, each player removes as many counters as they want from one heap. The player who removes the last counters wins.

In NIM, the two players are entirely equivalent: both have the exact same sets of moves and the only difference between them is who plays the first move. Such games are called *impartial*. The combinatorial games for which the players have different sets of moves are called *partizan*. **In the rest of this manuscript, we will only consider impartial games.**

As was said in the previous chapter, there are several problems in combinatorial games theory: which player wins, what is the winning strategy, and what is the complexity of deciding those? To study game positions, we define the *outcome*: a position is in $\mathcal{N}$ if the first player has a winning strategy, and in $\mathcal{P}$ if the second player has a winning strategy. It is easy to see that those are the only possible issues for impartial games. Thus, Decision Problem 4.2 is the first question in the study of impartial games.

Decision Problem 4.2.

IMPARTIAL GAME

INPUT: The ruleset of an impartial game, a position J.

QUESTION: Do we have $J \in \mathcal{N}$?

There is a finite algorithm answering to Decision Problem 4.2. We define the *options* of a position J, denoted $\mathrm{opt}(J)$, as the set of positions that a player can reach by playing a move in J. We can notice that $J \in \mathcal{P}$ if and only if $\mathrm{opt}(J) = \emptyset$ or for all $J' \in \mathrm{opt}(J)$, we have $J' \in \mathcal{N}$; and that $J \in \mathcal{N}$ if and only if there exists $J' \in \mathrm{opt}(J)$ such that $J' \in \mathcal{P}$. Thus, Algorithm 1 answers to Decision Problem 4.2 and always terminates. However, its computation time is exponential in the input size. The memory space is generally efficient, since it only depends on the maximal number of rounds. For some games, such as NIM, the memory space is technically exponential in the input size (since an input for NIM is a series of integers coded in binary and thus the maximal number of rounds is exponential in the input size). However, Fraenkel [46] suggests that this complexity can be reduced to a polynomial complexity since the high complexity of the game is due to the small size of the input rather than to the inherent difficulty of the computations. Thus, for games with a small input size (for example a list of integers), we will generally consider that the spatial complexity of Algorithm 1 is polynomial[1].

Algorithm 1: ComputeOutcome(J)

```
Input: A position J of an impartial game
Output: The outcome of J
begin
    S = opt(J) while S ≠ ∅ do
        Let J' ∈ S
        if ComputeOutcome(J') = P then
            return N
        else
            S = S \ {J'}
    return P
```

However, Algorithm 1 still has a high complexity. Thus, several tools have been developed to improve the study of impartial games.

4.1.2 Sum of games and resolution of NIM

For several games, a series of moves can divide the game board in several disjoint subboards. This is illustrated on Figure 4.1 (the position is a position of CRAM, a game where the players alternate playing dominos on a grid, until no more move is available). This allows us to define the disjoint sum of two games:

[1]The idea is that the complexity is polynomial in function of what the input *is* rather than how it is encoded.

Definition 4.3. *Let J_1 and J_2 be two positions of a game. The* disjoint sum *of J_1 and J_2, denoted $J_1 + J_2$, is the game where both players alternate playing either on J_1 or on J_2. The game ends when no move is available anymore in both games.*

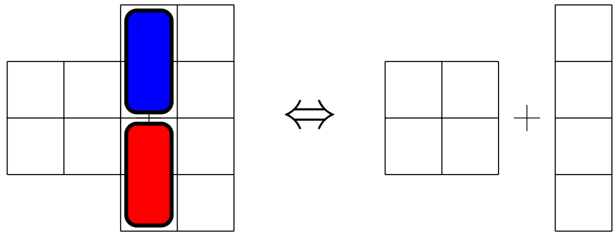


Figure 4.1: The position of CRAM on the left can be considered as the disjoint sum of the two positions on the right.

Note that we can sum two positions of different games, and that the disjoint sum is both associative and commutative. Furthermore, we can easily see that the sum of a position with itself is in $\mathcal{P}$: the second player only has to replicate the move of the first player on the other component, and this way will play the last move. We also obtain several interesting properties, which allow us to study the outcome of a sum of games in function of the outcomes of the two games, as shown on Table 4.1. However, studying the sum of two positions in $\mathcal{N}$ is more difficult. In order to do this, we will need the intuitions that are given by the resolution of NIM.

Proposition 4.4. *Let J be a position of a game. Then, $J + J \in \mathcal{P}$.*

Proposition 4.5. *Let J_1 and J_2 be two positions of a game. If $J_1 \in \mathcal{P}$, then $J_1 + J_2$ has the same outcome than J_2.*

$+$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{N}$	$\mathcal{N}$	?

Table 4.1: Outcome of a sum of games in function of the outcomes of the two games.

In 1901, Bouton fully solved the game of NIM [22]. We will denote a position of NIM with k heaps of $x_1, x_2, \ldots, x_k$ counters by the sequence $(x_1, \ldots, x_k)$. Note that this position is exactly the sum $(x_1)+\ldots+(x_k)$. The resolution of NIM is based on the definition of the *nim-sum*: the nim-sum of the integers $x_1, \ldots, x_k$ is the integer that, when encoded in base 2, is exactly the bitwise sum without carry of the integers $x_1, \ldots, x_k$ written in base 2.

For example, let $x_1 = 7$, $x_2 = 2$ and $x_3 = 11$. We write them in base 2 and align the weak bits together to get:

$$\begin{aligned} x_1 &= & 111 \\ x_2 &= & 10 \\ x_3 &= & 1011 \end{aligned}$$

Their nim-sum is then the bitwise sum without carry:

$$\begin{array}{rr} x_1 = & 111 \\ x_2 = & 10 \\ x_3 = & 1011 \\ \hline x_1 \oplus x_2 \oplus x_3 = & 1110 \end{array}$$

Thus, $x_1 \oplus x_2 \oplus x_3 = 14$. Bouton used this operation to solve NIM (furthermore, his proof gives the winning strategy):

Theorem 4.6 (Bouton (1901) [22]). *Let $J = (x_1, \dots, x_k)$ be a position of* Nim. *Then, $J \in \mathcal{P}$ if and only if $x_1 \oplus \dots \oplus x_k = 0$.*

Note that this gives a polynomial (both in space and in time) algorithm to decide the outcome of a position of Nim. This idea was then expanded independently by Sprague and Grundy to work on all impartial games.

4.1.3 Grundy values and the Sprague-Grundy Theorem

A consequence of Theorem 4.6 is that two positions of Nim with the same nim-sum for their heaps of counters behave exactly the same way when summed to other positions. For example, we can substitute the position $(1, 2, 4)$ by the position (7). This intuition makes us define the equivalence of two games:

Definition 4.7. *Let J_1 and J_2 be two positions of a game. J_1 and J_2 are* equivalent*, denoted $J_1 \equiv J_2$, if and only if for any position J_3 of a game, the outcomes of $J_1 + J_3$ and $J_2 + J_3$ are the same.*

The properties of the sum of the games ensure us that $\equiv$ is an equivalence relation. Thus, we can define equivalence classes of games. The equivalence class of a position J is called its *Grundy value* and is denoted $\mathcal{G}(J)$. The decision problem of computing the outcome of a position is now doubled with a stronger problem, which is the computation of its Grundy value.

We can translate Proposition 4.5 in an equivalence way, and determine whether two positions have the same Grundy value:

Proposition 4.8. *Let J_1 be a position of an impartial game such that $J_1 \in \mathcal{P}$. Then, for any position J_2 of a game, $\mathcal{G}(J_1 + J_2) = \mathcal{G}(J_2)$.*

Proposition 4.9. *Let J_1 and J_2 be two positions of an impartial game. Then, $\mathcal{G}(J_1) = \mathcal{G}(J_2)$ if and only if $J_1 + J_2 \in \mathcal{P}$.*

In particular, two positions of Nim with one heap have different Grundy values since one is an option of the other. This justifies the bijection between Grundy values of Nim positions and nonnegative integers: we state that for any nonnegative integer n, $\mathcal{G}((n)) = n$. This allows us to compute the Grundy value of a position in function of the Grundy value of its options, by using the mex operator: let S be a set of nonnegative integers, the mex of S, denoted $\mathrm{mex}(S)$, is the smallest nonnegative integer not in S. By Theorem 4.6 and the definition of the Grundy value of a Nim position, we can now compute the Grundy value of any impartial game by using the mex of the Grundy value of its options. To do this, we use the following theorem, which states that any position of an impartial game is equivalent to a Nim position:

Theorem 4.10 (Sprague-Grundy Theorem (1935, 1939) [51, 99]). *Let J be a position position of an impartial game, and let $n = \mathrm{mex}(\{\mathcal{G}(J') \mid J' \in \mathrm{opt}(S)\})$. Then, $\mathcal{G}(J) = n$,* i.e. *J is equivalent to the position (n) of* Nim.

This allows us to give a characterization of the positions that are in $\mathcal{P}$: they have a Grundy value of 0. Furthermore, this gives us an algorithm, detailed in Algorithm 2, to compute the Grundy value of a position. Again, this algorithm is generally exponential in time and polynomial in space. Furthermore, we have the following:

Proposition 4.11. *For any position J of an impartial game, $\mathcal{G}(J) \leq |\,\mathrm{opt}(J)|$.*

Algorithm 2: ComputeGrundy(J)

Input: A position J of an impartial game
Output: The Grundy value of J
begin
  $S = \emptyset$ **for** $J' \in \mathrm{opt}(J)$ **do**
    $S = S \cup \{\mathrm{ComputeGrundy}(J')\}$
  return $\mathrm{mex}(S)$

Finally, we can use the Sprague-Grundy Theorem, combined with the characterization of the Grundy values of Nim, to compute the Grundy value of a disjoint sum. This gives us powerful tools to study impartial games. In particular, games that are easy to decompose into sums of smaller positions can be studied by using Proposition 4.12, while we can characterize the Grundy values in many games by using Proposition 4.9. They are extensively used in the taking-breaking games that we are going to review in Section 4.2 and in the games that we will study in the next two chapters.

Proposition 4.12. *Let J_1 and J_2 be two positions of an impartial game. Then :*

$$\mathcal{G}(J_1 + J_2) = \mathcal{G}(J_1) \oplus \mathcal{G}(J_2)$$

4.2 Taking-Breaking games

4.2.1 Taking and breaking

In this section, we will review the taking-breaking games, which are variants of Nim. A game is *taking-breaking* if it is played on heaps of counters and all the moves are of three kinds: taking moves (removing counters from a heap), breaking moves (dividing a heap in several heaps) or both. For instance, Nim is a taking-breaking game in which all moves are breaking moves. There are many other games in this family, and in some of them the rules may differ in function of the sizes of the heaps, or where the players may play in several heaps at the same time. We are only going to consider the taking-breaking games where players can only play in one heap and where the rules do not change depending on the position.

We will reuse the notation introduced for Nim: a position of a taking-breaking game with k heaps of size $n_1, \dots, n_k$ will be denoted $(n_1, \dots, n_k)$. The main problems in the study of taking-breaking games are to solve Decision Problems 4.13 and 4.14, and to study their complexity.

Decision Problem 4.13.

Taking-breaking game
INPUT: A position $J = (n_1, \dots, n_k)$ of a taking-breaking game.
QUESTION: Do we have $J \in \mathcal{N}$?

Decision Problem 4.14.

Grundy value of a taking-breaking game
INPUT: A position $J = (n_1, \dots, n_k)$ of a taking-breaking game, an integer g.
QUESTION: Do we have $\mathcal{G}(J) = g$?

In the games that we study, it is only necessary to study the positions of the form (n), since positions with several heaps of counters can be studied with Proposition 4.12: if we can answer to Decision Problem 4.14 in polynomial time for one heap of counters, then we can answer to it in polynomial time for several heaps. Thus the study of those games is focused on the study of their *Grundy sequence*, which is the sequence of its Grundy values for the positions $(0), (1), (2), (3), \dots$.

Several authors noticed that many Grundy sequences of taking-breaking games exhibit some regularity [5, 15, 42]. In particular, this kind of regularity allows us to answer to Decision Problem 4.14 in polynomial time, so the study of Grundy sequences is focused on the detection of regularity, among which the most classical are periodicity, arithmetic periodicity and ultimate (arithmetic) periodicity. Some specific regularities, such as sapp-regularity, have been defined to study the Grundy sequences of games with Grundy sequences in which most values are regular and the irregular values can be fully characterized.

4.2.2 Subtraction games

Subtraction games are the most simple family of taking-breaking games, in which all moves are taking moves. A *subtraction game* is characterized by a set S, and is denoted SUB(S). In this game, a player can remove k counters from a heap of and only if $k \in S$. If S is finite, then SUB(S) is called a *finite subtraction game*.

For example, Nim is a subtraction game with $S = \mathbb{N}^*$. Subtraction games have been extensively studied, and we have several periodicity results on their Grundy sequences:

Theorem 4.15 ([15])**.** *The Grundy sequence of a finite subtraction game is ultimately periodic.*

Theorem 4.16 ([5])**.** *If S is a finite set, then the game* $\mathrm{SUB}(\mathbb{N} \setminus S)$ *has a ultimately arithmetic periodic Grundy sequence.*

Furthermore, we can compute the period and preperiod of a finite subtraction game:

Theorem 4.17 ([5])**.** *Let N be the greatest element of a set S of positive integers. If there are two integers n_0 and T such that $\mathcal{G}(n) = \mathcal{G}(n+T)$ for all $n_0 \leq n \leq n_0 + N - 1$, then the game* $\mathrm{SUB}(S)$ *is ultimately periodic with period T and preperiod n_0.*

If $t \notin S$ and the Grundy sequences of $\mathrm{SUB}(S)$ and $\mathrm{SUB}(S \cup \{t\})$ are the same, then we say that we can *adjoin* t to S. The following theorem explains when we can adjoin an integer to the set of a finite subtraction game:

Theorem 4.18 ([15, 67])**.** *Let S be a finite set of positive integers, T and n_0 respectively the period and preperiod of* $\mathrm{SUB}(S)$*, and t a positive integer. Then:*

- *If $t < n_0 + T$, then t can be adjoined to S if and only if $\mathcal{G}(n) \neq \mathcal{G}(n+t)$ for any integer $0 \leq n < n_0 + T$;*
- *If $t \geq n_0 + T$, then t can be adjoined to S if and only if $t - T$ can be adjoined to S.*

Several questions remain open on subtraction games. For example, can we improve the periodicity test of Theorem 4.17 and get better periods and preperiods? Is it possible to detect if such a game has a periodic or ultimately periodic Grundy sequence? However, the many structural results allowed the study of more difficult taking-breaking games with the same idea of finding regularities in the Grundy sequences.

4.2.3 Octal games and generalizations

The next step in the study of games is to allow breaking moves. First, we will allow to break a heap into at most two heaps. Figure 4.2 illustrates such a game. For this category of games, an octal code can be defined

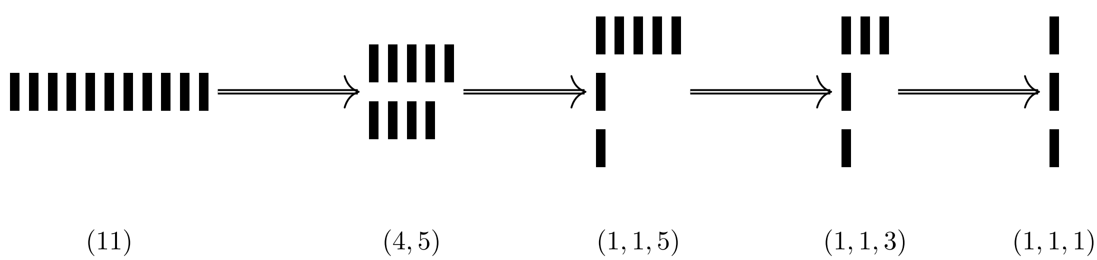


Figure 4.2: A taking-breaking game in which players can remove 2 counters from a heap, and may split it into two nonempty heaps.

to fully determine the ruleset: the ith integer is positive if the taking move removing i counters is allowed, and its value will determine the type of breaking that is allowed. The code is defined as follows, and the several values of numbers in the code are summed up in Table 4.2. This octal code explains why such games are called *octal games*.

Definition 4.19. *Let $d_0, d_1, d_2, d_3, \ldots$ be a sequence of integers in base 8, of the form $d_i = b_i^0 + 2 * b_i^1 + 4 * b_i^2$ and such that $d_0 \in \{0, 4\}$. The octal game $\mathbf{d_0.d_1d_2d_3}\ldots$ is the taking-breaking games in which the players can remove k counters from a heap if and only if $d_k \neq 0$. They may then split the heap into j nonempty heaps if and only if $b_k^j \neq 0$.*

If there is an ℓ such that for all $i > \ell$ we have $d_i = 0$ then such a game is called a *finite octal game* and is denoted $\mathbf{d_0.d_1d_2d_3}\ldots\mathbf{d_\ell}$. Thus, the game presented on Figure 4.2 is the finite octal game $\mathbf{0.07}$. Note that subtraction games are a subfamily of octal games for which $d_i \in \{0, 3\}$ for all i.

Just like subtraction games, there is a periodicity test for octal games:

Value of d_i	Condition for removing i counters from a heap
0	Forbidden
1	After the move, the heap is empty
2	After the move, the remaining counters form a nonempty heap
3	After the move, either the remaining counters form a nonempty heap or the heap is empty
4	After the move, the remaining counters form two nonempty heaps
5	After the move, either the remaining counters form two nonempty heaps or the heap is empty
6	After the move, the remaining counters form either one or two nonempty heap
7	No condition

Table 4.2: The legal moves in an octal game in function of the value of the ith number of the code.

Theorem 4.20 ([15]). *For a finite octal game* $\mathbf{d_0.d_1d_2d_3}\dots\mathbf{d}_\ell$*, if there are two integers* n_0, T *such that* $\mathcal{G}(n) = \mathcal{G}(n+T)$ *for* $n_0 \leq n \leq 2n_0 + 2T + \ell$*, then the Grundy sequence is ultimately periodic with period* T *and preperiod* n_0*.*

However, the study of the Grundy sequences of octal games is computationnally very hard due to the exponential number of options. While a game like **0.07** has been solved by direct computation (it has a Grundy sequence with period 34 and preperiod 68), the game **0.007** is still open and no regularity has been detected even though 2^{28} values have been computed. Many octal games for which the periodicity of the Grundy sequence is still open have been compiled by Flammenkamp [42]. This difficulty explains why there is still no general periodicity theorem for octal games, even the finite ones. However, statistical arguments have led Guy to state the following conjecture:

Conjecture 4.21 (Guy Conjecture [15]). *The Grundy sequence of a finite octal game is ultimately periodic.*

It is possible to generalize the definition of octal games to allow the splitting of a heap into more than two nonempty heaps. If the breaking moves can leave up to three nonempty heaps, then we are studying the *hexadecimal games*, and going further gives the *generalized hexadecimal games*. Those games have even more difficult to study Grundy sequences, which led to the introduction of specific regularities (sapp-regularity or ruler regularity, for example). Furthermore, an arithmetic periodicity test similar to Theorem 4.20 has been stated for hexadecimal games: Austin [7] proved it in the case where the saltus is a power of 2, and Howse et Nowakowski [71] improved it to take in account any saltus. However, this test requires 7 regular periods, unlike the test for octal games which only requires 2 regular periods.

Generalized hexadecimal games have been less studied. In [32], we studied the specific case where only breaking moves are allowed: given a list of integers L, the *pure breaking game with cut-set* L is the game where a heap can be split into k nonempty heaps if and only if $k-1 \in L$. We find a periodicity test for this family of games, which only requires 3 or 4 regular periods:

Theorem 4.22 (D., Duchêne, Larsson, Paris [32]). *We consider a pure breaking game with cut-set* L*. Let* T, n_0 *be two integers and* S *a power of 2. If the Grundy sequence verifies the following conditions:*

1. *For* $n \in \{n_0+1, \dots, n_0+3T\}$, $\mathcal{G}(n+T) = \mathcal{G}(n) + S$ *;*
2. *For* $n \in \{n_0+1, \dots, n_0+T\}$, $\mathcal{G}(n) \in \{0, \dots, S-1\}$ *;*
3. *For* $n \in \{n_0+3T+1, \dots, n_0+4T\}$ *and* $g \in \{0, \dots, S-1\}$, (n) *has an option* O_n *with* m *heaps verifying* $\mathcal{G}(O_n) = g$, $m \geq 2$, $m \in L$.

Then, the Grundy sequence is ultimately arithmetic periodic with period T*, preperiod* n_0 *and saltus* S*.*

Furthermore, the study of several subfamilies of pure breaking games seems to indicate that all those games have ultimately arithmetic periodic sequences, except when the cut-set is $\{1, 2\}$:

Conjecture 4.23 ([32]). *A pure breaking game with cut-set* $L \neq \{1, 2\}$ *has a ultimately arithmetic periodic Grundy sequence.*

Also, recall that many taking-breaking games do not enter the frame of generalized hexadecimal games (and their more specific subfamilies like subtraction or octal games), such as Grundy's Game [51, 15] or Wythoff [105]. Many of these games exhibit an interesting behaviour and are extensively studied. Thus, this domain of research is very active and leads to many interesting problems. It is not surprising that the idea of taking and breaking has been expanded to other structures than heaps of counters, and more specifically to graphs.

4.2.4 Vertex deletion games

Vertex deletion games are games in which the players remove vertices and their incident edges according to specific rules. The problems are the same than in taking-breaking games: computing the outcome, the winning strategy and the Grundy value, as well as studying the complexity of such computations. Note that, for the vertex deletion games where a move can only happen in one component, the study can be limited to connected graphs since Proposition 4.12 gives us the following:

Proposition 4.24. *Let $G = \bigcup_{i=0}^{k} G_i$ be a graph, where the G_i or the connected components of G. Then, $\mathcal{G}(G) = \bigoplus_{i=0}^{k} \mathcal{G}(G_i)$ for any vertex deletion game where a move can only affect one G_i.*

The first known example of a vertex deletion game is Node-Kayles, introduced in 1978 by Schaefer [90]. In this game, the players select one vertex and remove it along with all its neighbours. The game ends when the graph is emptied. Schaefer proved that deciding the outcome of Node-Kayles is PSPACE-complete in general graphs, and later an algorithm running in time $O(1.4423^n)$ (where n is the order of the graph) has been found [19]. Node-Kayles has also been studied on several graph classes such as the graphs with bounded asteroidal number [18] or subdivided stars [44].

A variant where the players delete two adjacent vertices from the graph, called Arc-Kayles, has also been introduced by Schaefer. A game of Arc-Kayles is depicted on Figure 4.3. It is interesting to notice that Arc-Kayles played on a path is exactly **0.07**, which has a ultimately periodic Grundy sequence [52]. Similarly, Arc-Kayles on a grid is Cram, which is easy in some cases (in the case of even-even or even-odd grids) and very difficult in some others: a study of $3 \times n$ grids for $n \leq 20$ showed that no regularity seems to appear [78]. Not much is known on Arc-Kayles: deciding its outcome is FPT [77][2] and the game has been studied on cycles, wheels and subdivided stars with three paths [100]. In particular, for this last case, the regularity of the sequence of values is expressed by fixing one path, letting another one vary from 1 to 34 and studying the Grundy value of the star when the length third path increases. This gives 34 Grundy sequences, for which a periodicity test is given. The authors conjecture that those sequences are ultimately periodic.

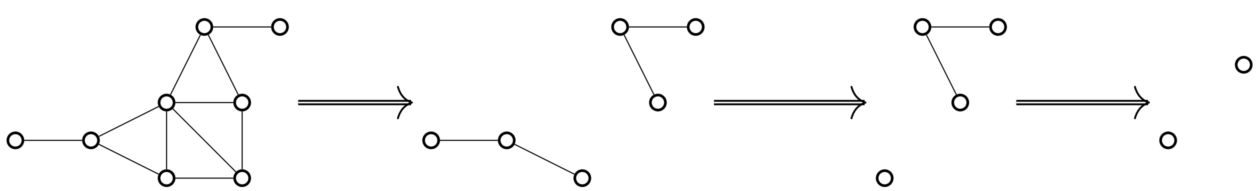


Figure 4.3: A game of Arc-Kayles. The game ends when the game is empty or an independent set.

Those observations and the problem of defining Grundy sequences for vertex deletion games lead us to propose a more general frame for those games, which is an extension of the definition of subtraction and octal games to play them on graphs.

4.2.5 Subtraction and octal games on graphs

In order to extend the definition of octal games on graphs, we need to define taking and breaking moves. First, to keep the link with octal games, where counters can only be removed from one heap, we will allow vertices

[2]There is a computable function f such that we can decide the outcome in time $O(f(k) + poly(n))$ where k is the number of rounds.

to be removed in the same connected component. Furthermore, we impose that the removed vertices induce a connected subgraph: otherwise, subtraction games on graphs would be exactly equivalent to the version on heaps of counters since it is always possible to remove k vertices from a graph without disconnecting it. As for the breaking moves, there are two ways to think about them: either the important point is that the splitting into two heaps matter (which justifies the definition of hexadecimal games), or the important point is to break the connexity of the heap. We choose this second interpretation and allow the players to leave as many connected components as they want. This choice also allows to encompass several vertex deletion games such as ARC-KAYLES.

In [12], we propose the following extension of Definition 4.19 to play them on graphs:

Definition 4.25. *Let $d_0, d_1, d_2, d_3, \ldots$ be a sequence of number in base 8, of the form $d_i = b_i^0 + 2 * b_i^1 + 4 * b_i^2$ and such that $d_0 \in \{0, 4\}$. The octal game on graphs $\mathbf{d_0.d_1d_2d_3}\ldots$ is the vertex deletion game in which the players can remove a connected subgraph of k vertices if and only if $d_k \neq 0$. They may empty the graph if $b_k^0 = 1$, leave it connected if $b_k^1 = 1$ and split it into several connected components if $b_k^2 = 2$.*

This definition allows us to consider many vertex deletion games as octal games. For example, ARC-KAYLES is **0.07** and GRIM [1] is **0.6**. However, our definition does not include NODE-KAYLES since our rule for taking moves forces a fixed number of vertices while in NODE-KAYLES the deletion rule depends on the neighbourhood considered. Note that, by our definition, an octal game played on a path is exactly the octal game played on a heap of counters.

The main problems that we will consider in the topic of octal games on graphs are the computation of the outcome and the Grundy value as well as the complexity of this computation. However, a question arises: if the study of octal games can focus on the Grundy sequence, how can we define a notion of regularity in octal games on graphs? We will reuse some of the work on the subdivided stars for ARC-KAYLES in [100] and study the variation of the Grundy value of a graph when we attach a path to one vertex and make its size vary. Let G be a graph and u one of its vertices, we define $G \underset{u}{\bullet\!\!-\!\!\bullet} P_k$ as the graph obtained by attaching a path of length k to the vertex u, as depicted on Figure 4.4. Thus, we want to study Decision Problem 4.26. In the next chapter, this is the question that we are going to focus on, more specifically in the case of subtraction games.

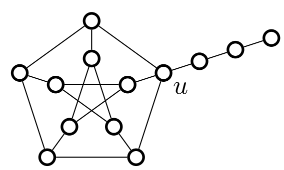


Figure 4.4: Illustration of the notation $G \underset{u}{\bullet\!\!-\!\!\bullet} P_k$, with G the Petersen Graph and $k = 3$.

Decision Problem 4.26.

SEQUENCE OF AN OCTAL GAME ON GRAPHS

INPUT: An octal game $\mathbf{d_0.d_1d_2d_3}\ldots$, a graph $G(V, E)$, a vertex u.

QUESTION: Is the sequence $(\mathcal{G}(G \underset{u}{\bullet\!\!-\!\!\bullet} P_k))_{k \geq 0}$ ultimately (arithmetic) periodic?

Chapter 5

Subtraction games on graphs

The work in this chapter was realized during the GAG[1] ANR project. It was realized in collaboration with **Laurent Beaudou**, **Pierre Coupechoux**, **Sylvain Gravier**, **Julien Moncel**, **Aline Parreau** and **Éric Sopena**. A paper has been published [12] and another one submitted [35]. Furthermore, the work was presented during Graphes@Lyon 2015, the Journées Graphes et Algorithmes 2015 and the Combinatorial Game Theory Colloquium 2.

Chapter abstract

In this chapter, we will study a subfamily of the octal games on graphs as defined in Definition 4.25: the connected subtraction games. Given a set of integers L, the game $\mathrm{CSG}(L)$ is the vertex deletion game where the players can remove a connected subgraph of k vertices if $k \in L$. An example is shown on Figure 5.1. In Section 5.1 we give general periodicity results which have links with the classical subtraction games. In Section 5.2 we study the family $\mathrm{CSG}(I_N)$ (where $I_N = \{1, \dots, N\}$) which has a very simple behaviour in the classical definition, and find ultimate periodicity results. In Section 5.3 we will study the specific case of $\mathrm{CSG}(I_2)$ on subdivided bistars, exhibiting a pseudo-sum of games. Finally, in Section 5.4, we will study the game $\mathrm{CSG}(\{2\})$ which is a variant of Arc-Kayles where disconnecting the graph is forbidden.

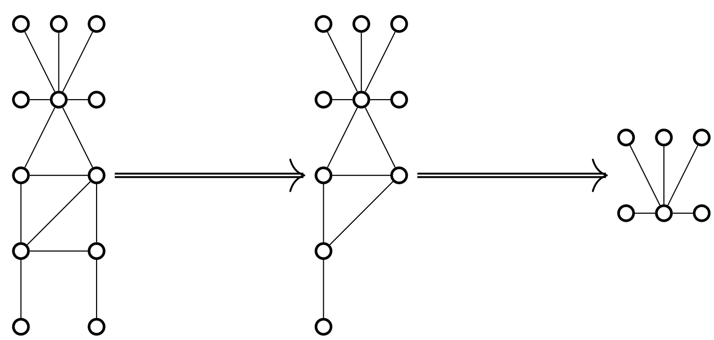


Figure 5.1: A game of $\mathrm{CSG}(\{2, 4\})$. In the right-hand position, it is impossible to remove a connected subgraph of 2 or 4 vertices without disconnecting the graph, thus the game ends.

5.1 General results

Recall that we want to find polynomial algorithms for computing the outcome and the Grundy value of a given graph for a connected subtraction game. In order to do this, we study the graphs of the form $G \underset{u}{\bullet\!\!-\!\!\bullet} P_k$

[1] Graphs and Games : `https://projet.liris.cnrs.fr/gag/`

that were defined at the end of the previous chapter. In particular, given a graph G, a vertex u and a list of integers L, we define the function $f_{L,G,u}$ as follows:

$$\begin{array}{rcrcl} f_{L,G,u} & : & \mathbb{N} & \to & \mathbb{N} \\ & & k & \mapsto & \mathcal{G}(G \underset{u}{\bullet\!\!-\!\!\bullet} P_k) \text{ for } \mathrm{CSG}(L). \end{array}$$

Studying this function allows us to work on Decision Problem 4.26. We are able to give a positive answer to this decision problem for finite connected subtraction games, similarly to Theorem 4.15:

Theorem 5.1. *Let L be a finite set of integers, G a graph and u a vertex of G. The function $f_{L,G,u}$ is ultimately periodic.*

Sketch of proof. The proof is similar to the classical proof of Theorem 4.15: for a large enough k, there are three possible kinds of moves. The first kind is to play to a graph of the form $G \underset{u}{\bullet\!\!-\!\!\bullet} P_{P_{k-i}}$, the second one is to play to $G' \underset{u}{\bullet\!\!-\!\!\bullet} P_{P_k}$ with G' a connected subgraph of G containing u, and the third one is to play to a P_{k-i} by removing G. The number of options being bounded, we can find two integers k_1 and k_2 for which the Grundy values of $G \underset{u}{\bullet\!\!-\!\!\bullet} P_{k_1}$ and $G \underset{u}{\bullet\!\!-\!\!\bullet} P_{k_2}$ are equal. We then prove by induction that the function $f_{L,G,u}$ becomes periodic from k_1 with some large period T. For this, we prove that the mex computation for $f_{L,G,u}(k+T)$ uses options that have the same values than the options of $f_{L,G,u}(k)$ by induction hypothesis. □

However, the period given by this proof is very large, as well as the preperiod. Since all the connected subtraction games on graphs that we studied have the same period than their classical counterpart, we ask the following question:

Open Question 5.2. *Does the function $f_{L,G,u}$ have the same period than the Grundy sequence of the equivalent subtraction game on heaps of counters?*

More generally, finding a stronger periodicity result for connected subtraction games would allow us to construct efficient algorithms for computing the Grundy value of a given graph.

We also prove a result that is inspired by Proposition 4.11:

Proposition 5.3. *For every G and every connected subtraction game, $\mathcal{G}(G) \leq |G|$.*

We then focus specifically on the family $\mathrm{CSG}(I_N)$, that we study on subdivided stars (which are the simplest graph family after the paths and cycles). Note that we call $S_{\ell_1,\ldots,\ell_k}$ the subdivided star with k paths of $\ell_1, \ldots, \ell_k$ vertices.

5.2 The family $\mathrm{CSG}(I_N)$ on subdivided stars

The family $\mathrm{CSG}(I_N)$, where the players can remove up to N connected vertices from a graph, is well-studied in the classical definition (and, as such, on paths). It has a very regular behaviour:

Theorem 5.4 (Folklore). *Let $N \geq 1$ be an integer. For every integer n, the connected subtraction game $\mathrm{CSG}(I_N)$ verifies $\mathcal{G}(P_n) = n \bmod (N+1)$. In other words, the function $f_{I_N,\emptyset,u}$ is periodic with period $N+1$.*

In this section, we will begin by studying the first nontrivial game in this family, $\mathrm{CSG}(I_2)$, on subdivided stars. Then, we will study the family $\mathrm{CSG}(I_N)$ on the same class. We will then prove that adjoining integers is not as easy as it was in the classical subtraction games.

5.2.1 $\mathrm{CSG}(I_2)$ on subdivided stars

In this game, the players can remove one or two adjacent vertices from a graph. We prove the following:

Theorem 5.5. *For all integers $\ell_1, \ldots, \ell_k$, $\mathcal{G}(S_{\ell_1,\ldots,\ell_k}) = \mathcal{G}(S_{\ell_1 \bmod 3,\ldots,\ell_k \bmod 3})$ for the game $\mathrm{CSG}(I_2)$. In other words, the function $f_{I_2,S_{\ell_1,\ldots,\ell_k},u}$ (where u is either the central vertex or a leaf) is periodic with period 3.*

Sketch of proof. It is enough to prove that we can append a P_3 to a leaf or to the central vertex of a subdivided star without changing the Grundy value. For this, we study the sum $S_{\ell_1,\ldots,\ell_k} + S_{\ell_1+3,\ldots,\ell_k}$ and prove by induction that it is in $\mathcal{P}$. For every move of the first player, we find an answer of the second player leading to a position in $\mathcal{P}$. □

This theorem allows to study the Grundy values of subdivided stars since we only need to consider stars with paths of length 1 and 2. From such stars, the only moves consist in removing a path of length 1 or 2 or in changing a path of length 2 into a path of length 1. From the empty graph which has a Grundy value of 0, we build a table of the Grundy values of subdivided stars, which is shown on Figure 5.2. We can notice a regular behaviour after the first few lines: the lines where the star has an odd number of paths are of the form $1203(12)^*$ while those where the star has an even number of paths are of the form $03120(30)^*$. This gives us a polynomial algorithm for computing the Grundy value of any subdivided star for the game CSG(I_2). Now, we will see if such a result exists for the general family CSG(I_N).

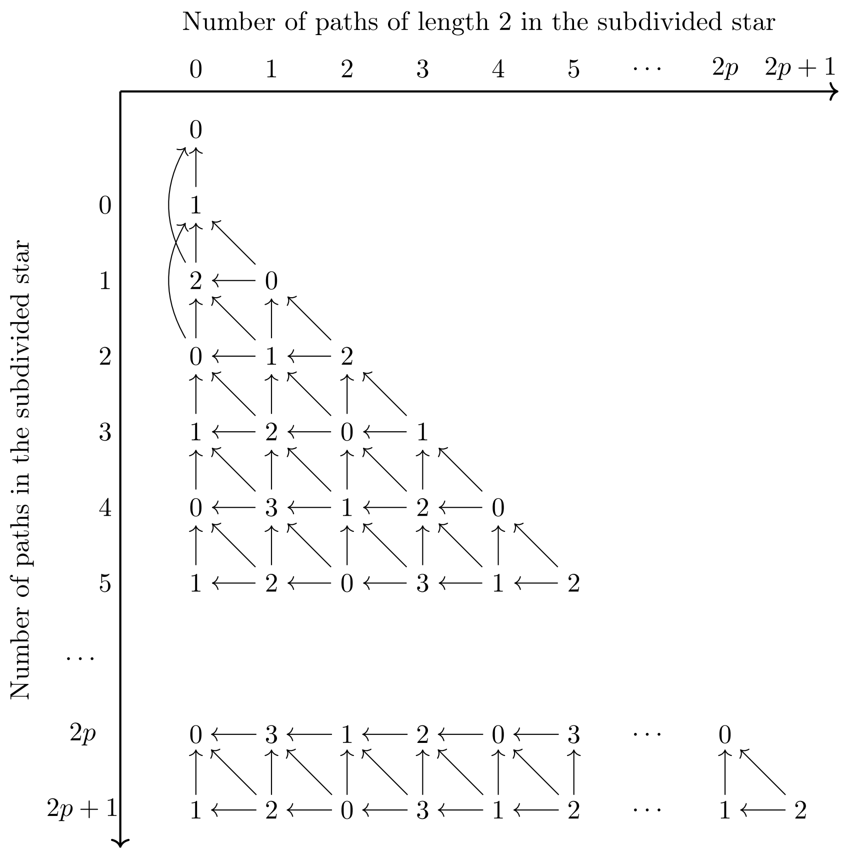


Figure 5.2: Table of the Grundy values of subdivided stars for the game CSG(I_2).

5.2.2 CSG(I_N) on stars

We first find several general results on the family CSG(I_N). The first one allows us to study "small" graphs.

Proposition 5.6. *For every positive integer N and graph G :*

- $\mathcal{G}(G) = 0$ *if* $|G| \in \{0, N+1\}$ *;*
- $\mathcal{G}(G) = 1$ *if* $|G| \in \{1, N+2\}$ *;*
- $\mathcal{G}(G) \geq 2$ *if* $|G| \in \{2, \ldots, N\}$ *;*

for the game CSG(I_N).

Our most used method to prove that $G \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}$ has the same Grundy value than G is to study the sum $G + G \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}$ and to prove that it is in $\mathcal{P}$. The following meta-lemma indicates that, in this case, it is always enough to find an answer of the first player's moves on the component G that remove u:

Lemma 5.7. *Let $N \geq 1$ be an integer, we consider the game* CSG(I_N)*. Let G be a graph, and u a vertex of G. Assume that G is minimal in the sense that $\mathcal{G}(G) \neq \mathcal{G}(G \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1})$, but $\mathcal{G}(H) = \mathcal{G}(H \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1})$ for any subgraph H of G that contains u. Then, all winning moves on $G + G \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}$ are in the component G, delete u and leave at least two vertices.*

We make use of this lemma to prove several results on simple stars, or stars where a path of length $N+1$ is appended to the central vertex. Note that the second result is a reinforcement of Theorem 5.1 since it is a case where the function $f_{L,G,u}$ is purely periodic with the same period than the classical subtraction game.

Theorem 5.8. *Let n be a nonnegative integer, and $N \geq 3$. Then:*

$$\mathcal{G}(K_{1,n}) = \begin{cases} 0 & \text{if } K_{1,n} = \emptyset \\ 1 & \text{if } n = 0 \\ 2 & \text{if } n \leq N-1 \text{ and } n \text{ is odd} \\ 3 & \text{if } n \leq N-1 \text{ and } n \text{ is even} \\ 0 & \text{if } n = N+i \text{ with } i \text{ is even} \\ 1 & \text{if } n = N+i \text{ with } i \text{ is odd} \end{cases}$$

for the game CSG(I_N).

Theorem 5.9. *Let n be a nonnegative integer, and $N \geq 3$. If u is the central vertex of the star $K_{1,n}$, then for every integer k, $\mathcal{G}(K_{1,n} \underset{u}{\bullet\!\!-\!\!\bullet} P_k) = \mathcal{G}(K_{1,n} \underset{u}{\bullet\!\!-\!\!\bullet} P_{k \bmod (N+1)})$ for the game* CSG(I_N).

Next, we study subdivided stars. As was the case in [100], we will first focus on the subdivided stars with three paths and fix the length of one path to 1. We prove that the function $f_{L,G,u}$ is ultimately periodic for those stars, with the same period than the classical subtraction game, but a preperiod for some of the stars.

Theorem 5.10. *If $N \geq 3$, then for every nonnegative integers k, ℓ:*

$$\mathcal{G}(S_{1,k,\ell}) = \begin{cases} \mathcal{G}(S_{1,(k \bmod (N+1))+N+1,N}) & \text{if } k > N \text{ et } \ell \equiv N \bmod (N+1) \\ \mathcal{G}(S_{1,N,(\ell \bmod (N+1))+N+1}) & \text{if } \ell > N \text{ et } k \equiv N \bmod (N+1) \\ \mathcal{G}(S_{1,k \bmod (N+1),\ell \bmod (N+1)}) & \text{otherwise} \end{cases}$$

for the game CSG(I_N).

Furthermore, we give the explicit Grundy values for those stars:

Proposition 5.11. *Assume that $N \geq 3$, we consider the game* CSG(I_N)*. The Grundy value of the subdivided star $S_{1,k,\ell}$ reducted by Theorem 5.10 can be computed the following way:*

- *If $k + \ell + 2 \equiv 0 \bmod (N+1)$, then $\mathcal{G}(S_{1,k,\ell}) = 0$.*
- *If $k + \ell + 2 \equiv 1 \bmod (N+1)$, then $\mathcal{G}(S_{1,k,\ell}) = 1$.*
- *If $k, \ell < N$, then:*

$$\mathcal{G}(S_{1,k,\ell}) = \begin{cases} k + \ell & \text{if } k + \ell \leq N-2 \text{ and } k, \ell \text{ odd} \\ k + 1 & \text{if } k + \ell \leq N-2 \text{ and } k = \ell \neq 0 \text{ even} \\ k + \ell + 2 \bmod (N+1) & \text{otherwise.} \end{cases}$$

- *If $\ell = N$ and $k = a(N+1) + b$, let $r_N = \lfloor \frac{N}{2} \rfloor$ if $N \equiv 2, 3 \bmod 4$ and $\lfloor \frac{N-2}{2} \rfloor$ if $N \equiv 0, 1 \bmod 4$; x_k the number of stars $S_{1,i,N}$ where $i \in \{a(N+1)+1, \ldots, N-1\}$ such that $\mathcal{G}(S_{1,i,N}) > N$; and $m = \text{mex}(\{\mathcal{G}(S_{1,k,i}) \mid i < N\})$. Then :*

$$\mathcal{G}(S_{1,k,\ell}) = \begin{cases} N & \text{if } k = N-1 \text{ or } k = 2N \\ N-1 & \text{if } k = N-2 \text{ or } k = 2N-1 \\ N + x_k + 1 & \text{if } k \in \{1, \ldots, r_N\} \text{ or } m \geq N-1 \\ m & \text{otherwise.} \end{cases}$$

Note that most of the Grundy values of the stars $S_{1,k,\ell}$ are equal to the size of the star modulo $N+1$. However, the values are very difficult to express when one of the two paths is equal to N modulo $N+1$. Theorem 5.10 and Proposition 5.11 give us a polynomial algorithm to compute the Grundy value of a subdivided star of the form $S_{1,k,\ell}$ for the game $\mathrm{CSG}(I_N)$. In particular, this allows to have some base cases for a reduction theorem similar to Theorem 5.5 for specific games in this family like $\mathrm{CSG}(I_3)$:

Theorem 5.12. *For all integers $\ell_1, \ldots, \ell_k$,*

$$\mathcal{G}(S_{\ell_1,\ldots,\ell_k}) = \mathcal{G}(S_{\ell_1 \bmod 4, \ldots, \ell_k \bmod 4})$$

for the game $\mathrm{CSG}(I_3)$.

However, if reduction theorems for subdivided stars exist for small values of N, they cannot exist for all values since the function $f_{I_N, S_{1,k,\ell}, u}$ (with u a leaf of the paths of length k and ℓ) has a preperiod for specific values of k and ℓ. An interesting problem would be to characterize for which values of N reduction theorems are possible in subdivided stars, and get better ultimate periodicity results for the other games.

5.2.3 The adjoining problem on graphs

We saw with Theorem 4.18 that adding some integers to the set characterizing a subtraction game could, in some cases, not change the Grundy sequence. In particular, for every integer N, and every positive integer $M \not\equiv 0 \bmod (N+1)$, the Grundy sequences of the games $\mathrm{SUB}(I_N)$ and $\mathrm{SUB}(I_N \cup \{M\})$ are the same. However, this result cannot be extended to subdivided stars:

Observation 5.13. *For all $N \geq 3$, there is a positive integer $M \not\equiv 0 \bmod (N+1)$ such that there is a graph G and a vertex u of G such that $\mathcal{G}(G) \neq \mathcal{G}(G \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1})$ for the game* $\mathrm{CSG}(I_N \cup \{M\})$.

Proof. Let $S = K_{1,N+2}$ and call u the central vertex of S. We let $M = 2N+4$ and study the game $\mathrm{CSG}(I_N \cup \{M\})$. The position $S + S \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}$ is in $\mathcal{N}$. The first player's strategy is to empty $S \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}$ (which is possible since $|S \underset{u}{\bullet\!\!-\!\!\bullet} P_{N+1}| = M$). The second player can only remove a leaf on S. After two other such forced moves, the first player can end the game by removing the N last vertices, and wins the game. $\square$

Thus, adjoining integers does not behave the same way between the connected subtraction games and the classical ones. However, in some cases, such an adjoining is possible: we studied the game $\mathrm{CSG}(I_2 \cup \{4\})$ (which, in the classical definition, is obtained by adjoining in I_2) and found that the same reduction theorem than Theorem 5.5 holds. Thus, an interesting research problem is to find a theorem similar to Theorem 4.18 and characterize which integers can be adjoined to a subtraction set for some graph families, for example subdivided stars.

5.3 $\mathrm{CSG}(I_2)$ on subdivided bistars

At the beginning of Section 5.2, we found a reduction theorem for subdivided stars for the game $\mathrm{CSG}(I_2)$. In this section, we are going to prove a similar theorem for the family of subdivided bistars, and see how we can create a pseudo-sum of games to compute the Grundy value of a bistar. Let S and S' be two subdivided stars, we denote by $S \overset{m}{\bullet\!\!-\!\!\bullet} S'$ the graph obtained by linking the two central vertices of S and S' by a path of m edges (if $m = 0$ or if $m \geq 1$ and either of the two stars is empty or a path with an endpoint as a central vertex, then the graph is a subdivided star).

We first prove that we can, like with subdivided stars, reduce the paths of a subdivided bistar without changing the Grundy value:

Theorem 5.14. *For all integers $\ell_1, \ldots, \ell_k, \ell'_1, \ldots, \ell'_k, m$:*

$$\mathcal{G}(S_{\ell_1,\ldots,\ell_k} \overset{m}{\bullet\!\!-\!\!\bullet} S_{\ell'_1,\ldots,\ell'_k}) = \mathcal{G}(S_{\ell_1 \bmod 3, \ldots, \ell_k \bmod 3} \overset{m \bmod 3}{\bullet\!\!-\!\!\bullet} S_{\ell'_1 \bmod 3, \ldots, \ell'_k \bmod 3})$$

for the game $\mathrm{CSG}(I_2)$.

Note that the path linking the two central vertices can also be reduced to its length modulo 3, thus a subdivided bistar with a central path of length $3k$ can be reduced to a subdivided star with the same Grundy value.

Now that we have reduced the length of the paths in a subdivided bistar, we want to compute its Grundy value in function of the Grundy value of the two subdivided stars. We can see that playing on a subdivided bistar is very similar to playing independently on the two stars, except at the end when the central vertices can be removed. This leads us to refine the equivalence relation and define a pseudo-sum of games to compute the Grundy value of a subdivided bistar. There are two cases to consider, depending on the length of the middle path.

When there is one edge between the two stars We denote the subdivided bistar $S\overset{1}{\bullet\!-\!\bullet}S'$ by $S\bullet\!-\!\bullet S'$. We refine the equivalence relation $\equiv$ the following way: two subdivided stars S and S' are $\sim_1$-*equivalent*, denoted by $S \sim_1 S'$, if and only if, for every subdivided star $\hat{S}$, $S\bullet\!-\!\bullet\hat{S} \equiv S'\bullet\!-\!\bullet\hat{S}$. This refinement gives us new equivalence classes which are refinements of the Grundy values:

- $\mathcal{C}_1^* = \{P_1, S_{1,2}, S_{2,2,2}\}$ (those stars have Grundy value 1);
- $\mathcal{C}_2^* = \{P_2, S_{2,2}\}$ (those stars have Grundy value 2);
- $\mathcal{C}_2^{\square} = \{S \mid \mathcal{G}(S) = 2$ and S contains one or three paths of length 2$\}$;
- $\mathcal{C}_3^{\square} = \{S \mid \mathcal{G}(S) = 3$ and S contains one or three paths of length 2$\}$;
- For $i \in \{0,1,2,3\}$, $\mathcal{C}_i = \{S \mid \mathcal{G}(S) = i$ and S is not in a previous class$\}$.

Theorem 5.15. *The equivalence classes of $\sim_1$ are exactly the sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_1^*$, $\mathcal{C}_2$, $\mathcal{C}_2^*$, $\mathcal{C}_2^{\square}$, $\mathcal{C}_3$ and $\mathcal{C}_3^{\square}$. Furthermore, Table 5.1 describes how we can compute the Grundy value of $S\bullet\!-\!\bullet S'$ in function of the equivalence classes of S and S'.*

	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_1^*$	$\mathcal{C}_2$	$\mathcal{C}_2^*$	$\mathcal{C}_2^{\square}$	$\mathcal{C}_3$	$\mathcal{C}_3^{\square}$
$\mathcal{C}_0$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_1$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_1^*$	$\oplus$	$\oplus$	2	$\oplus$	0	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_2$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_2^*$	$\oplus$	$\oplus$	0	$\oplus$	1	1	$\oplus$	0
$\mathcal{C}_2^{\square}$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	1	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_3$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$\mathcal{C}_3^{\square}$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	0	$\oplus$	$\oplus$	$\oplus$

Table 5.1: Computing the Grundy value of $S\bullet\!-\!\bullet S'$ in function of the equivalence classes of S and S'. Recall that $\oplus$ is the nim-sum.

Note that, in many cases, we have $\mathcal{G}(S\bullet\!-\!\bullet S') = \mathcal{G}(S) \oplus \mathcal{G}(S')$.

Sketch of proof. We first prove that $\mathcal{C}_1^*$ and $\mathcal{C}_2^*$ are equivalence classes, then we prove that if two subdivided stars S and S' are in the same class that we defined, then $S\bullet\!-\!\bullet\hat{S} + S'\bullet\!-\!\bullet\hat{S} \in \mathcal{P}$ for every subdivided star $\hat{S}$. In order to do this, we study every move of the first player and find an answer for the second player. A thorough case study then proves the equivalence classes. We then obtain Table 5.1 by choosing one representant for each class and computing the Grundy value of the subdivided bistar. $\square$

When there are two edges between the stars Similarly to the previous case, we refine the relation $\equiv$ the following way: two subdivided stars S and S' are $\sim_2$-*equivalent*, denoted by $S \sim_2 S'$, if and only if, for every subdivided star $\hat{S}$, $S\overset{2}{\bullet\!-\!\bullet}\hat{S} \equiv S'\overset{2}{\bullet\!-\!\bullet}\hat{S}$. This gives us new equivalence classes:

- $\mathcal{D}_0^* = \{S \mid \mathcal{G}(S) = 0$ and S contains zero or two paths of length 2$\}$;

- $\mathcal{D}_1^* = \{P_1, S_{1,2}, S_{2,2,2}\}$ (those stars have Grundy value 1);
- $\mathcal{D}_1^{\square} = \{S \mid \mathcal{G}(S) = 1$ and S contains zero or two paths of length 2$\}$;
- $\mathcal{D}_2^* = \{P_2, S_{2,2}\}$ (those stars have Grundy value 2);
- $\mathcal{D}_2^{\square} = \{S \mid \mathcal{G}(S) = 2$ and S contains one or three paths of length 2$\}$;
- $\mathcal{D}_3^{\square} = \{S \mid \mathcal{G}(S) = 3$ and S contains one or three paths of length 2$\}$;
- For $i \in \{0, 1, 2, 3\}$, $\mathcal{D}_i = \{S \mid \mathcal{G}(S) = i$ and S is not in a previous class$\}$.

Theorem 5.16. *The equivalence classes of $\sim_2$ are exactly the sets $\mathcal{D}_0$, $\mathcal{D}_0^*$, $\mathcal{D}_1$, $\mathcal{D}_1^*$, $\mathcal{D}_1^{\square}$, $\mathcal{D}_2$, $\mathcal{D}_2^*$, $\mathcal{D}_2^{\square}$, $\mathcal{D}_3$ and $\mathcal{D}_3^{\square}$. Furthermore, Table 5.2 describes how to compute the Grundy value of $S\overset{2}{\bullet\!\!-\!\!\bullet}S'$ in function of the equivalence classes of S and S'.*

	$\mathcal{D}_0$	$\mathcal{D}_0^*$	$\mathcal{D}_1$	$\mathcal{D}_1^*$	$\mathcal{D}_1^{\square}$	$\mathcal{D}_2$	$\mathcal{D}_2^*$	$\mathcal{D}_2^{\square}$	$\mathcal{D}_3$	$\mathcal{D}_3^{\square}$
$\mathcal{D}_0$	$\oplus$	$\oplus_1$	$\oplus$	2	$\oplus_1$	$\oplus$	0	$\oplus_1$	$\oplus$	$\oplus_1$
$\mathcal{D}_0^*$	$\oplus_1$	$\oplus_1$	$\oplus_1$	2	$\oplus_1$	$\oplus_1$	0	$\oplus_1$	$\oplus_1$	$\oplus_1$
$\mathcal{D}_1$	$\oplus$	$\oplus_1$	$\oplus$	3	$\oplus_1$	$\oplus$	1	$\oplus_1$	$\oplus$	$\oplus_1$
$\mathcal{D}_1^*$	2	2	3	0	3	0	1	1	1	0
$\mathcal{D}_1^{\square}$	$\oplus_1$	$\oplus_1$	$\oplus_1$	3	$\oplus_1$	$\oplus_1$	1	$\oplus_1$	$\oplus_1$	$\oplus_1$
$\mathcal{D}_2$	$\oplus$	$\oplus_1$	$\oplus$	0	$\oplus_1$	$\oplus$	2	$\oplus_1$	$\oplus$	$\oplus_1$
$\mathcal{D}_2^*$	0	0	1	1	1	2	2	2	3	3
$\mathcal{D}_2^{\square}$	$\oplus_1$	$\oplus_1$	$\oplus_1$	1	$\oplus_1$	$\oplus_1$	2	0	$\oplus_1$	1
$\mathcal{D}_3$	$\oplus$	$\oplus_1$	$\oplus$	1	$\oplus_1$	$\oplus$	3	$\oplus_1$	$\oplus$	$\oplus_1$
$\mathcal{D}_3^{\square}$	$\oplus_1$	$\oplus_1$	$\oplus_1$	0	$\oplus_1$	$\oplus_1$	3	1	$\oplus_1$	0

Table 5.2: Computing the Grundy value of $S\overset{2}{\bullet\!\!-\!\!\bullet}S'$ in function of the equivalence classes of S and S'. Recall that $\oplus$ is the nim-sum. Furthermore, $x \oplus_1 y$ means $x \oplus y \oplus 1$.

Sketch of proof. Like with $\sim_1$, we begin by proving the correctness of $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$. Then, we study the Grundy value of $B = S\overset{2}{\bullet\!\!-\!\!\bullet}S'$ in function of the classes of S and S'. We use induction on the order of B, and we use the direct computation of the Grundy values by the mex operator. We have 36 cases to consider (one per combination of classes, minus $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$), and a complete study of them [13] gives us the result. $\square$

Going further? Together, Theorems 5.14 to 5.16 allow us to have a polynomial time algorithm to compute the Grundy value of any subdivided bistar for the game CSG(I_2). This resolution is given by a reduction theorem: paths in a subdivided bistar can be reduced to their length modulo 3 without changing the Grundy value. However, this reduction does not work anymore for trees in general, and even for tristars, as shown on Figure 5.3.

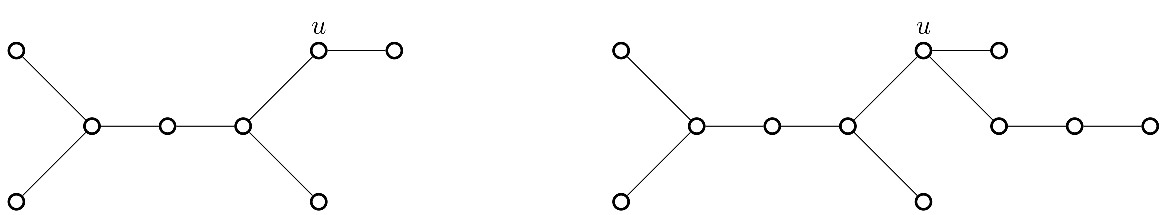


Figure 5.3: Appending a P_3 to u on the left-hand tree gives us the right-hand tree. Those trees have different outcomes. This means that pure reduction theorems such as Theorems 5.5 and 5.14 are not valid on trees.

A computer study of the Grundy values of caterpillars[2] for the game CSG(I_2) allowed us to find graphs with Grundy values going as far 11. This makes us doubt that the Grundy values of trees are even bounded:

Conjecture 5.17. *For every integer $n \geq 4$, there is a tree T such that $\mathcal{G}(T) \geq n$ for the game* CSG(I_2).

An interesting research question is the study of graph classes that have reduction theorems for the game CSG(I_2), and under which conditions.

5.4 CSG($\{2\}$) on graphs

At the end of Section 4.2, we introduced ARC-KAYLES, a vertex deletion game where the players remove two adjacent vertices. This game is equivalent to the octal game **0.07** on graphs. However, as we noted, the few studies that focused on this game suggest that it is particularly hard to study, since a regular behaviour is only conjectured on subdivided stars with three paths among which the length of one is fixed (compare with the family CSG(I_N) where the behaviour is fully known for some of those stars). In this section, we sill study the game CSG($\{2\}$), also called **0.03**, which is a variant of ARC-KAYLES where the players are not allowed to disconnect the graph. Studying this game may give us some insight on ARC-KAYLES since, for a given graph, the options for CSG($\{2\}$) are included in the options for ARC-KAYLES.

5.4.1 First results

The game CSG($\{2\}$) is well-known in the classical definition, and thus on paths:

Theorem 5.18 (Folklore)**.** *In the game* CSG($\{2\}$)*, for any integer $n \geq 0$, $\mathcal{G}(P_n) = 0$ if and only if $n \equiv 0, 1 \bmod 4$. Otherwise, $\mathcal{G}(P_n) = 1$.*

This is due to the fact that all moves are forced. The same result holds for cycles and a similar move exists for complete bipartite graph:

Proposition 5.19. *Let m and n be two positive integers and assume that $m \geq n$. For the game* CSG($\{2\}$)*, $\mathcal{G}(K_{m,n}) = 0$ if and only if n is odd. Otherwise, $\mathcal{G}(K_{m,n}) = 1$.*

Proof. Each move removes a vertex in each part, and the graph cannot be disconnected. Thus, the game will end when one part is reduced to 1 vertex, and will last $n - 1$ turns. □

Another class that we study is the wheels. A wheel W_n is a cycle on n vertices plus one universal vertex. Inspired by [100], we prove the following:

Theorem 5.20. *For the game* CSG($\{2\}$)*, for any integer $n \geq 0$, $\mathcal{G}(W_n) = 1$ if and only if $n \equiv 1, 2 \bmod 4$. Otherwise, $\mathcal{G}(W_n) = 0$.*

Proof. The result is true for $n \in \{0, 1, 2\}$. For every integer $n \geq 3$, we have $\text{opt}(W_n) = \{P_{n-1}, PZ_{n-2}\}$ where PZ_k is the graph formed by a path on k vertices plus one universal vertex. However, we know that $\text{opt}(P_{n-1}) = \{P_{n-3}\} \subseteq \text{opt}(PZ_{n-2})$, thus $\mathcal{G}(PZ_{n-2}) \geq \mathcal{G}(P_{n-1})$. Depending on the Grundy values of the options of W_n, there are several cases to consider:

1. If $\mathcal{G}(P_{n-1}) = 0$ and $\mathcal{G}(PZ_{n-2}) = 0$, then $\mathcal{G}(W_n) = 1$;
2. The case where $\mathcal{G}(P_{n-1}) = 0$ and $\mathcal{G}(PZ_{n-2}) = 1$ cannot happen since $\mathcal{G}(P_{n-1}) = 0 \Leftrightarrow \mathcal{G}(P_{n-3}) = 1$ but $P_{n-3} \in \text{opt}(PZ_{n-2})$;
3. If $\mathcal{G}(P_{n-1}) = 0$ and $\mathcal{G}(PZ_{n-2}) > 1$, then $\mathcal{G}(W_n) = 1$;
4. The case where $\mathcal{G}(P_{n-1}) = 1$ and $\mathcal{G}(PZ_{n-2}) = 0$ cannot happen by our previous discussion;
5. If $\mathcal{G}(P_{n-1}) = 1$ and $\mathcal{G}(PZ_{n-2}) \geq 1$, then $\mathcal{G}(W_n) = 0$.

Thus we have $\mathcal{G}(W_n) = 1$ if and only if $\mathcal{G}(P_{n-1}) = 0$, which is if and only if $n \equiv 1, 2 \bmod 4$; and otherwise $\mathcal{G}(W_n) = 0$. □

[2]A caterpillar is a tree where every vertex is within distance 1 of some path.

5.4.2 Trees

While Arc-Kayles is very hard on trees (the only families of trees on which there are results are paths and subdivided stars with three paths), and even a game such as CSG(I_2) does not have a pure reduction theorem on trees more complex than subdivided bistars, we will see that CSG($\{2\}$) is completely fixed on a given tree and that no strategy is even required to win:

Theorem 5.21. *Let T be a tree. For the game* CSG($\{2\}$)*, there are three possible cases:*

1. $\mathrm{opt}(T) = \emptyset$ *and thus* $\mathcal{G}(T) = 0$*;*
2. $\forall T' \in \mathrm{opt}(T), T' \in \mathcal{N}$ *and thus* $\mathcal{G}(T) = 0$*;*
3. $\forall T' \in \mathrm{opt}(T), T' \in \mathcal{P}$ *and thus* $\mathcal{G}(T) = 1$*.*

Proof. The proof is by induction on the number of vertices in T. The base cases are all the trees with no options (note that they can have as many vertices as we want: take a star for example). For those trees, we have $\mathcal{G}(T) = 0$.

Assume now that $\mathrm{opt}(T) \neq \emptyset$, we prove that the outcome of T changes for any move. The idea of the proof is that when a move is available at a given turn, it will always be available at a later turn (except when $T = P_3$, in which case the result holds by Theorem 5.18) until it is played. The winner will then be decided by the number of turns.

If e is an edge in T, we denote by $T \setminus e$ the graph obtained after removing the two endpoints of e from T. There are two cases:

- Assume there is an edge e such that $T \setminus e \in \mathcal{N}$, then for every edge e' such that $T \setminus e' \in \mathrm{opt}(T)$ we have $T' \setminus e' \in \mathrm{opt}(T')$, except if e and e' share an endpoint which is only possible if $T = P_3$ (which is a case we already considered). However, $T \setminus e' = (T' \setminus e') + e$. Since, by induction hypothesis, $T' \setminus e \in \mathcal{P}$, we have $T \setminus e' \in \mathcal{N}$. This reasoning applies for every edge $e' \neq e$, so we have $T \in \mathcal{P}$ which proves the result.
- Assume that for every edge e we have $T \setminus e \in \mathcal{P}$. By definition, we have $\mathcal{G}(T) = 1$.

Those cases prove the desired result. □

Note that Theorem 5.21 combined with Proposition 4.24 ensures that the Grundy value of a forest is either 0 or 1. A natural question is to try to extend this result to other similar games: can we find a result similar to Theorem 5.21 for games of the family CSG(N)? Otherwise, for which values of N?

5.4.3 Grids

Grids are the board on which Cram is defined (Arc-Kayles restricted to grids is exactly Cram). We study grids of height 2 and 3, and prove that for the first family, like trees, no strategy is possible. However, for the second family, a player has a winning strategy that ensures the emptying of the grid.

Theorem 5.22. *For the game* CSG($\{2\}$)*, let $G_{2,n}$ be a grid of height 2. Then, $\mathcal{G}(G_{2,n}) = 0$ if and only if n is even. Otherwise, $\mathcal{G}(G_{2,n}) = 1$.*

Proof. We prove the result by using an invariant. A $(1,2)$-grid is a connected induced subgraph of a grid of height 2 such that the number of consecutive columns of height 1 is even. This is depicted on Figure 5.4. Note that there is always a legal move on a $(1,2)$-grid in the game CSG($\{2\}$): either removing a column of height 2 on the sides or removing two consecutive columns of height 1 on the sides.

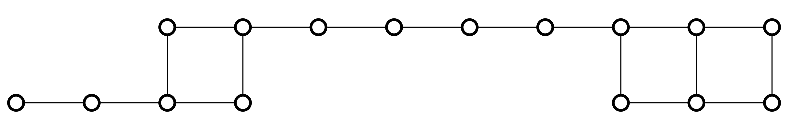


Figure 5.4: A $(1,2)$-grid.

There are two kinds of legal moves on a $(1,2)$-grid:

- Removing a column, which is necessarily on the side, leaves a $(1, 2)$-grid;
- Removing two vertices that are horizontally adjacent. Since disconnecting the graph is forbidden, the move leaves a $(1, 2)$-grid.

Thus, all the possible moves will be played until the graph is emptied, and there are n moves on a $2 \times n$ grid. This proves the result. $\square$

Theorem 5.23. *For the game* CSG($\{2\}$)*, let $G_{3,n}$ be a grid of height 3. Then, $G_{3,n} \in \mathcal{P}$ if and only if $\lfloor \frac{3n}{2} \rfloor$ is even.*

Proof. The proof is based on an invariant that one player (the first if $\lfloor \frac{3n}{2} \rfloor$ is odd, the second otherwise) will maintain. A $(1, 3)$-grid is a connected induced subgraph of a grid of height 3 such that all columns are of height 1 or 3, and in all columns of height 1 the vertex is either in the top row or in the bottom row. Note that a subgraph of a grid of height 3 isomorphic to a path is a $(1, 3)$-grid: we can "flatten" so that it verifies the conditions. Given a $(1, 3)$-grid G with k columns, we define the word $s(G) = s_1 \dots s_k$ where $s_i \in \{1^-, 1^+, 3\}$ is the height of the ith column (if the vertex is in the bottom row, then $s_i = 1^-$; otherwise $s_i = 1^+$). A $(1, 3)$-grid is depicted on Figure 5.5.

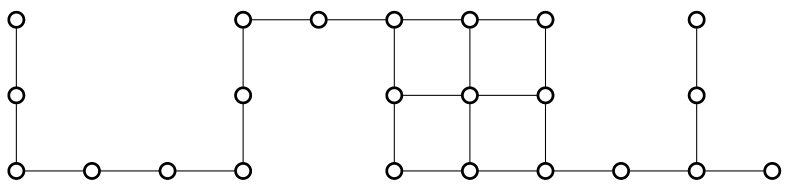


Figure 5.5: A $(1, 3)$-grid G with $s(G) = 31^-1^-31^+3331^-31^-$.

We can always play from a $(1, 3)$-grid (of order greater than 1) to another $(1, 3)$-grid. Indeed, if a player can remove two vertices from a column, the resulting graph will be a $(1, 3)$-grid. Otherwise, for every integer i such that $s_i = 3$, we have $s_{i-1} = 1^-$ (resp. 1^+) and $s_{i+1} = 1^+$ (resp. 1^-), which implies that the graph is a path, and thus a $(1, 3)$-grid.

Furthermore, for every move on a $(1, 3)$-grid (of order greater than 3), then there is an answer leaving a $(1, 3)$-grid. If the move leaves a $(1, 3)$-grid, then by the previous discussion we are done. Otherwise, the move was horizontal. There are three possible kinds of horizontal moves to consider:

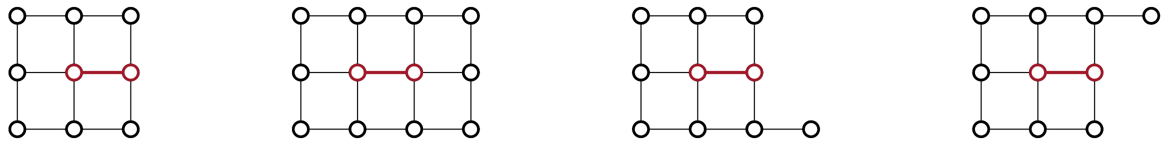
1. Deleting two vertices from the middle row in columns i and $i + 1$. Since the players cannot disconnect the graph, there are only three configurations in which this move is available (modulo the symmetries). The first one is when the column $i + 1$ is the last one and $s_{i-1} = 3$; the second one is when $s_{i-1} = 3$ and $s_{i+2} = 3$; the third one is when $s_{i-1} = 3$ and $s_{i+2} = 1^-$ or $s_{i+2} = 1^+$. In all cases, answering on the top row (if $s_{i+2} = 1^-$ in the third case) or the bottom row (if $s_{i+2} = 1^+$ in the third case) in columns i and $i + 1$ leaves a $(1, 3)$-grid. The configurations are depicted on Figure 5.6.

2. Deleting two vertices from the top row in columns i and $i + 1$. This time, there are six possible configurations (modulo the symmetries), depicted on Figure 5.7. We will not detail the proof, but answering on the middle row in columns i and $i + 1$ leaves a $(1, 3)$-grid.

3. Deleting three vertices from the bottom row in columns i and $i+1$. This case is the same as the previous one.

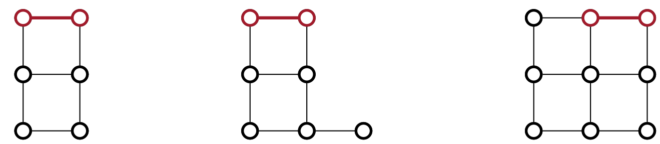
Assume now that $\lfloor \frac{3n}{2} \rfloor$, which is the number of moves that are played if the grid is emptied or reduced to a single vertex, is even. The second player can always leave a $(1, 3)$-grid whatever the first player does until the game ends, and wins. On the contrary, if $\lfloor \frac{3n}{2} \rfloor$ is odd, then the first player plays to a $(1, 3)$-grid and leaves a losing position to the other player. This proves the result. $\square$

Thus, a player can force the grid to be emptied at the end of the game. However, it seems harder to guarantee the existence of such a strategy on grids of greater height. Determining whether this is possible for $k \times n$ grids for $k \geq 4$ is an interesting question.

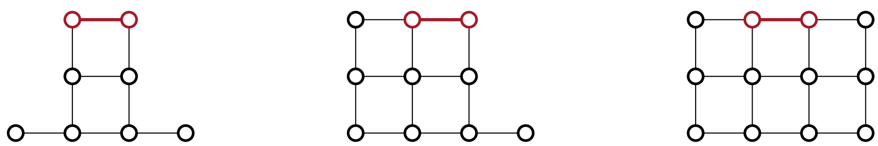


333 ends $s(G)$ | $3333 \in s(G)$ | $3331^- \in s(G)$ | $3331^+ \in s(G)$

Figure 5.6: In those configurations, the structure of the $(1,3)$-grid can be broken by playing on the middle row (the moves are in red and bold). Playing on the top or bottom row restores the structure.



$s(G) = 33$ | 331^- begins $s(G)$ | 333 ends $s(G)$



$1^-331^- \in s(G)$ | $3331^- \in s(G)$ | $3333 \in s(G)$

Figure 5.7: In those configurations, the structure of the $(1,3)$-grid can be broken by playing on the top or bottom row (the moves are in red and bold). Playing on the middle row restores the structure.

5.5 Conclusion and perspectives

The results of this chapter show the interest of generalizing taking-breaking games such as subtraction and octal games on graphs. Indeed, we are able to find (ultimate) periodicity results on graphs that are similar to the equivalent results in the classical definition. However, the study of these games is more difficult on graphs than on heaps of counters: even the family $\mathrm{CSG}(I_N)$, which is well-known in the classical version, is already hard on families such as subdivided stars. Some structure seems to appear, which makes us hope that stronger and more general results can be reached. Furthermore, while we gave several polynomial algorithms for some families of games on specific graph classes, a more general study of the complexity of deciding the Grundy value of a graph for a specific game would be interesting. In particular, the study of connected subtraction or octal games could further our understanding of several vertex deletion games such as Arc-Kayles. In the next chapter, we will use another method to improve our knowledge on Arc-Kayles, and define a weighted variant of this game.

Chapter 6

Weighted Arc-Kayles

The work in this chapter was realized during the GAG ANR project, in collaboration with **Nicolas Bousquet**, **Valentin Gledel** and **Marc Heinrich**. A paper has been submitted for publication [34], and the work was presented during the Combinatorial Game Theory Colloquium 2 and the Seminario Preguntón of the UNAM Juriquilla.

Chapter abstract

In this chapter, we study a weighted variant of Arc-Kayles, called WAK (for Weighted Arc-Kayles): the players select an edge and decrease the weight of the endpoints. When a vertex has weight 0, its incident edges cannot be selected anymore. We begin by giving in Section 6.1 the definitions, some first results and a link with a chesspieces placement game. Then in Section 6.2, we explain how we can reduce graphs, which will give us an exponential reduction between WAK and Arc-Kayles. In Section 6.3, we give a polynomial time algorithm to decide the outcome of WAK on trees of depth at most 2. Finally, in Sections 6.4 and 6.5, we prove that the Grundy values of WAK are unbounded (which gives the same result for Arc-Kayles and Node-Kayles) before giving a periodicity theorem when all weights but one are fixed.

6.1 Definitions

6.1.1 Weighted Arc-Kayles

In this chapter, the graphs that we consider are weighted. A weighter graph $G(V, E, \omega)$ is a graph $G(V, E)$ with a weight function $\omega : V \to \mathbb{N}$ which assigns weights to vertices of the graph. Furthermore, the graphs we will consider may have *loops*, that is to say edges uu with $u \in V$, in which case we say that a loop is *attached to u*.

In Weighted Arc-Kayles, that we will from now on call WAK, the players alternate selecting an edge. The weights of the endpoints of the edge are reduced by 1. An edge cannot be selected if it has an endpoint with weight 0 (this means that vertices with weight 0 can be removed from the graph along with their endpoints without changing the Grundy value). If the players select a loop, then the weight of the endpoint is reduced by 1 (this choice is made since loops are actually created by reductions, as we will see in Section 6.2; the variant where the weight is reduced by 2 could also be studied). Examples of moves are depicted on Figure 6.1 (to simplify the figures, when no confusion is possible vertices will be designed by their weights). WAK has been suggested as an interesting variant of Arc-Kayles in [72]. Note in particular that if all weights are equal to 1 and if there is no loop, then the game is exactly Arc-Kayles.

6.1.2 The rooks game on a holed chessboard

We began studying WAK since it is a model for a chesspiece placement game inspired from the queens game [83]. In the game that we study, the chessboard is a rectangle and there is a subset of the cases of the chessboard that are called *holes*. At their turn, the players play a rook on a case in such a way that no other

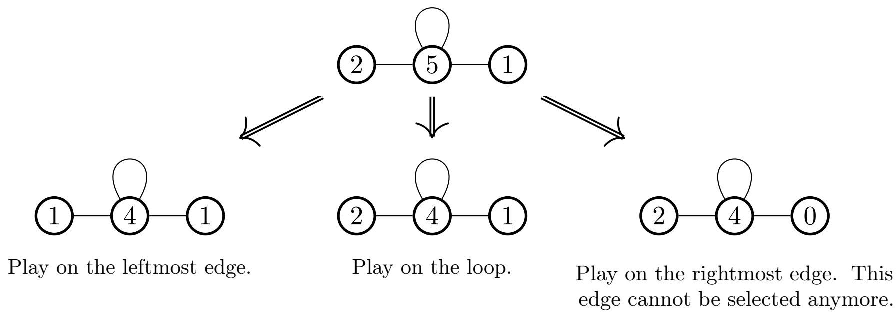


Figure 6.1: Example of moves for WAK.

rook is attacking it. The twist is that rooks cannot attack through holes. Figure 6.2 shows an example of an end position of such a game. We prove that WAK is a model for the game of rooks on a holed chessboard.

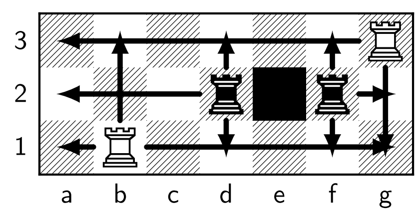


Figure 6.2: An end position of a game of the rooks on a holed chessboard.

Theorem 6.1. *For any position J of the rooks game on a holed chessboard, there is a position $G(V, E, \omega)$ of* WAK *such that $\mathcal{G}(J) = \mathcal{G}(G)$.*

Sketch of proof. We define a *vertical rectangle cover* of a chessboard as a set of rectangles such that no two rectangles intersect each other, all the cases of the chessboard are covered, no rectangle contains a hole, and the only cases that are directly above or below a rectangle are either holes or outside of the chessboard. We can similarly design a *horizontal rectangle cover*.

For each rectangle in those covers, we create a vertex. The vertex has a weight equal to the number of columns (resp. rows) in the rectangle of the vertical (resp. horizontal) cover. An edge exists between two vertices that are created this way if and only if the two rectangles that they represent intersect. The weights Now, we see that playing a rook on the chessboard removes a row from exactly one horizontal rectangle and a column from exactly one vertical rectangle. An equivalent move on WAK is to select the edge between the vertices that represent those rectangles.

This transformation is depicted on Figure 6.3. □

While it is clear that all graphs that we obtain with the construction are bipartite, an open question is whether any bipartite graph can be transformed into a holed chessboard with the inverse construction.

6.1.3 First graphs

We study some basic graphs for WAK. Note that if there is no edge in G, then $\mathcal{G}(G) = 0$. The simplest graph with an edge is the graph with a single vertex u with a loop attached to it. It is easy to see that the game is a parity game, and thus $\mathcal{G}(G) = \omega(u) \bmod 2$. The second case is when there are two vertices u and v linked by one edge. In this case, the game is a parity game which ends when one of the weights reaches 0, and thus $\mathcal{G}(G) = \min(\omega(u), \omega(v)) \bmod 2$.

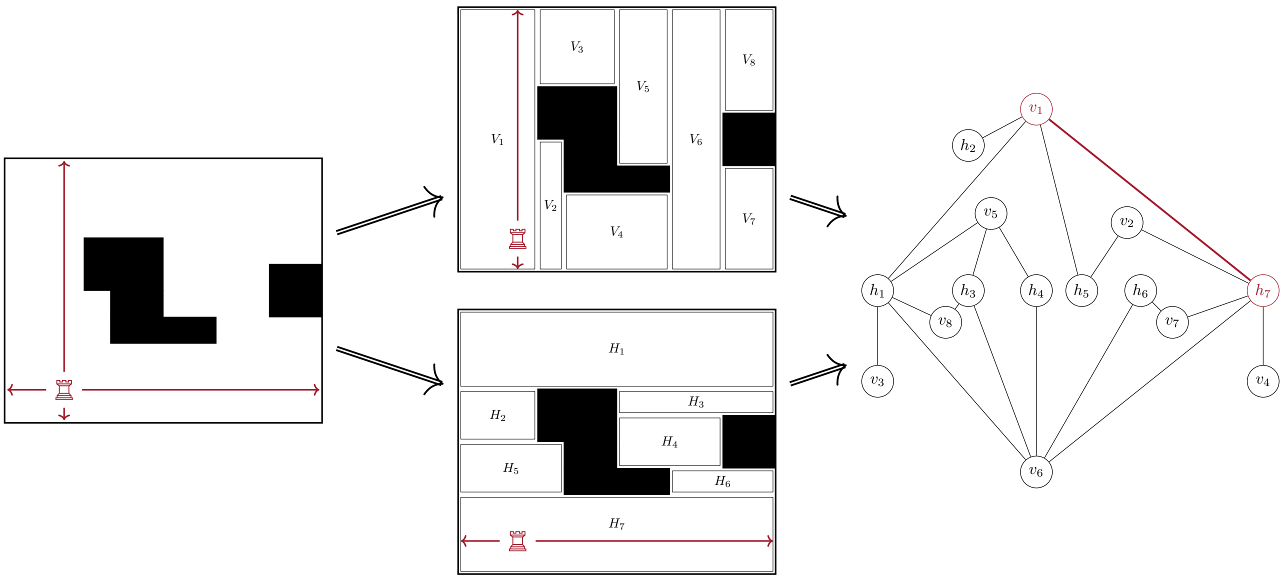


Figure 6.3: How we construct a WAK position from the rooks game on a holed chessboard using rectangle covers. Playing a rooks as shown in red is equivalent to removing a row from H_7 and a column from V_1, and thus to playing on the edge h_7v_1 on the graph.

Now, let us study the graph with two adjacent vertices, one of which has a loop attached to it. On this graph, two moves are available: either the edge or the loop. But playing on the edge is useless, since even if the vertex with no loop is reduced to 0, the loop will still be available to play on. And on the contrary, when the vertex with a loop attached to it is reduced to 0, then the game ends. Thus, the vertex with no loop is "useless" to the game. This observation is the basis on the definition of reduction of those graphs.

6.2 The reduction lemma

We can notice that some vertices have no impact on the game. We define three such kinds of vertices:

1. A vertex u is *useless* if $uu \notin E$ and for all v such that $uv \in E$, we have $vv \in E$.
2. A vertex u is *heavy* if $uu \notin E$ and $\omega(u) \geq \sum_{v \in N(u)} \omega(v)$.
3. Two vertices u and v are *false twins* if $N(u) = N(v)$, $uu \in E \Leftrightarrow vv \in E$, and $uv \notin E$.

We define the reduction operations associated with each of those kinds of vertices. Those are depicted on Figure 6.4.

Definition 6.2. *Let $G(V, E, \omega)$ be a weighted graph. A* reduction *of G is a weighted graph G' obtained by an arbitrary sequence of the following operations, which are called* reduction operations*:*

1. *Deleting a useless vertex u;*
2. *Deleting a heavy vertex u and attaching a loop to each of its neighbours;*
3. *Fuse two false twins u_1 and u_2 into a unique vertex u with $N(u) = N(u_1)$, $uu \in E \Leftrightarrow u_1u_1 \in E$ and $\omega(u) = \omega(u_1) + \omega(u_2)$.*

A sequence of reduction operations will also be called a reduction. We prove that the reduction does not change the Grundy value of the graph:

Lemma 6.3 (Reduction lemma)**.** *If G is a weighted graph and G' is a reduction of G, then $\mathcal{G}(G) = \mathcal{G}(G')$.*

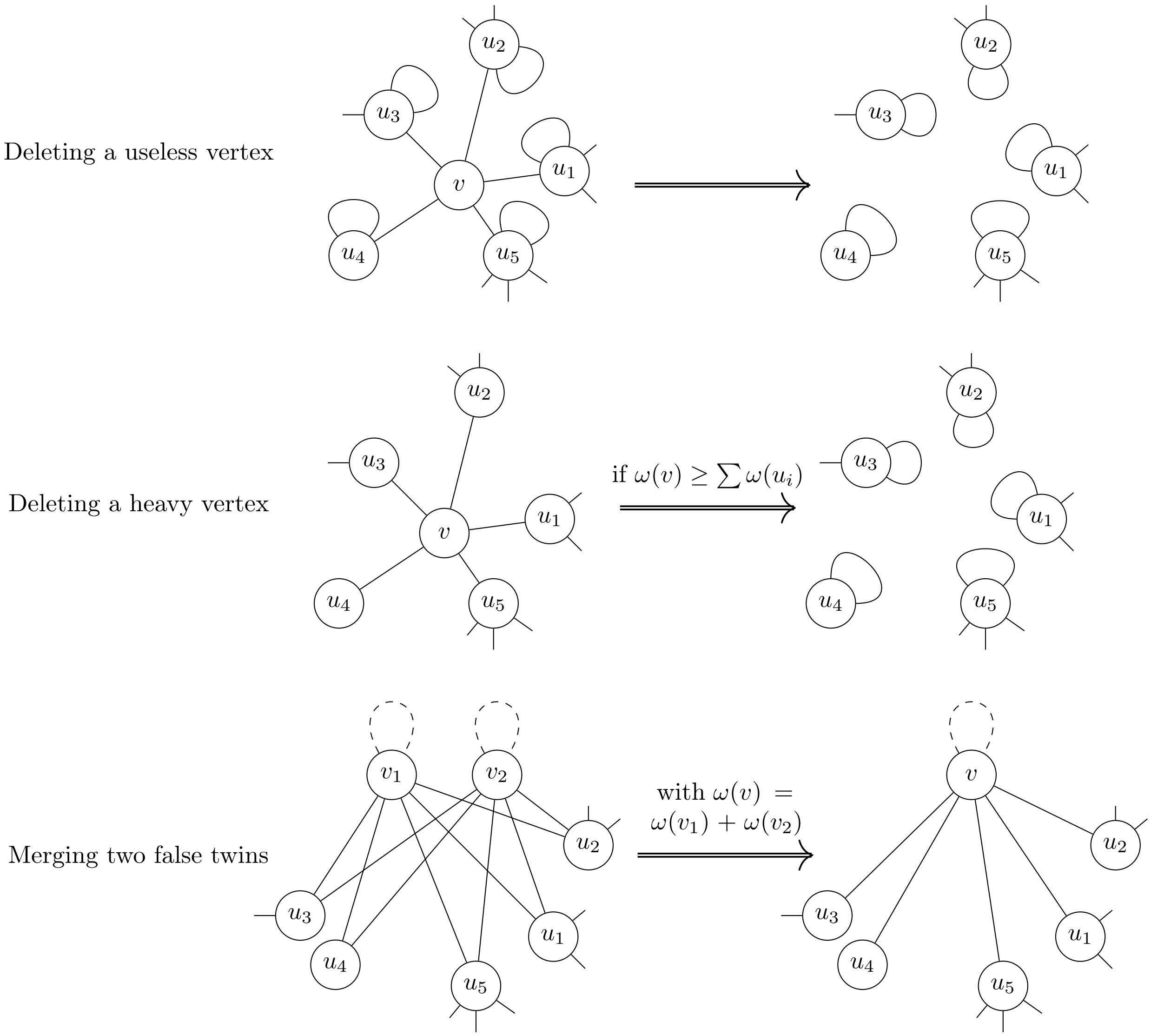


Figure 6.4: The reduction operations (two vertices v_1 and v_2 with a dashed loop represent $v_1v_1 \in E \Leftrightarrow v_2v_2 \in E$).

Sketch of proof. The proof is by induction on the total weights of the vertices. Let G be a weighted graph and G' a graph obtained after applying one reduction operation on G. We prove that for every option of G, there is an equivalent option of G', and conversely. For this, we find equivalent moves in the original graph and the reduced graph.

Let e be an edge. If e is both in G and G', then the playing on it is equivalent on the two graphs. If e is incident with the useless vertex in G that was removed, then the equivalent move in G' is to play on a loop of its neighbours (and conversely). If e is incident with the heavy vertex in G that was removed, then the equivalent move in G' is to play on a loop that was added (and conversely). If e is incident with a false twin in G that was merged, then the equivalent move in G' is to play on the edge that is incident with the merged vertex (and conversely). This covers all the cases, and proves the lemma. $\square$

A graph that cannot be reduced by a reduction operation is called *canonical*. A natural question is to ask whether a given weighted graph has a unique canonical graph. Note that this is not the same thing as asking if the reduction operations are commutative (for example, the merging of false twins can give a heavy vertex that can be removed, but the two operations cannot be done in the other order). We can now restrain our study to canonical graphs, which is a huge simplification. For example, in a star, all leaves are false twins and can be merged, which gives one of the simple graphs that we studied in the previous section.

Furthermore, the reduction operations allow us to find an reduction between Arc-Kayles and WAK:

Theorem 6.4. *Let G be a weighted graph. Then, there is a graph G' such that the Grundy value of G for* WAK *is the same than the Grundy value of G' for* Arc-Kayles.

Sketch of proof. From a weighted graph, we create an unweighted graph the following way:

1. For every vertex u, create $\omega(u)$ vertices;
2. For every non-loop edge uv, create all the edges between the vertices created from u and those created from v;
3. For every loop uu, duplicate each vertex created from u and add an edge between each vertex and their duplicata.

This is depicted on Figure 6.5. Note that building G' is a reversal of the reduction, since all vertices created from loops are heavy and all the other vertices that are created from a vertex are false twins. In particular, applying the reduction from G' (with all weights equal to 1) gives us G, and we have $\mathcal{G}(G) = \mathcal{G}(G')$ by the reduction lemma. This proves the result. □

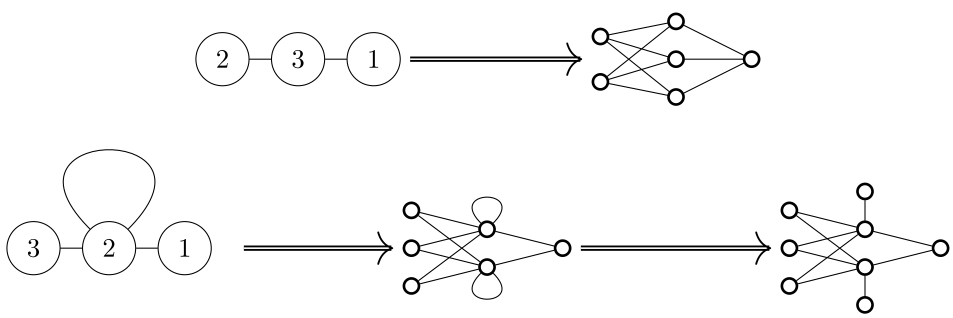


Figure 6.5: How to reduce WAK to Arc-Kayles.

Note that this reduction is not polynomial: a vertex u is transformed into $\omega(u)$ vertices, which is exponential in the size of the binary encoding of $\omega(u)$. A polynomial reduction between WAK and Arc-Kayles would give us a way of studying the complexity of those games.

6.3 Trees of depth 2

In this section, we will study the loopless trees of depth 2. We prove the following theorem:

Theorem 6.5. *There is a polynomial algorithm computing the outcome of a game of* WAK *on a loopless tree of depth 2.*

In order to prove this result, we need to prove several intermediary results. First, we study the graph comprised of two adjacent vertices, each with a loop:

Proposition 6.6. *Let $G(V, E, \omega)$ be the weighted graph where $V = \{u, v\}$, $E = \{uv, uu, vv\}$ and $\omega(u), \omega(v) \geq 0$. Then, $\mathcal{G}(G) = ((\omega(u) + \omega(v))) \bmod 2) + 2 \times (\min(\omega(u), \omega(v)) \bmod 2)$.*

M / m	even	odd
even	0	1
odd	3	2

Table 6.1: The Grundy value of the graph comprised of two adjacent vertices u and v, each with a loop. Note that $m = \min(\omega(u), \omega(v))$ and $M = \max(\omega(u), \omega(v))$

This is summarized in Table 6.1. The proof is by exploring each of the three possible moves.

More generally, the method that we use to prove the outcome of WAK on a givan graph is to partition the set of weight functions (called positions) for this graph into several subsets. We identify one of these subsets as the set of positions in $\mathcal{P}$, and we prove that they are indeed the only positions in $\mathcal{P}$ under some conditions. We use this method to prove that the two leftmost graphs of Figures 6.6 and 6.7, if they are canonical, have the same outcome as the rightmost graphs.

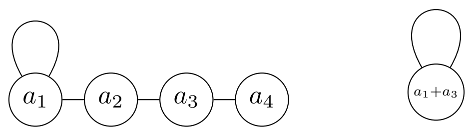


Figure 6.6: Those graphs have the same outcome.

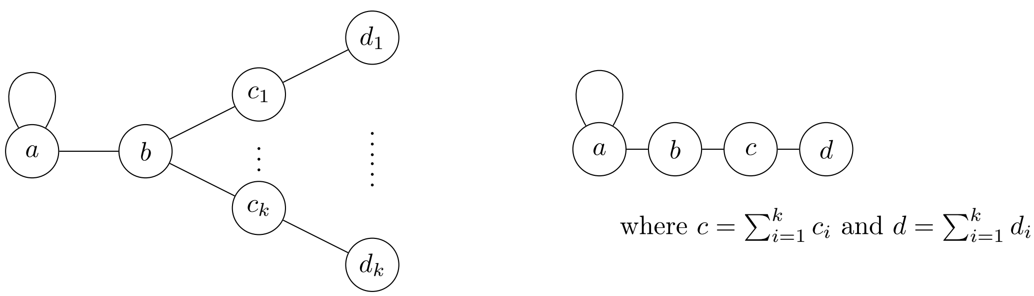


Figure 6.7: Those graphs have the same outcome.

Proof of Theorem 6.5. Let T be a loopless tree of depth 2. Two leaves that are adjacent to the same vertex (the root or an internal vertex) are false twins, so we use the reduction lemma to merge them together. In the resulting tree, each vertex is adjacent to at most one leaf. By the reduction lemma, vertices of depth 2 that are heavy can be removed and a loop added to their neighbour. All the vertices with loops are now false twins, and we can again merge them together with the reduction lemma. The graph that we obtain is the leftmost graph on Figure 6.7. Those steps are depicted on Figure 6.8.

However, the graph that we obtain has the same outcome as the leftmost graph on Figure 6.6. If it is canonical, then the game is a parity game. Otherwise, we can apply the reduction lemma and obtain graphs that we know how to solve (such as the graph from Proposition 6.6 if the third vertex is heavy, for example).

The algorithm that we described in this proof is polynomial, and gives us the outcome of any loopless tree of depth 2. □

Note that the outcome of a loopless tree of depth 2 gives us the outcome of the rooks game on a holed chessboard if the only hole is on the side of the chessboard. We also note that the outcome only depends on parities of sums of weights and on inequalities between weights. However, this is not the case for all graphs: in the case of C_3, the outcomes seem to depend on the modulo 4 of weights. Thus, some interesting questions emerge: do we have more complexy behaviours for outcomes on graphs with higher density? Are all bipartite graphs as "well-behaved" as trees of depth 2?

6.4 The Grundy values are unbounded

The question of whether the Grundy values are bounded are not is open in many vertex deletion games. Some results exist on the vertex and edge deletion game Graph Chomp [84], but the families that are studied tend to have ultimately periodic sequences of Grundy values, such as the connected subtraction games that we studied in Chapter 5. For example, an increasing quantity of irregular values have been found for Node-Kayles on subdivided stars with three paths [44], but the authors do not say whether those irregular values are bounded are not. For Arc-Kayles, the question of finding graphs with arbitrarily large Grundy values has recently been asked [100]. In this section, we prove the following result:

Theorem 6.7. *The Grundy values of* WAK *are unbounded.*

Sketch of proof. We inductively construct a family $G_1, G_2, \ldots, G_n, \ldots$ such that $\mathcal{G}(G_i) \neq \mathcal{G}(G_j)$ for all $i \neq j$, and such that for every graph G_i there is a vertex u_i with a loop such that playing on the loop is a winning move. G_1 is simply a single vertex with a loop and weight 1. We then construct G_{n+1} the following way, depicted on Figure 6.9:

1. For every integer $i \in \{1, \ldots, n\}$, construct two copies G'_i and G''_i of G_i ;
2. Construct the vertex u_{n+1} with weight 1 and a loop attached to it, and link it to every u'_i of every G'_i.

It is then easy to see that G_{n+1} has a different Grundy value than every G_i: for all $i \in \{1, \ldots, n\}$: playing on the edge $u_{n+1}u'_i$ leaves a disjoint sum of graphs which has Grundy value $\mathcal{G}(G_i)$. By the computation of Grundy value by mex of the options, we have $\mathcal{G}(G_{n+1}) \neq \mathcal{G}(G_i)$. Furthermore, playing on the loop attached to u_{n+1} leaves a disjoint sum of graphs that are two by two the same, and thus has Grundy value 0, which proves that playing on $u_{n+1}u_{n+1}$ is a winning move. □

Since WAK can be reduced to Arc-Kayles by Theorem 6.4, and since playing a game of Arc-Kayles on G is equivalent to playing a game of Node-Kayles on the line graph $L(G)$, we have the following result:

Corollary 6.8. *The Grundy values of* Node-Kayles *and* Arc-Kayles *are unbounded.*

This answers to several questions of the literature. However, the construction in the proof of Theorem 6.7 gives us exponential order graphs (G_n has order 3^{n-1}). An interesting problem is to search polynomial families with unbounded Grundy values for WAK and Arc-Kayles.

6.5 Conclusion and perspectives

In this chapter, we studied a weighted variant of Arc-Kayles that allowed us to gain some insight on Arc-Kayles. In particular, we proved that the Grundy values are unbounded, which is an open question in many vertex deletion games. We even prove periodicity results, similar to the results in Chapter 5, when all weights but one are fixed:

Theorem 6.9. *Let G be a graph with vertices $u_1, u_2, \ldots, u_n$ such that $u_1u_1 \in E$. We fix the integers $\omega_i \geq 0$ for $i \in \{2, \ldots, n\}$, let $\{S_x\}_{x\geq 0}$ the sequence defined as follows: for all x, S_x is the outcome of the game of* WAK *played on G with the weight function ω such that $\omega(u_1) = x$ and $\omega(u_i) = \omega_i$ for $i \in \{2, \ldots, n\}$. Then, the sequence $\{S_x\}_{x\geq 0}$ is ultimately periodic with period 2 and preperiod at most $2\sum_{i\geq 2}\omega_i$.*

Theorem 6.10. *Let G be a weighted graph. The sequence $\{\mathcal{G}((x, \omega_2, \ldots, \omega_n))\}_{x\geq 0}$ of the Grundy values of G with weight functions $(x, \omega_2, \ldots, \omega_n)$ is ultimately periodic with period 2. Furthermore, if the Grundy values of the periodic part of the sequence are bounded by an integer k, then there is a constant c_k depending only on k such that the preperiod is at most $2\sum_{i\geq 2}\omega_i + c_k$.*

Many questions remain open on WAK, and solving them could give us some insight for Arc-Kayles. This is for example the case of complexity, where the research of a polynomial reduction between Arc-Kayles and WAK is still open. Other interesting questions are the study of more complex graph classes, in particular those linked with the rooks game on a holed chessboard, such as the one depicted on Figure 6.10.

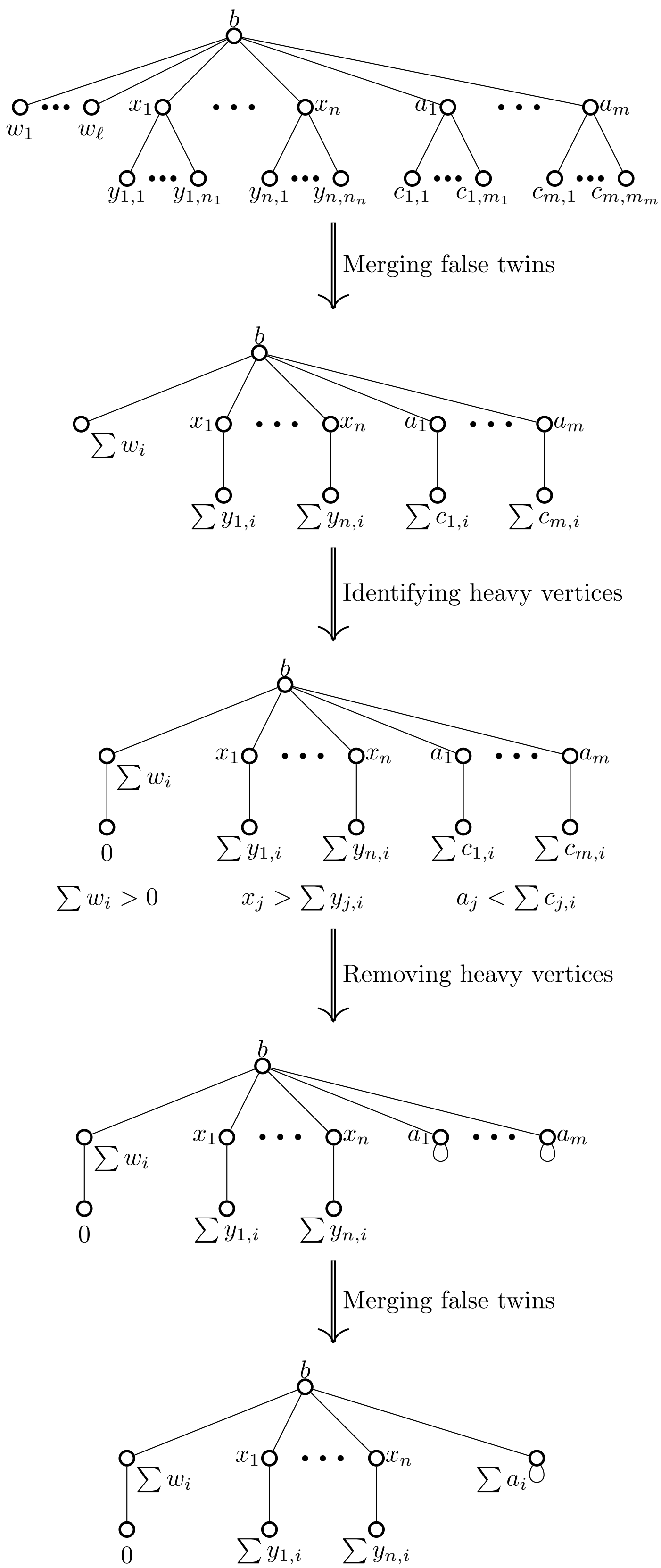


Figure 6.8: Reducing a loopless tree of depth 2 to the leftmost graph on Figure 6.7.

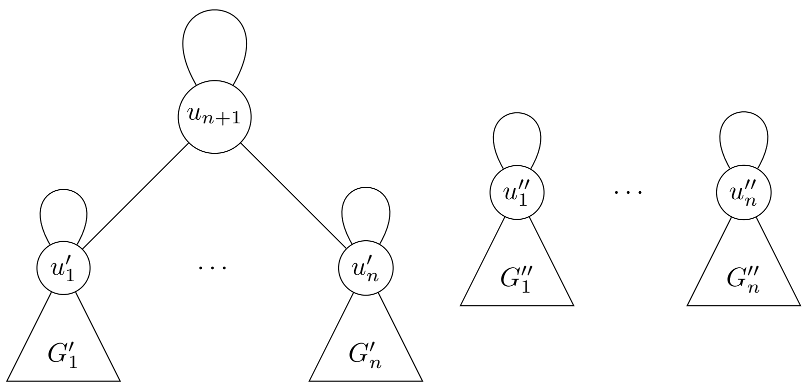


Figure 6.9: The inductive construction of G_{n+1}. Each vertex has weight 1.

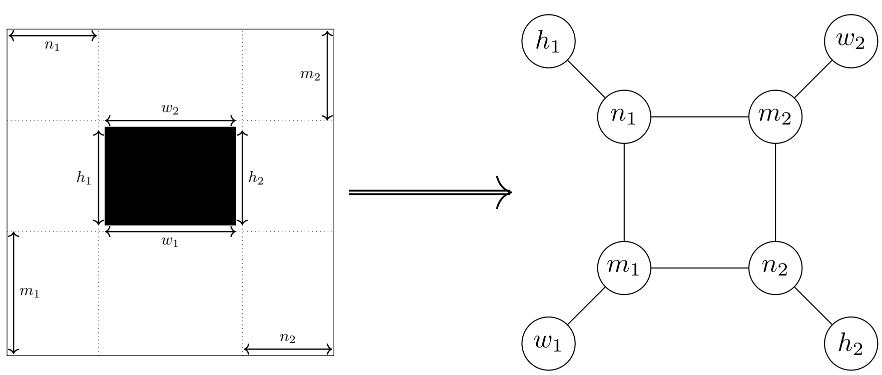


Figure 6.10: A chessboard with a hole in the middle, and the equivalent WAK graph.

Conclusion

Conclusion

During this thesis, we studied several graph problems (Part I) and combinatorial games played on graphs (Part II), as well as showing many links between the two fields (Chapter 3). In Chapter 1, we studied the Murty-Simon Conjecture, which we strengthened. We also proved an improved bound for a specific subfamily. In Chapter 2, we studied a vertex-distinguishing edge coloring with a constant ratio between the upper and lower bounds, while for most other such colorings the ratio is of order $\frac{n}{\log_2 n}$. In Chapter 3 we showed how to construct games from graph properties and parameters and talked about the new problems that arise with those constructions, notably in terms of complexity. In Chapter 4, we recalled the basis of the Sprague-Grundy Theory and did an overview of the literature on taking-breaking games, that we extended to play them on graphs. In Chapter 5 we studied subtraction games on graphs and proved several regularity results, both generally and on specific games and graphs families. Finally, in Chapter 6, we studied a weighted variant of classical vertex deletion game Arc-Kayles and showed in particular that the Grundy values of our variant and Arc-Kayles are unbounded.

Some of the problems that we saw in this manuscript are unsolved, or bring up new questions which are as many new research problems. We will summarize the most interesting ones:

- Proving the strengthened conjecture on D2C graphs (Conjecture 1.5). Measuring the difference between a complete bipartite graph and an other D2C graph is a possible idea for solving this.
- Proving that there is no graph G with $\chi_{\cup}(G) = \lceil \log_2(|V(G)| + 1) \rceil + 2$ (Conjecture 2.15), for example by studing the value of the parameter $\chi_{\cup}$ on trees or forests of stars subdivided at most once.
- Determining the complexity of deciding the value of $\chi_{\cup}$ for a given graph (Decision Problem 2.17).
- Proving that all pure breaking games (but the one with cut-set $\{1, 2\}$) have a ultimately arithmetic periodic Grundy sequence (Conjecture 4.23).
- Verifying if connected subtraction games have the same period than the classical subtraction games (Open Question 5.2) for a given definition of the Grundy sequence (appending a path to a specific vertex).
- Studying the complexity of connected subtraction games: general complexity, complexity of specific families (for example CSG(I_N)), complexity on specific graph classes.
- Determining for which values of N a reduction theorem exist in subdivided stars for the games in the family CSG(I_N). For other values of N, study the smallest preperiod that we can reach.
- Determining whether the Grundy values of CSG(I_N) are unbounded is also interesting. For example, the game CSG(I_2) seems to have high Grundy values on trees (Conjecture 5.17).
- Determining for which graph classes we can find reduction theorems for CSG(I_2).
- Characterizing under which conditions we can adjoin an integer to a subtraction set.
- Study the family CSG($\{N\}$) on trees.

- Find a polynomial reduction from WAK to Arc-Kayles, and study the complexity of WAK, in order to gain a better understanding of the complexity of vertex deletion games.
- Finding polynomial order graph families with unbounded Grundy value for WAK or Arc-Kayles.
- Studying graphs for WAK that allow to solve positions of the rooks game on a holed chessboard.

In conclusion, this thesis, which is at the intersection between graphs and games, covers topics from both fields. One of my goals was to show that adapting classical games to play them on graphs could lead to interesting results, and the regularity results and reduction theorems for connected subtraction games are a proof that this interest is not in vain. My goal in the future is to maintain this balance between graphs and games. I would like to keep on working on problems that we talked about in this manuscript, more specifically on the strengthening of the Murty-Simon Conjecture and on connected subtraction games.

Bibliography

[1] Richard Adams, Janae Dixon, Jennifer Elder, Jamie Peabody, Oscar Vega, and Karen Willis. Combinatorial analysis of a subtraction game on graphs. *International Journal of Combinatorics*, 2016, 2016.

[2] Louigi Addario-Berry, Robert EL Aldred, Ketan Dalal, and Bruce A Reed. Vertex colouring edge partitions. *Journal of Combinatorial Theory, Series B*, 94(2):237–244, 2005.

[3] Méziane Aïder, Sylvain Gravier, and Souad Slimani. Relaxed locally identifying coloring of graphs. *Graphs and Combinatorics*, 32(5):1651–1665, 2016.

[4] Martin Aigner, Eberhard Triesch, and Zsolt Tuza. Irregular assignments and vertex-distinguishing edge-colorings of graphs. In *Annals of Discrete Mathematics*, volume 52, pages 1–9. Elsevier, 1992.

[5] Michael Albert, Richard Nowakowski, and David Wolfe. *Lessons in play: an introduction to combinatorial game theory*. CRC Press, 2007.

[6] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[7] Richard Bruce Austin. Impartial and partisan games. 1976.

[8] Camino Balbuena, Adriana Hansberg, Teresa W Haynes, and Michael A Henning. Total domination edge critical graphs with total domination number three and many dominating pairs. *Graphs and Combinatorics*, 31(5):1163–1176, 2015.

[9] Paul N Balister, Ervin Gyori, Jenö Lehel, and Richard H Schelp. Adjacent vertex distinguishing edge-colorings. *SIAM Journal on Discrete Mathematics*, 21(1):237–250, 2007.

[10] Paul N Balister, OM Riordan, and Richard H Schelp. Vertex-distinguishing edge colorings of graphs. *Journal of graph theory*, 42(2):95–109, 2003.

[11] Cristina Bazgan, Amel Harkat-Benhamdine, Hao Li, and Mariusz Woźniak. On the vertex-distinguishing proper edge-colorings of graphs. *Journal of Combinatorial Theory, Series B*, 75(2):288–301, 1999.

[12] Laurent Beaudou, Pierre Coupechoux, Antoine Dailly, Sylvain Gravier, Julien Moncel, Aline Parreau, and Eric Sopena. Octal games on graphs: The game 0.33 on subdivided stars and bistars. *Theoretical Computer Science*, 2018.

[13] Laurent Beaudou, Pierre Coupechoux, Antoine Dailly, Sylvain Gravier, Julien Moncel, Aline Parreau, and Eric Sopena. Octal games on graphs: The game 0.33 on subdivided stars and bistars. full proof of theorem 22. working paper or preprint, June 2018.

[14] Claude Berge. Sur les jeux positionnels. *Cahiers du Centre Etudes Rech. Opér*, 18:91–107, 1976.

[15] Elwyn R Berlekamp, John H Conway, and Richard K Guy. *Winning Ways for Your Mathematical Plays*. AK Peters/CRC Press, 2001-2004.

[16] Hans L Bodlaender. On the complexity of some coloring games. *International Journal of Foundations of Computer Science*, 2(02):133–147, 1991.

[17] Hans L Bodlaender and Dieter Kratsch. The complexity of coloring games on perfect graphs. *Theoretical Computer Science*, 106(2):309–326, 1992.

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

[19] Hans L Bodlaender, Dieter Kratsch, and Sjoerd T Timmer. Exact algorithms for kayles. *Theoretical Computer Science*, 562:165–176, 2015.

[20] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Citeseer, 1976.

[21] Nicolas Bousquet, Antoine Dailly, Eric Duchene, Hamamache Kheddouci, and Aline Parreau. A vizing-like theorem for union vertex-distinguishing edge coloring. *Discrete Applied Mathematics*, 232:88–98, 2017.

[22] Charles L Bouton. Nim, a game with a complete mathematical theory. *The Annals of Mathematics*, 3(1/4):35–39, 1901.

[23] Boštjan Brešar, Sandi Klavžar, and Douglas F Rall. Domination game and an imagination strategy. *SIAM Journal on Discrete Mathematics*, 24(3):979–991, 2010.

[24] Anita C Burris and Richard H Schelp. Vertex-distinguishing proper edge-colorings. *Journal of graph theory*, 26(2):73–82, 1997.

[25] Louis Caccetta and Roland Häggkvist. On diameter critical graphs. *Discrete Mathematics*, 28(3):223–229, 1979.

[26] Ján Černỳ, Mirko Horňák, and Roman Soták. Observability of a graph. *Mathematica Slovaca*, 46(1):21–31, 1996.

[27] Gary Chartrand, Michael S Jacobson, Jeno Lehel, Ortrud R Oellermann, Sergio Ruiz, and Farrokh Saba. Irregular networks. *Congr. Numer*, 64(197-210):250th, 1988.

[28] Xiang'en Chen. Point-distinguishing chromatic index of the union of paths. *Czechoslovak Mathematical Journal*, 64(3):629–640, 2014.

[29] Claude A Christen and Stanley M Selkow. Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B*, 27(1):49–59, 1979.

[30] Vašek Chvátal and P Erdös. Biased positional games. In *Annals of Discrete Mathematics*, volume 2, pages 221–229. Elsevier, 1978.

[31] John H Conway. *On numbers and games.* AK Peters/CRC Press, 2000.

[32] Antoine Dailly, Eric Duchene, Urban Larsson, and Gabrielle Paris. Partition games are pure breaking games. *arXiv preprint arXiv:1803.02621*, 2018.

[33] Antoine Dailly, Florent Foucaud, and Adriana Hansberg. A strengthening of the murty-simon conjecture for diameter 2 critical graphs. *En cours de rédaction*, 2018.

[34] Antoine Dailly, Valentin Gledel, and Marc Heinrich. A generalization of arc-kayles. *International Journal of Game Theory*, 2018.

[35] Antoine Dailly, Julien Moncel, and Aline Parreau. Connected subtraction games on subdivided stars. *En cours de rédaction*, 2018.

[36] Eric Duchene. *HABILITATION A DIRIGER DES RECHERCHES Graphes et jeux combinatoires.* PhD thesis, Universite Claude Bernard Lyon 1, 2015.

[37] Éric Duchêne, Valentin Gledel, Aline Parreau, Gabriel Renault, and Simon Schmidt. Maker-breaker domination game, 2018.

[38] Paul Erdös and John L Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3):298–301, 1973.

[39] Louis Esperet, Sylvain Gravier, MickaÃ Montassier, Pascal Ochem, Aline Parreau, et al. Locally identifying coloring of graphs. *the electronic journal of combinatorics*, 19(2):P40, 2012.

[40] Genghua Fan. On diameter 2-critical graphs. *Discrete mathematics*, 67(3):235–240, 1987.

[41] Stephen A Fenner and John Rogers. Combinatorial game complexity: an introduction with poset games. *arXiv preprint arXiv:1505.07416*, 2015.

[42] Achim Flammenkamp. Sprague-grundy values of octal games. `http://wwwhomes.uni-bielefeld.de/achim/octal.html`, accédé le 30 avril 2018.

[43] Evelyne Flandrin, Antoni Marczyk, Jakub Przybyło, Jean-François Saclé, and Mariusz Woźniak. Neighbor sum distinguishing index. *Graphs and Combinatorics*, 29(5):1329–1336, 2013.

[44] Rudolf Fleischer and Gerhard Trippen. Kayles on the way to the stars. In *International Conference on Computers and Games*, pages 232–245. Springer, 2004.

[45] Florent Foucaud, Iiro Honkala, Tero Laihonen, Aline Parreau, and Guillem Perarnau. Locally identifying colourings for graphs with given maximum degree. *Discrete Mathematics*, 312(10):1832–1837, 2012.

[46] Aviezri S Fraenkel. Complexity, appeal and challenges of combinatorial games. *Theoretical Computer Science*, 313(3):393–415, 2004.

[47] Zoltán Füredi. The maximum number of edges in a minimal graph of diameter 2. *Journal of Graph Theory*, 16(1):81–98, 1992.

[48] Martin Gardner. Mathematical games. *Scientific American*, 244(4), 1981.

[49] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[50] Daniel Gonçalves, Aline Parreau, and Alexandre Pinlou. Locally identifying coloring in bounded expansion classes of graphs. *Discrete Applied Mathematics*, 161(18):2946–2951, 2013.

[51] Patrick M Grundy. Mathematics and games. *Eureka*, 2:6–9, 1939.

[52] Richard K Guy and Cedric AB Smith. The g-values of various games. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 514–526. Cambridge University Press, 1956.

[53] András Gyárfás and Jenö Lehel. On-line and first fit colorings of graphs. *Journal of Graph theory*, 12(2):217–227, 1988.

[54] Ervin Győri, Mirko Horňák, Cory Palmer, and Mariusz Woźniak. General neighbour-distinguishing index of a graph. *Discrete Mathematics*, 308(5-6):827–831, 2008.

[55] Ervin Győri and Cory Palmer. A new type of edge-derived vertex coloring. *Discrete Mathematics*, 309(22):6344–6352, 2009.

[56] Alfred W Hales and Robert I Jewett. Regularity and positional games. In *Classic Papers in Combinatorics*, pages 320–327. Springer, 2009.

[57] Denis Hanson and Ping Wang. A note on extremal total domination edge critical graphs. *Utilitas Mathematica*, 63:89–96, 2003.

[58] Frank Harary and Michael Plantholt. The point-distinguishing chromatic index. *Graphs and Applications (F. Harary and JS Maybee, Eds.), Wiley-Interscience, New York*, pages 147–162, 1985.

[59] Frédéric Havet and Xuding Zhu. The game grundy number of graphs. *Journal of Combinatorial Optimization*, 25(4):752–765, 2013.

[60] Teresa Haynes and Michael Henning. A characterization of diameter-2-critical graphs with no antihole of length four. *Open Mathematics*, 10(3):1125–1132, 2012.

[61] Teresa Haynes, Michael Henning, Lucas Merwe, and Anders Yeo. A maximum degree theorem for diameter-2-critical graphs. *Open Mathematics*, 12(12):1882–1889, 2014.

[62] Teresa W Haynes and Michael A Henning. A characterization of diameter-2-critical graphs whose complements are diamond-free. *Discrete Applied Mathematics*, 160(13-14):1979–1985, 2012.

[63] Teresa W Haynes and Michael A Henning. A characterization of p5-free, diameter-2-critical graphs. *Discrete Applied Mathematics*, 169:135–139, 2014.

[64] Teresa W Haynes, Michael A Henning, Lucas C Van Der Merwe, and Anders Yeo. On a conjecture of murty and simon on diameter 2-critical graphs. *Discrete Mathematics*, 311(17):1918–1924, 2011.

[65] Teresa W Haynes, Michael A Henning, and Anders Yeo. A proof of a conjecture on diameter 2-critical graphs whose complements are claw-free. *Discrete Optimization*, 8(3):495–501, 2011.

[66] Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. *Positional games*. Springer, 2014.

[67] Nhan Bao Ho. On the expansion of three-element subtraction sets. *Theoretical Computer Science*, 582:35–47, 2015.

[68] Mirko Hornak and Roman Sotak. The fifth jump of the point-distinguishing chromatic index of kn, kn. *Ars Combinatoria*, 42:233–242, 1996.

[69] Mirko Horňák and Roman Soták. Localization of jumps of the point-distinguishing chromatic index of $k\_\{n, n\}$. *Discussiones Mathematicae Graph Theory*, 17(2):243–251, 1997.

[70] Mirko Hornak and N Zagaglia. On the point-distinguishing chromatic index of complete bipartite graphs. *Ars Combinatoria*, 80:75–85, 2006.

[71] S Howse and Richard J Nowakowski. Periodicity and arithmetic-periodicity in hexadecimal games. *Theoretical computer science*, 313(3):463–472, 2004.

[72] Melissa Huggan. *Impartial Intersection Restriction Games*. PhD thesis, Carleton University Ottawa, 2015.

[73] Afrouz Jabalameli, Morteza Saghafian, MohammadMahdi Shokri, Mohsen Ferdosi, Sorush Bahariyan, et al. Improving the bounds on murty_simon conjecture. *arXiv preprint arXiv:1610.00360*, 2016.

[74] Michał Karoński, Tomasz Łuczak, and Andrew Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, 1(91):151–157, 2004.

[75] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[76] Tomasz Krawczyk and Bartosz Walczak. Asymmetric coloring games on incomparability graphs. *Electronic Notes in Discrete Mathematics*, 49:803–811, 2015.

[77] Michael Lampis and Valia Mitsou. The computational complexity of the game of set and its theoretical applications. In *Latin American Symposium on Theoretical Informatics*, pages 24–34. Springer, 2014.

[78] Julien Lemoine and Simon Viennot. Computation records of normal and misère cram. `http://sprouts.tuxfamily.org/wiki/doku.php?id=records#cram`, accédé le 18 juin 2018.

[79] Tong Li, Cunquan Qu, Guanghui Wang, and Xiaowei Yu. Neighbor product distinguishing total colorings. *Journal of Combinatorial Optimization*, 33(1):237–253, 2017.

[80] S Lo. On edge-graceful labelings of graphs. *Congr. Numer*, 50:231–241, 1985.

[81] Willem Mantel. Problem 28. *Wiskundige Opgaven*, 10(60-61):320, 1907.

[82] Nicolas Martins and Rudini Sampaio. Locally identifying coloring of graphs with few p4s. *Theoretical Computer Science*, 707:69–76, 2018.

[83] Hassan Noon and Glen Van Brummelen. The non-attacking queens game. *The College Mathematics Journal*, 37(3):223–227, 2006.

[84] Cormac O'Sullivan. A vertex and edge deletion game on graphs. *arXiv preprint arXiv:1709.01354*, 2017.

[85] Aline Parreau. *Problèmes d'identification dans les graphes.* PhD thesis, Université de Grenoble, 2012.

[86] James B Phillips and Peter J Slater. An introduction to graph competition independence and enclaveless parameters. *Graph Theory Notes of New York*, 41:37–41, 2001.

[87] Jan Plesnik. Critical graphs of given diameter. *Acta Fac. Rerum Natur. Univ. Comenian. Math*, 30:71–93, 1975.

[88] N Zagaglia Salvi. On the point-distinguishing chromatic index of kn, n. *Ars Combin. B*, 25:93–104, 1988.

[89] N Zagaglia Salvi. On the value of the point-distinguishing chromatic index of kn, n. *ARS COMBINATORIA*, 29:235–244, 1990.

[90] Thomas J Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16(2):185–225, 1978.

[91] Robert Scheidweiler. Neighbour distinguishing gap colorings. *Cycles and Colourings 2012*, page 33.

[92] Simon Schmidt. *Jeux à objectif compétitif sur les graphes.* PhD thesis, Université Grenoble Alpes, 2016.

[93] Ben Seamone. The 1-2-3 conjecture and related problems: a survey. *arXiv preprint arXiv:1211.5122*, 2012.

[94] Ben Seamone and Brett Stevens. Sequence variations of the 1-2-3 conjecture and irregularity strength. *arXiv preprint arXiv:1211.0463*, 2012.

[95] Aaron N Siegel. *Combinatorial game theory*, volume 146. American Mathematical Soc., 2013.

[96] Gustavus J Simmons. The game of sim. *Mathematical Solitaires & Games*, page 50, 1980.

[97] Michael Sipser. *Introduction to the Theory of Computation.* Cengage Learning, 2012.

[98] Joanna Skowronek-Kaziów. 1, 2 conjecture—the multiplicative version. *Information Processing Letters*, 107(3-4):93–95, 2008.

[99] Richard Sprague. Über mathematische kampfspiele. *Tohoku Mathematical Journal, First Series*, 41:438–444, 1935.

[100] B Stevens and M Huggan. Polynomial time graph families for arc-kayles. *INTEGERS*, 16, 2016.

[101] Benny Sudakov and Van H Vu. Local resilience of graphs. *Random Structures & Algorithms*, 33(4):409–433, 2008.

[102] Mohammed Amin Tahraoui, Eric Duchêne, and Hamamache Kheddouci. Gap vertex-distinguishing edge colorings of graphs. *Discrete Mathematics*, 312(20):3011–3025, 2012.

[103] L.C. van der Merwe, C.M. Mynhardt, and Teresa Haynes. Total domination edge critical graphs. 54, 11 1998.

[104] Tao Wang. On murty-simon conjecture. *arXiv preprint arXiv:1205.4397*, 2012.

[105] Willem A Wythoff. A modification of the game of nim. *Nieuw Arch. Wisk*, 7:199–202, 1907.

[106] Wenchao Zhang and Xuding Zhu. The game grundy indices of graphs. *Journal of Combinatorial Optimization*, 30(3):596–611, 2015.

[107] Zhongfu Zhang, Linzhong Liu, and Jianfang Wang. Adjacent strong edge coloring of graphs. *Applied Mathematics Letters*, 15(5):623–626, 2002.