FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF MASTER’S THESIS

Title:	Taking and Breaking Games
Student:	Bc. Šimon Lomič
Supervisor:	RNDr. Tomáš Valla, Ph.D.
Study Programme:	Informatics
Study Branch:	System Programming
Department:	Department of Theoretical Computer Science
Validity:	Until the end of winter semester 2020/21

Instructions

Combinatorial games is a class of finite two-player games with full information and no chance. Taking and breaking games are a subclass of combinatorial games involving heaps of tokens, where players alternately choose a heap, remove some tokens and split the remaining heap into several other heaps, according to given particular rules.
The goals of this thesis are:
1) To survey existing results in the field.
2) Try to attack several open problems in solving Subtraction games, Octal and Hexadecimal games, and others.
3) Perform experimental evaluation of various game solving algorithms.

The thesis emphasises theoretical research aspects of the field.

References

Will be provided by the supervisor.

doc. Ing. Jan Janoušek, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague February 19, 2019

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Taking and Breaking Games

Bc. Šimon Lomič

Department of Theoretical Computer Science
Supervisor: RNDr. Tomáš Valla, Ph.D

May 9, 2019

Acknowledgements

First and foremost, I would like to thank my supervisor Tomáš Valla for introducing me into this fascinating topic and for many invaluable insights. Second, I thank my family for supporting me during my studies. Third, I am grateful to my friends and coworkers for listening, offering me advice, and supporting me through this entire process. And last but not least, I want to thank my beloved Patricia for everything she does for me.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on May 9, 2019 . . . .

Citation of this thesis

Lomič, Šimon. *Taking and Breaking Games.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstrakt

V této práci analyzujeme několik otevřených problémů v oblasti nestranných i stranných her typu Taking and Breaking. Pro nestranné odčítací hry dokážeme existenci hry s aperiodickou nim-sekvencí a periodickou sekvencí výhra-prohra. Analyzujeme ekvivalenční třídy těchto her a nalézáme řešení jedné z těchto tříd. Také představujeme novou hru typu Taking and Breaking, kterou z velké části vyřešíme. V oblasti stranných her provedeme analýzu několika odčítacích her a her typu Pure Breaking. Pro tyto hry také představíme obecnou techniku testování aritmetické periodicity. Pro automatické řešení nestranných her typu Taking and Breaking navrhujeme několik algoritmů. Práci uzavíráme důkazem PSPACE-těžkosti nestranné zobecněné odčítací hry a EXPTIME-těžkosti této hry ve stranné variantě.

Klíčová slova kombinatorická teorie her, hry typu Taking and Breaking, Sprague-Grundyho teorie, nestranné hry, stranné hry, odčítací hry, nim-hodnoty

Abstract

In this thesis we examine several open problems in taking and breaking games under the impartial and partizan setting. We prove the existence of an impartial subtraction game with aperiodic nim-sequence and periodic outcome sequence. We also analyze the equivalence classes of subtraction games and provide a solution to one of these classes. We introduce a new taking and breaking game and partially solve it. Then we solve several partizan subtraction games and partizan pure breaking games and describe a general technique for testing arithmetic periodicity of these games. Moreover, we design some game solving algorithms for impartial taking and breaking games. We prove PSPACE-hardness for a generalized subtraction game under the impartial setting and EXPTIME-hardness under the partizan setting.

Keywords combinatorial game theory, taking and breaking games, Sprague-Grundy theory, impartial games, partizan games, subtraction games, nimbers

Contents

List of Figures

List of Tables

List of used symbols

$\mathbb{N}$	Set of all positive integers (natural numbers).
$\mathbb{N}_0$	Set of all non-negative integers.
$[a, b)$	Interval starting at a (inclusive), ending at b (exclusive).
$\lfloor x \rfloor$	Floor of x, the maximal integer $\leqslant x$.
$\lceil x \rceil$	Ceiling of x, the least integer $\geqslant x$.
$\lvert x \rvert_y$	x mod y.
$x \oplus y$	Bitwise sum of integers, also called a *nim-sum*.
$\mathcal{O}(n)$	The standard "Big O" notation that describes the limiting behavior of functions.
$\mathit{mex}(S)$	The minimal excluded value in S.
Σ	An alphabet (any finite set).
Σ^*	Set of all words in an alphabet Σ.
$\{g^L \mid g^R\}$	the game; g^L and g^R are sets options of Left and Right player.
G^L, G^R	A typical option of Left (Right) player in the game G.
$G + H$	$:= \{G^L + H, G + H^L \mid G^R + H, G + h^R\}$, a sum of games.
$G' \in G$	G' is an option of the impartial game G.
$n \cdot G$	A scalar multiple of the game.
$G \parallel H$	Game G is incomparable with the game H.
$G \,\Vert\!\!\triangleright\, H$	Game G is greater or incomparable with the game H.
$G \,\triangleleft\!\!\Vert\, H$	Game G is less than or incomparable with the game H.
$\mathcal{L}, \mathcal{R}, \mathcal{P}, \mathcal{N}$	Sets of left, right, lost and won games, respectively.
$\Phi(G)$	A value of the game based on the game mapping function Φ.
$\mathrm{b}(G)$	Birthday of the game G.
$\mathcal{G}(G)$	Grundy value of an impartial game G.
$\mathcal{C}(G)$	Canonical form of the game G.
$\mathit{rcf}(G)$	Reduced canonical form of the game G.
$o(G)$	Reduced canonical form of the game G.
$\mathbb{G}$	A group of all game values.
$\mathbb{G}_n$	A group of game values born by the day n.

Some simplest games are denoted by the following standard notation:

$$
\begin{array}{rclrclrcl}
0 & := & \{|\} & * & := & \{0|0\} & 1 & := & \{0|\} \\
\uparrow & := & \{0|*\} & \downarrow & := & \{*|0\} & -1 & := & \{|0\} \\
\uparrow n* & := & n \cdot \uparrow + * & \uparrow n & := & n \cdot \uparrow & \frac{1}{2} & := & \{0|*\} \\
\Uparrow & := & \uparrow + \uparrow & \Downarrow & := & \downarrow + \uparrow & \frac{3}{2} & := & \{1|2\} \\
\uparrow^{[2]} & := & \{\uparrow|*\} & \downarrow^{[2]} & := & \{*|\uparrow\} & & & \\
*n & := & \{*0, *1, \ldots, *(n-1)\} & & & & & &
\end{array}
$$

The following table summarizes our naming conventions for various mathematical and game entities:

Font	**Example**	**Represents**
Lowercase Italic	*first*, *even*, ...	Emphasizing term
Lowercase Bold	**game**, **ruleset**, ...	Introducing a new term
Capital Roman	$G, H, I, J, \ldots$	Games and their values
	$A, B, C, S, T, K, \ldots$	Sets of integers
Calligraphic	$\mathcal{L}, \mathcal{R}, \mathcal{P}, \mathcal{N}$	Sets of games based on their outcome
	$\mathcal{A}, \mathcal{T}, \mathcal{H}, \ldots$	Sets of integers with special meaning
	$\mathcal{S}, \mathcal{SD}, \mathcal{PB} \ldots$	Games which are represented by sets
Lowercase Calligraphic	rcf, mex, ...	Functions with special meaning
Lowercase Roman	$x, y, z, a, b, c, \ldots$	Integer variables
Sans Serif	P, NP, AC^0, PSPACE	Complexity classes
	D, $\mathsf{d_0{\cdot}d_1d_2d_3}$, $\mathsf{4{\cdot}421}$, ...	Code-digit games
Capital Greek	$\Gamma, \Lambda, \ldots$	Rulesets
Small Caps	OUTCOME, NIM, ...	Ruleset names and problems

Introduction

People have been playing games in some form since the earliest civilizations first arose over 5,000 years ago. To explore the nature of these games, we could consider their two not necessarily disjunct subclasses: the games that are challenging to the point that people will enjoy playing them (for example CHESS) and games that are challenging to mathematicians, to play and ponder about, but not necessarily engaging to ordinary people (like the game NIM). Interestingly, both of these classes usually tend to be computationally intractable, which makes them a great way to challenge our mathematical methods. Games can be played by two or more players or even by teams of players, but there also exist single player games (also called *puzzles*) and even zero-player games (e.g. Conway's Game of Life).

Combinatorial game theory studies games of pure strategy played by two players who alternate moves without using dice or other chance elements. Also, there is no hidden information, such as cards that can be seen only by their owners. Furthermore, the outcome of these games is restricted to winning or losing and no draws are allowed.

Taking and breaking games involve heaps of tokens. The players alternate in choosing some particular heap and removing some tokens from it while splitting the remaining tokens in the heap into several smaller heaps. When one hears this simple definition, he might not realize the hidden complexity of answering even simple questions about these games such as "Who will win the game if both players play optimally?" or "Which move should I play to ensure victory?".

Together with the famous game of NIM, Taking and breaking games are among the earliest and most studied combinatorial games. Maybe for their apparent simplicity, these games were the first impulse for many people to build this beautiful theory. The appeal of studying the games with heaps lies in their simple definition of a position (a collection of numbers), in their flexibility to various restrictions while maintaining an infinite class of games, and above all, their property that moves decompose the game into several disjoint subgames that do not influence each other.

While a new theory that allows us to understand a bit more about these games is being crafted, new questions arise and make us even today realize that we are still at the beginning of the path to understanding them completely. In addition to a natural appeal of games, there are applications and connections to various areas including complexity, logic, graph, and matroid theory, networks, error correcting codes, surreal numbers, on-line algorithms and biology [25, ch. 7].

In this thesis, we aim at presenting an in-depth survey for the topic of taking and breaking games. Our goal is to identify several open problems in this field and make an attempt to attack them by applying the presented theory. This will be done both from the point of view of combinatorics and by applying algorithmic approaches for game solving.

Standard References

The bibliography at the end of this thesis lists many references to refer to for more details about particular topics. Several of them are classics that stand out as key sources for the basics of the theory discussed here. To distinguish them from other references, we reference these books using the following abbreviations in brackets:

- *Winning Ways for Your Mathematical Plays*, by Berlekamp, Conway, and Guy [42, WW]. Originally published in 1982, immediately recognized as a cornerstone of the recreational mathematics literature, which transformed the field of combinatorial game theory from recreational pastime activity into a mature mathematical discipline.

- *On Numbers and Games*, by Conway [10, ONAG], the definitive source for the axiomatic theory of combinatorial games, which was published in 1976. Among others, it introduced the first coherent theory for partizan games.

- *Lessons in play: an introduction to combinatorial game theory*, by Albert, Nowakowski, and Wolfe [1, LIP]. Published in 2007, this introductory textbook on combinatorial games contains many examples and proofs that are not available elsewhere.

- *Combinatorial game theory*, by Siegel [71, CG]. Comprehensive and up-to-date textbook which provides an in-depth presentation of most of the topics studied in combinatorial game theory. Probably the most definitive work on the subject.

- *Games of No Chance*, an ongoing series of five volumes of papers published by Cambridge University Press [55, 58, 56, 57, 48, GONC]. These papers are generally of very high quality. Each volume also contains an updated list of unsolved problems.

Organization of the Thesis

In Chapter 1 we review some mathematical background which is necessary for the theory discussed in this thesis.

The basic theory of combinatorial games is described in Chapter 2. We start by describing the disjunctive theory for partizan games. Then we narrow the theory down to impartial games. The last section of this chapter describes algorithmic and complexity topics related to combinatorial games.

Chapter 3 then focuses on the main topic of this thesis, the taking and breaking games. We survey state-of-art results on subtraction, octal and hexadecimal games and on their partizan generalizations.

We intend for this text to be self-contained.Therefore in Chapters 2 and 3 we mention many already published proofs, which we occasionally slightly simplify or revise in order to match the approaches applied in this thesis.

Chapter 4 presents our contributions to the field, including several results on subtraction games, code-digit games in general and their partizan variants. The last section lists some game solving algorithms, a few computational results and also a hardness proof for a generalization of subtraction games.

Chapter **1**

Preliminaries

An understanding of combinatorial game theory requires knowledge in a wide range of mathematical subjects. This chapter gives an overview of the mathematical background that is essential for the theory discussed in the next chapters. It is provided primarily for reference and review, detailed proofs and examples are omitted. Most definitions are laid out in a similar way as they appear in FIT CTU courses, namely BI-AG1, BI-ZDM, MI-CPX, and MI-MPI.

Through the text, we use standard notations. By $\mathbb{N}$ we denote the set of all positive integers (natural numbers), whereas N_0 denotes all non-negative integers. $\mathbb{Z}$ denotes the set of all integers, while Z_n is the set of all integers modulo n. $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}^+$ the set of strictly positive real numbers. If $x \in \mathbb{R}$, then $\lfloor x \rfloor$ denotes the maximal integer $\leqslant x$ (the **floor** of x), and $\lceil x \rceil$ denotes the least integer $\geqslant x$ (the **ceiling** of x). We will also denote the set of odd integers *odds* and the set of even integers *evens*.

Set of all subsets of a set S (the **power set**) is denoted $\mathcal{P}(S)$ and $|S|$ denotes the cardinality of the set S. We say that $A_1, A_2, \ldots, A_k$ is a **partition** of a set S if and only if $\bigcup_{i=1}^{k} A_i = S$ and for any $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$ implies $A_i \cap A_j = \emptyset$. If a set $A \subseteq B$ but $A \neq B$, we say it is a **proper** subset, denoted $A \subset B$.

A **groupoid** is an ordered pair $(S, \circ)$ where S is a set and $\circ$ a binary operation $\circ : S \times S \to S$. A **semigroup** is an associative groupoid, a semigroup where exists an **identity** element e such that $e \circ s = s \circ e$ for all s is called **monoid**. A monoid where for each element $s \in S$ exists the **inversion** element s^{-1} such that $s \circ s^{-1} = s^{-1} \circ s = e$ is called **group**. Furthermore, if $\circ$ is commutative, we say that $(S, \circ)$ is **commutative** (groupoid, semigroup, monoid, group). The commutative group is known as the **Abelian group**. So if $(S, \circ)$ is an Abelian group with an identity e then for any $a, b, c \in S$ we have

(a) (associativity) $(a \circ b) \circ c = a \circ (b \circ c)$;

(b) (neutrality) $a \circ 0 = 0 \circ a = a$;

(c) (inverse) $a \circ a = e$;

(d) (commutativity) $a \circ b = b \circ a$.

A **binary relation** R on sets A and B is a subset of their Cartesian product $A \times B$. A **partially ordered set** (or just **poset**) is an ordered pair $(S, \leqslant)$ where S is a set and $\leqslant$ a binary relation on S, that satisfies the following axioms:

(a) (reflexivity) $x \leqslant x$ for all $x \in S$;

(b) (antisymmetry) $x \leqslant y$ and $y \leqslant x$ implies $x = y$ for all $x, y \in S$;

(c) (transitivity) $x \leqslant y$ and $y \leqslant z$ imply $x \leqslant z$ for all $x, y, z \in S$.

The relation $\leqslant$ is then called a **partial order** of the set S. Two elements $a, b \in S$ are **comparable** if $a \leqslant b$ or $b \leqslant a$. Otherwise they are **incomparable**. We say that a **covers** b whenever $a \leqslant b$ and there is no c such that $a \leqslant c \leqslant b$. A **Hasse diagram** of a poset $(S, \leqslant)$ is its drawing where each element is represented by a vertex in the plane and two elements $a, b \in S$ are connected with a curve that goes upwards from a to b whenever y covers x. A poset is called a **lattice** if it has a **top element**, denoted $\top$, such that for any $s \in S$ is $s \leqslant \top$, and also a **bottom element** denoted $\bot$, such that or any $s \in S$ is $\bot \leqslant s$. A poset where all pairs are comparable is called a **linear ordering**.

An **equivalence relation** $\sim$ on a set S, is a binary relation that is reflexive ($x \sim x$), symmetric ($x \sim y$ if and only if $y \sim x$) and transitive ($x \sim y$ and $y \sim z$ implies $x \sim z$). A subset $T \subset S$ such that $a \sim b$ for any $a, b \in T$ and there is no $c \in S$ with $c \sim a$ and $c \notin T$, is called an **equivalence class** of x by $\sim$. The equivalence class is denoted by $[x]_\sim := \{s \in S : s \sim x\}$.

Graph Theory

A **graph** is an ordered pair (V, E). V is a set of its vertices and E is a set of its edges where each edge is an unordered pair of distinct vertices from V. Given a graph G, we will denote its set of vertices as $V(G)$, and its set of edges as $E(G)$. Graph H is a **subgraph** of graph G (denoted $H \subseteq G$) when $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

A **degree** of a vertex v in graph G (denoted $\deg(v)$) is a number of edges in $E(G)$ containing the vertex v. A vertex v in G is **adjacent** to a vertex u (or neighbor of u) if $\{u, v\} \in E(G)$.

By $N(v)$ we denote the set of adjacent vertices of v in G. A **path** of length $n \geqslant 0, n \in N$ is a graph P_n such that:

$$P_n = \Big(\{0, 1, \dots, n\},\ \{\{i, i+1\} : i \in \{0, \dots, n-1\}\}\Big).$$

A **cycle** of length $n \geqslant 0, n \in N$ is a graph C_n such that:

$$C_n = \Big(\{0, 1, \dots, n\},\ \{\{i, i+1\} : i \in \{0, \dots, n-1\}\} \cup \{\{n, 1\}\}\Big).$$

We call a connected graph that does not contain a cycle as a subgraph a **tree**. A **rooted tree** is an ordered triple (V, E, r) where a vertex v in graph G is a **list** if $\deg(v) = 1$.

Graph G is **connected**, if there is a path in G between each pair of vertices in $V(G)$. A **connected component** in graph G is any maximal (in inclusion) connected subgraph of G. We call a graph G **bipartite** if a set of vertices $V(G)$ can be decomposed into two disjoint sets such that no two vertices within the same set are adjacent. A **vertex cover** is a set of vertices such that each edge of the graph is incident to at least one vertex of the set.

A **directed graph** (a **digraph**) is an ordered pair (V, E). V is a set of its vertices and E is a set of its *directed* edges, meaning that each edge is an *ordered* pair of distinct vertices from V. For a directed edge $(u, v) \in E$ we say that it is **starting** in u and **ending** in v. In a directed graph $D = (V, E)$ we define the **in-degree** $\deg^-(u)$ and the **out-degree** $\deg^+(u)$ as a number of edges ending and starting in the vertex u, respectively. A vertex u with a $\deg^-(u) = 0$ is called a **source**, while a vertex v with $\deg^+(v)$ is called a **sink**. A **topological ordering** of a digraph G is a linear ordering of its vertices such that for every directed edge $(u, v) \in E(G)$, u comes before v in the ordering.

Boolean Algebra

A **Boolean formula** is a well-formed parenthesized expression involving **variables** (symbols denoted by small letters x, y, z), the binary connectives $\wedge$ (conjunction), $\vee$ (disjunction), the unary connective $\neg$ (negation), and parentheses. If X is a set of variables, then $F(X)$ denotes a Boolean formula over variables in X.

An **assignment** α is a function from set of variables X to the set $\{0, 1\}$ where 0 and 1 in this context denote Boolean values *true* and *false*. The **value** of a formula $F(X)$ under assignment α is the evaluation of the formula when replacing the variables by their assigned value. A **literal** is either x or $\neg x$ where x is a variable.

A formula is in **conjunctive normal form** (CNF) if it is a conjunction of disjunctions of literals. A formula is in **disjunctive normal form** (DNF) if it is a disjunction of conjunctions of literals. For positive integer k we say that formula is in k-DNF, if it has the form $C_1 \vee C_2 \vee \ldots \vee C_k$ where each $C_i (1 \leqslant i \leqslant k)$ is a conjunction of at most k literals; k-CNF is defined dually. The term C_i in this form is called a **clause**.

Let F and G be Boolean formulas. **De Morgan's laws** tell us that $\neg(F \vee G) \Leftrightarrow \neg F \wedge \neg G$ and $\neg(F \wedge G) \Leftrightarrow \neg F \vee \neg G$.

Complexity Theory

An **alphabet** is any finite set, usually denoted as Σ. Elements of the alphabet are called **symbols**. A **word** or a **string** in an alphabet is a finite sequence of symbols. We denote by Σ^* the set of all words in the alphabet Σ and by $|y|$ the length of the word y, that is the number of (occurrences of) symbols in it. A language is a set of words. In a **decision problem**, we are given some input and a question and the task is to decide whether the answer to this question is *yes* or *no*. When studying decision problems we encode them in some suitable alphabet. Most often the alphabet $\{0, 1\}$ can be used for our purposes. This way the set of all inputs to the decision problem with the positive answer corresponds to some language over this alphabet. Hence we use the terms decision problem and language as synonyms.

The **time complexity** of an algorithm is the maximum number of steps it performs on inputs of a given length. The running time is usually measured in so-called *"Big O"* notation that describes the limiting behavior of this function. Let $f(n)$ and $g(n)$ be finite and positive functions. We then say that $g(n)$ is an **asymptotic upper bound** of $g(n)$ (denoted $f(n) = \mathcal{O}(g(n))$) if

$$(\exists c \in \mathbb{R}^+)\ (\exists n_0 \in \mathbb{N}^+)\ (\forall n \geqslant n_0)\ f(n) \leqslant c \cdot g(n).$$

RAM Model

When talking about algorithms, we will describe them and measure their complexity on the model of computation called the **Random Access Machine** (RAM). Each of this family of models consists of a separated program and data memory. The data memory is made of integer cells that are addressable by integers. The **program**, stored in the program memory, is a sequence of **instructions** that will also be sequentially executed. There are two types of instructions: arithmetical and control instructions. Arithmetical instructions usually take two input arguments and a single output argument. The arguments can be either constants, direct or indirect addresses to the data memory. The control instructions are jumps to a specific location in the program memory, conditional jumps and the halting instruction. At the start of the program, the input is written at the beginning of the data memory. After the program halts, the output is stored on agreed location in the data memory.

When it will be necessary, we will consider the RAM model that limits the size of integers stored in a single memory cell to some w bits. Let us denote $R(n)$ as the **binary representation** of the integer n in a two's complement representation. We will consider the following set of instructions on integers, all of which can be executed in constant time: AND, OR, XOR (all performed bitwise), $+$ (addition), $-$ (subtraction), $*$ (multiplication), $/$ (division), % (modulo), NOT (bitwise negation), $\gg$ (right bit-shift), $\ll$ (left bit-shift) and $>, <, =$ (comparison). Many other constant time operation can be implemented using these basic ones. Mareš describes how to define the operation populationCount(n) (the number of one-bits in $R(n)$) and the operation LSB(n) (the position of the *least significant one-bit*).

Sometimes we will denote the bitwise XOR operation as $\oplus$. This operation has the following interesting properties

Lemma 1.1 (Properties of bitwise XOR)**.** Let $a, b, c \in \mathbb{N}_0$. Then:

(a) *(commutativity)* $a \oplus b = b \oplus a$

(b) *(associativity)* $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

(c) *(neutrality)* $a \oplus 0 = 0 \oplus a = a$

(d) *(inverse)* $a \oplus a = 0$

(e) $a \oplus b \oplus c = 0$ if and only if $a \oplus b = c$

Proof. The properties (a) - (d) are well known and follow easily from the definition. To prove the last one, let us show, that for any $a, b \in \mathbb{N}_0 : a \oplus b = 0 \Leftrightarrow a = b$. For contradiction, suppose that $a \neq b$. Then, $(a)_2 \neq (b)_2$, and let us assume that they differ at i-th bit. However, by definition of xor, $(a \oplus b)_2$ must both have their i-th bit turned on which is a contradiction. $\square$

Complexity Classes

When talking about complexity of problems, the **Turing Machine** (TM) model is used more often. For a proper definition of this model, we refer the reader to [3, p. 20]. The class of all decision problems having a polynomial algorithm that correctly decides whether the input string is in the corresponding language or not is called P. Such problems are called **polynomially solvable**. The class NP (stands for nondeterministic polynomial time) consists of all problems that can be solved in polynomial time by a non-deterministic

Turing machine. Alternatively, these are such problems that their solution is of polynomial size and can be verified by a deterministic Turing machine running in a polynomial time. PSPACE is the class of problems that can be solved by a deterministic Turing machine working in a polynomially limited memory. Problems in $\mathsf{EXPTIME}$ can be solved by a deterministic Turing machine in a time $2^{P(n)}$ for some polynomial P and the class $\mathsf{EXPSPACE}$ is the class of problems that can be solved by a deterministic Turing machine working in memory limited by $2^{P(n)}$ for some polynomial P. Problems in $\mathsf{NEXPTIME}$ can be solved by a non-deterministic Turing machine working in time limited by $2^{P(n)}$ for some polynomial P.

A **polynomial-time reduction** of a problem $P \subset \Sigma^*$ to a problem $Q \subset \Sigma^*$ is a function $\rho : \{0,1\}^* \to \{0,1\}^*$ such that ρ can be computed in polynomially bounded time and $w \in P$ if and only if $\rho(w) \in Q$. We say that P can be polynomially reduced to Q, and denote $P \lhd Q$. Let $\mathcal{C}$ be a complexity class and let $H \in \mathcal{C}$. We say that the problem is $\mathcal{C}$-**hard** if it holds that

$$(\forall P \in \mathcal{C})\ P \lhd H.$$

The problem H is $\mathcal{C}$-**complete** if it is $\mathcal{C}$-hard and $H \in \mathcal{C}$.

An n-input, single output **Boolean circuit** C is a directed acyclic oriented graph with n sources and one sink. All vertices that are neither source or sink are called **gates**. The gates are labeled with logical operations $\wedge$, $\vee$ and $\neg$ and for any such vertex u, the number $\deg^-(u)$ of these vertices is called **fan-in**. The **size** of the circuit, denoted $|C|$, is the number of vertices in it. The **depth** of an circuit is the length of the longest distance between a source and a sink. For an input $x \in 0, 1^n$, the output of C on x, denoted by $C(x)$, is defined in the natural way. We say that a circuit is AC^0 if it has constant depth and gates labeled with $\neg$ to have in-degree 1 and to be located immediately after the inputs. The complexity class AC^0 is then defined as the set of problems that are computable using an AC^0 circuit.

Chapter 2

Combinatorial Game Theory

In this chapter we will introduce the parts of combinatorial game theory that will be necessary for the analysis of taking and breaking games. Most of the following definitions and properties can be found in all the classical textbooks on combinatorial games [71, 1, 10, CG, LIP, ONAG]. The layout of this theory has been inspired by the way it is presented in the FIT CTU course MI-ATH.

Combinatorial games are two-player games with no hidden information and no chance elements. We imagine them played on some kind of **board** between two players whose usual names are **Left** and **Right**[1], who play alternately and whose moves affect the **position** (configuration of the board) in a manner defined by the **ruleset** (or rules) of the game. Both players have complete knowledge of the game state at all times ("complete information") and the effect of each move is determined before the move is made ("no chance elements").

Definition 2.1. **Game** G is a structure $\left\{g^L \mid g^R\right\}$, where

$$\begin{aligned} g^L &= \{G^L \mid L \text{ can move from } G \text{ to } G^L\} \text{ and} \\ g^R &= \{G^R \mid R \text{ can move from } G \text{ to } G^R\}. \end{aligned}$$

It is possible that $g^L, g^R = \emptyset$. We call the sets g^L and g^R the **Left** and **Right options**, respectively. We will denote a typical option of Left (Right) player G^L (G^R), respectively.

To avoid a confusion, throughout this thesis we will use the term **game** to refer to an individual position and not to the **ruleset**. The ruleset can be thought of as the definition of the layout of the moves from particular positions. In the formal sense, ruleset is simply a set of games (structures from the definition 2.1) that we play on and a move is a replacement of a

[1] Also known under aliases *Black* and *White*, *Red* and *Blue*, *Alice* and *Bob*, etc.

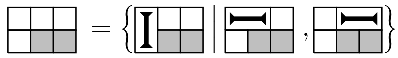


Figure 2.1: Position in the game of Domineering.

game by another game which is usually a modification of the first. To keep the distinction clear, we will use Roman capital letters $G, H, \ldots$ to denote games and capital Greek letters $\Gamma, \Lambda, \ldots$ to denote rulesets. When necessary, a position in a game which follows some ruleset Γ will be called a **Γ-position**.

Example 2.2. The game Domineering is usually played on an $n \times m$ checkerboard. On each turn, Left player can place a vertically rotated domino piece (of size 1×2) on any pair of free squares (without overlapping). Similarly, Right player can place a single horizontally rotated domino piece if possible.

This was an example of a ruleset while in Figure 2.1 is an example of a position (a game) of Domineering in which Left has one option and Right has two options. (One cannot play onto a gray square).

Definition 2.3. A **run** of G is a sequence of positions $G = G_0, G_1, \ldots$ such that G_{i+1} is an option of G_i. A **play** of G is a run $G = G_0, G_1, \ldots$ which alternates between Left and Right moves, i.e., if G_{i+1} is a Left (Right) option of G_i, then G_{i+2} is a Right (Left) option of G_{i+1}, respectively.

Definition 2.4. We say that the game H is a **subposition** of the game G if there exists a (possibly empty) sequence of consecutive moves (not necessarily alternating) leading from G to H.

Definition 2.5. Short games are combinatorial games that admit only finite play and have finite sets of options. That is, there does not exist a play that has an infinite length and the sets of Left and Right options (g^L and g^R) are finite for each subposition of G.

In this thesis we will consider only short games. The theory of so called **loopy** games, in which the plays need not be finite, has been developed by Conway, Li and Siegel in [10, 52, 72] and is surveyed in [71, CG, Chapter VI]

Definition 2.6. The **game tree** of a G is a rooted tree with vertices corresponding to subpositions of G. Each vertex of some position H has a left edge for each Left move $H^L \in h^L$ and a right edge for each Right move $H^R \in h^R$. The vertices distinguish between positions to which we got through a different run of G, so there is a one-to-one correspondence between runs of G and vertices of the game tree of G.

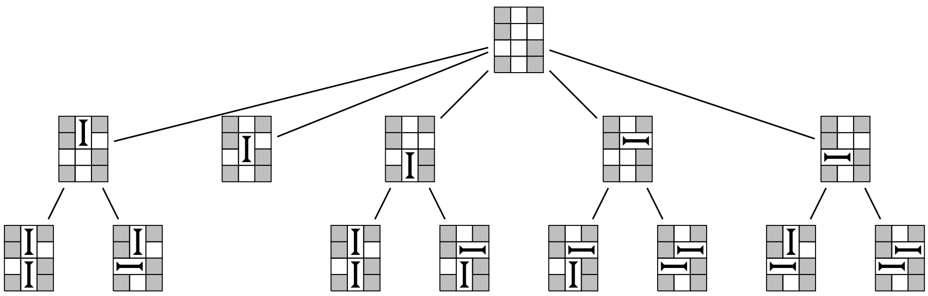


Figure 2.2: Game tree of a DOMINEERING position.

Figure 2.2 shows a game tree of a position in DOMINEERING. The owner of the edge (Left or Right) visualized by their slanting (left or right). Notice that the tree has nine lists even though there are only eight distinct **final positions** (positions that have no option).

Definition 2.7. Based on the mutual relationships of g^L and g^R over all positions we distinguish following game classes:

- **Impartial games**, when for every position G the sets of left moves and right moves are the same: $g^L \cap g^R = \emptyset$ holds. For example, DOMINEERING is not impartial. We will see that this property makes the analysis significantly simpler.
- **Black and White games** are games in which there is no position in which would Left and Right share any of their moves: $g^L = g^R$ holds. These games lie in some sense on the other side of the spectra of games than impartial games. Unfortunately, this restriction does not simplify the analysis of these games as much as for the impartial games.
- **All-small games**[2] have a similar restriction like impartial games in a more relaxed setup: A game G is All-small if for any position Left has an option if and only if Right has an option. For instance, games $0, \uparrow, \downarrow, *$ are All-small, but $1, 2, -3$ are not.

If we want to emphasize that a game is not impartial, we say it is **partizan**.

Now we have defined what a game is and how it is played, it remains to explain what does it mean to *win* a game. There are two standard conventions under which we define the **outcome** of a game:

[2] Also known as *dicotic*.

Definition 2.8. In a game played under the **normal play** convention the player who makes the last move, *wins*. In a game which follows the **misère play** convention, the player who makes the last move, *loses*.

The normal convention is obviously more natural: in a non-final position we find ourselves losing if we are unable to find any good move, so it makes complete sense to lose when we cannot find any move at all. This property of unnatural setup of misère play is a source of many difficulties in trying to understand the combinatorial structure of these games.

Definition 2.9. We say that Left player has a **winning strategy** as a *first* player, if there exists an integer n such that

$$(\exists G_1 \in g^L)(\forall G_2 \in g_1^R)(\exists G_3 \in g_2^L)(\forall G_4 \in g_3^R)\ldots \begin{cases} g_n^R = \varnothing & \text{under a normal play;} \\ g_n^L = \varnothing & \text{under a misère play,} \end{cases}$$

and has a winning strategy as a *second* player, if

$$(\forall G_1 \in g^L)(\exists G_2 \in g_1^R)(\forall G_3 \in g_2^L)(\exists G_4 \in g_3^R)\ldots \begin{cases} g_n^R = \varnothing & \text{under a normal play;} \\ g_n^L = \varnothing & \text{under a misère play.} \end{cases}$$

The winning strategy for Right player could be defined symmetrically.

Theorem 2.10 (Fundamental Theorem of Combinatorial Games)**.** Let $G = \{g^L \mid g^R\}$ be a short combinatorial game. Then Left has a winning strategy as a first or else Right has a winning strategy as a second, but not both.

Proof. (By induction on the number of subpositions of G). Let G^L be any option of G. It must have strictly fewer subpositions than G (at least the subposition G is missing). So by induction hypothesis, we may assume that either Left has a winning strategy as a first or Right has a winning strategy as a second. Note that because we could just rename the players, by symmetry this is equivalent to Right having a winning strategy as a first or Left having a winning strategy as a second.

So if Right has a winning strategy in all G^L playing first, he can definitely win in G playing second no matter what move Left makes. Conversely, if Left can win any of the G^L playing second, then he can simply win G by moving to it. $\square$

The proof of this theorem represents several typical properties of many proofs presented in this thesis. First, it is a proof by induction on some structural property of the game which can be easily applied due to the recursive definition of games. Second, the base case is deliberately omitted for the case when the player has no moves is trivially defined by the convention of the game. And third, since we have chosen the naming of the players, we can at any time switch it, so many properties must hold due to symmetry.

Definition 2.11. We define the **outcome** of G as follows:

$$o(G) = \begin{cases} \mathcal{L} & \text{if Left can win both as a first and a second,} \\ \mathcal{R} & \text{if Right can win both as a first and a second,} \\ \mathcal{P} & \text{if first player always loses and} \\ \mathcal{N} & \text{if first player always wins.} \end{cases}$$

By Theorem 2.10 it follows that the outcome of any game G is always unambiguously defined. So we can partition the set of all the games in such way, that each game belongs into one of the following **outcome classes**:

- $\mathcal{L}$: *left games*, where Left always wins,
- $\mathcal{R}$: *right games*, where Right always wins,
- $\mathcal{P}$: *lost games*, where first player always loses and
- $\mathcal{N}$: *won games*[3], where first player always wins.

The set of all games therefore is the union $\mathcal{L} \cup \mathcal{R} \cup \mathcal{P} \cup \mathcal{N}$.

Corollary 2.12. The proof of Theorem 2.10 provides an algorithm how to recursively calculate the outcome on any game. We will refer to it as **game-tree algorithm**. This algorithm is runs in linear time in the size of the game tree. However, most games are presented in much more succinct encoding than the game tree. Actually, the size of the game tree is for many games usually at least exponential in the size of the description of a game position, therefore this cannot be considered an effective algorithm. (Compare the game size of the game tree for a position of DOMINEERING compared to the size of the description of a position).

The main problems studied in the combinatorial game theory are

- **Outcome**: given a position, determine the outcome class.
- **Winning move**: if a player has a winning strategy, find a winning move among his options.
- Questions about the general combinatorial **structure** of these games.
- **Hardness** results that suggest that there exist games, for which are the above problems intractable.

[3]Also known as *fuzzy games*.

The focus of this thesis will be on *short games* under *normal play* convention, so without stating otherwise, assume that the discussed games belong to these categories.

We will consider a **solution** to a game to be the answer to the most widely studied problem: deciding the *outcome class* of a game.

2.1 Disjunctive Theory

We have seen that although Theorem 2.10 describes an algorithm, to determine the outcome in linear time in the size of the game tree. Without further knowledge obtained from the ruleset of the game, we are not able to do much better.

The *disjunctive theory* studies a fundamental property of many games that have proven useful in the analysis of combinatorial games: a typical position can be broken down into several independent components. We can observe this property in the end-game positions of Domineering. Near the end of the game, the board is almost completely filled with dominoes and the play in the maximal components of adjacent free squares do not influence each other. The same property holds for other so-called *territorial* games, out of which the best known is probably Go. We will see that the Taking and Breaking games which are the main subject of this thesis, have this property throughout the whole play by definition and so they are great candidates for disjunctive analysis. We will say that these games *decompose into a sum*.

Informally, we denote a *disjunctive sum* of games G and H as a composite game, denoted $G + H$, which can be described as a *simultaneous play* of both games in the following manner: a Left player's move consists of selecting one of the component games, say G, and making a legal move into some $G^L \in g^L$, leaving the other component intact. So after the move, the game is played on $G^L + H$, then Right continues analogously and the game continues in this way until some player is unable to make a move in any of the components.

Definition 2.13. The **disjunctive sum** of games $G = \left\{g^L \mid g^R\right\}$ and $H = \left\{h^L \mid h^R\right\}$ is formally defined by

$$G + H := \left\{G^L + H,\ G + H^L \mid G^R + H,\ G + h^R\right\}, \tag{2.1}$$

where G^L and G^R denote that G^L *ranges* over all Left options $G^L \in g^L$ and H^L range over all $H^L \in h^L$. The set of Left options of $G + H$ can be defined as a union

$$\{G^L + H : G^L \in g^L\} \cup \{G + H^L : H^L \in h^L\}, \tag{2.2}$$

but the notation (2.1) is almost always clearer than (2.2) and we will use it through the text without further comments.

Table 2.1: The outcome of a sum of two games based on their outcome under Normal play.

$+$	$\mathcal{L}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{N}$
$\mathcal{L}$	$\mathcal{L}$	$\mathcal{L}$	any	$\mathcal{L} \cup \mathcal{N}$
$\mathcal{P}$	$\mathcal{L}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{N}$
$\mathcal{R}$	any	$\mathcal{R}$	$\mathcal{R}$	$\mathcal{R} \cup \mathcal{N}$
$\mathcal{N}$	$\mathcal{L} \cup \mathcal{N}$	$\mathcal{N}$	$\mathcal{R} \cup \mathcal{N}$	any

When trying to determine the outcome of a game that decomposes into a sum, one obvious question we could be asking is whether it is possible to determine the outcome of a sum $G + H$ solely from the outcomes of some games G and H.

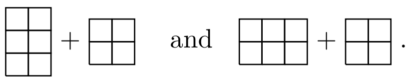
The following proposition shows that the addition of a lost game does not change the outcome.

Proposition 2.14. Let G and H be games and $o(H) = \mathcal{P}$. Then under Normal convention, $o(G) = o(G + H)$.

Proof. Without loss of generality, assume that Left has a winning strategy for G. Let the strategy be as playing first. Then he can also win on $G + H$ by starting with his winning move G^L and then replying to each move of Right into the same component as he played. Because in both G^L and H he has a winning strategy playing second, Right will be the first one out of moves.

Conversely, assume that Left has a winning strategy playing second. Then after any Right's move Left can again reply into the same component he played in. Since $H \in \mathcal{P}$, this strategy ensures that the first move in G will be done by Right and Left then can follow there with his winning strategy. □

Example 2.15. Unfortunately, similar rules cannot be derived for all the combinations of outcomes in the sum. Consider for instance the following Domineering games:



We can see that each component of both games is a won game. However, the outcome of the first sum $\mathcal{L}$ and the outcome of the second sum is $\mathcal{R}$. Therefore, some information about the structure of the game is needed in order to determine the outcome of a sum. The cases when the outcome is clear or partly clear are shown in Table 2.1. Proofs for these properties of sum can be found for example in [1, LIP, p. 55].

Definition 2.16. A set of games $\mathcal{A}$ is **closed**, if for any games $G, H \in \mathcal{A}$

- $G + H \in \mathcal{A}$ (closed under addition).
- For any subposition J of G is $J \in \mathcal{A}$ (*hereditary* closed).

The following notation can be used for a common way of solving a closed set of games $\mathcal{A}$ which decompose into sum.

Definition 2.17. Let $\mathcal{Q}$ be a monoid with binary operation $\cdot$ such that we are able to determine the outcome of any $x \in \mathcal{Q}$. The **game mapping function** Φ for a closed set of games $\mathcal{A}$ with ruleset Γ is a mapping $\Phi_\Gamma : \mathcal{A} \to \mathcal{Q}$ such that for some game $G \in \mathcal{A}$, which is represented as a disjoint sum of its components $G = G_1 + G_2 + \ldots + G_k$ is

$$\Phi_\Gamma(G) = \Phi_\Gamma(G_1) \cdot \Phi_\Gamma(G_2) \cdot \ldots \cdot \Phi_\Gamma(G_k)$$

and we can calculate the outcome $o(G)$ as using $o(\Phi_\Gamma(G))$.

The advantage of this approach is especially useful when there is such monoid $\mathcal{Q}$ and a mapping function Φ for which the calculation of $\Phi(G)$ and $o(\Phi(G))$ can be done more efficiently than calculating the $o(G)$ itself.

In the next section, we will enhance our definition of games by an equivalence and ordering operators which will result in a partially ordered Abelian group that will show to be a suitable structure for this approach on all games under Normal play. We will also introduce a much better structure for Normal impartial games: we will show that suitable monoid for these games is $\mathcal{Q} = \mathbb{N}_0$ where the multiplication $a \cdot b$ is defined as xor in binary.

Note: We will sometimes use the outcome function $o : \mathcal{A} \to Q$ with $Q = \{\mathcal{P}, \mathcal{N}, \mathcal{L}, \mathcal{R}\}$ as a game mapping function, despite the fact that we can not always tell the outcome of a sum of elements from the outcome classes Q (see the Table 2.1). The reason for this is that we are sometimes interested in the single-pile positions only, or we do not mind the limitation of the undefined sum for some cases.

2.2 Canonical Theory

The following definition is motivated by the fact that for solving a game we do sometimes not need to know about the exact structure of the game. For instance, in Proposition 2.14 we have shown that a lost game cannot change the outcome of any game when played in a sum. It follows that any pair of lost games behave the same *in a context* of any game, unlike it was in the example 2.15 for a pair of won games.

Definition 2.18. The games G and H are **equal**, denoted $G = H$ if for every game X is

$$o(G + X) = o(H + X).$$

Conversely, if there exists a game X such that $o(G + X) \neq o(H + X)$, we say that X **distinguishes** G **from** H.

Observation 2.19. $=$ is an equivalence relation on games.

Proposition 2.20. If $G = H$ then for any game J is $G + J = H + J$.

Proof. Since the disjunctive sum is associative and $G = H$, we have

$$o((G + J) + X) = o(G + (J + X)) = o(H + (J + X)) = o((H + J) + X)$$

for any game X. $\square$

Definition 2.21. The **game value** of a game is its equivalence class modulo $=$. The set of all game values is denoted by $\mathbb{G}$.

2.2.1 The Group of Game Values $\mathbb{G}$

To better formulate the induction used in the proof of Theorem 2.10, we assign an integer value to each game based on the observation that more complex games are derived from simpler games.

Definition 2.22. The **birthday** of game G is denoted $\mathrm{b}(G)$ and defined by

$$\begin{aligned} \mathrm{b}(0) &:= 0 \\ \mathrm{b}(G) &:= \max_{H \in g^L \cup g^R} b(H) + 1 \end{aligned}$$

If a game G has birthday $\mathrm{b}(G) = b$, we say it has been **born on** the day b and if $\mathrm{b}(G) \leqslant b$, we say it has been **born by** the day b. The set of game values of all games born by day n we denote $\mathbb{G}_n$. Note that we can recursively define $\mathbb{G}_n$ for $n \geqslant 1$ as follows:

$$\mathbb{G}_n = \Big\{ \{ g^L \mid g^R \} : g^L, g^R \subseteq \mathbb{G}_{n-1} \Big\}.$$

In Figure 2.3 we draw the game trees of the four games born by day one and two games born by day two. They are usually denoted 0 *(zero)*, $1, -1, *$ *(star)*, $\uparrow$ *(up)* and $\downarrow$ *(down)*.

Lemma 2.23. [1, LIP, p. 67] A number of games born by day n (denoted $\mathrm{g}(n)$) is finite.

$0 := \{|\}$ $\quad 1 := \{0\,|\}$ $\quad -1 := \{|\,0\}$ $\quad * := \{0 \mid 0\}$ $\quad \uparrow := \{0 \mid *\}$ $\quad \downarrow := \{* \mid 0\}$

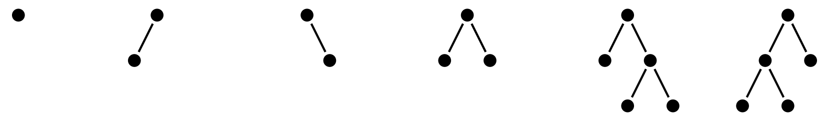


Figure 2.3: Some simplest games and their game trees.

Proof. By induction on n:

(a) If $n = 0$, then $\mathrm{g}(n) = 1$.

(b) If $n \geqslant 1$, then for each game $\left\{g^L \mid g^R\right\}$, it holds that $|g^L| \leqslant 2^{\mathrm{g}(n-1)}$ and $|g^R| \leqslant 2^{\mathrm{g}(n-1)}$. That means, there are $\mathrm{g}(n) \leqslant (2^{\mathrm{b}(n-1)})^2$ possible games which is a finite value.

□

To be able to distinguish between games that are equal and those who have complete the same structure, we introduce another term.

Definition 2.24. The game G is **identical** to H, denoted $G \cong H$, when G and H have isomorphic game trees.

Definition 2.25. The **negative** of G, denoted $-G$ is recursively defined by

$$-G := \left\{-G^R \mid -G^L\right\}.$$

We will define a **subtraction** of H from G as

$$G - H := G + (-H).$$

Lemma 2.26 (Idempotence of negative)**.** $-(-G) \cong G$.

Proof. (By induction on birthday).

$$\begin{aligned} -(-G) &\cong -(-\left\{G^L | G^R\right\}) \cong -\left\{-G^R | -G^L\right\} \cong \left\{-(-G^L) | -(-G^R)\right\} \\ &\overset{\text{IH}}{\cong} \left\{G^L | G^R\right\} \cong G. \end{aligned}$$

□

Lemma 2.27 (Distributivity of the opposite game)**.**

$$-(G + H) \cong (-G) + (-H).$$

Proof. (By induction on birthday).

$$\begin{aligned}
-(G+H) &\cong \left\{-(G+H)^R \mid -(G+H)^L\right\} \\
&\cong \left\{-(G^R+H), -(G+H^R) \mid -(G^L+H), -(G+H^L)\right\} \\
&\overset{\text{IH}}{\cong} \left\{-G^R-H, -G-H^R \mid -G^L-H, -G-H^L\right\} \\
&\cong (-G)+(-H).
\end{aligned}$$

□

Lemma 2.28. (Zero is the only lost game)

$$G = 0 \Leftrightarrow G \in \mathcal{P}.$$

Proof. ($\Rightarrow$) Zero is clearly a lost game.

($\Leftarrow$) Let $G \in \mathcal{P}$. We need to show that $o(G+X) = o(X)$ for any game X. But Proposition 2.14 says that adding a lost game will not change the outcome and G is lost, so we are done.

□

Theorem 2.29. The set of game values $\mathbb{G}$ with an operation + *(a disjunctive sum)* forms an Abelian group with 0 (*zero game*) as its identity element.

Proof. We need to show that zero is an *identity*, the sum on games is *commutative* and *associative* and that each game has its additive inverse. Let G, H, J denote arbitrary games.

(a) (identity) $G+0 = G$. In fact $G+0 \cong G$, since adding 0 does not change the structure of G at all: $G+0 \cong \left\{G^L+0 \mid G^R+0\right\} \cong \left\{g^L \mid g^R\right\}$.

(b) (commutativity) $G+H = H+G$: by induction on birthday

$$\begin{aligned}
G+H &\cong \left\{G^L+H, G+H^L \mid G^R+H, G+H^R\right\} \\
&\overset{\text{IH}}{\cong} \left\{H^L+G, H+G^L \mid H^R+G, H+G^R\right\} \\
&\cong H+G.
\end{aligned}$$

(c) (associativity) $G+(H+J) = (G+H)+J$ can be proven by breaking down the sum according to the definition and applying the same induction as above.

(d) (negatives) We need to show that $G-G = 0$. Since $G-G \cong \left\{g^L \mid g^R\right\} + \left\{-g^R \mid -g^L\right\}$, the second player has clearly a winning strategy by mirroring the first player, so $G-G \in \mathcal{P}$. The rest follows by Lemma 2.28.

□

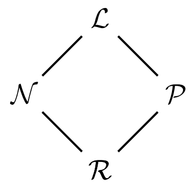


Figure 2.4: A Hasse diagram of the partial ordering of the outcome classes.

2.2.2 Partial-Order Structure

Again in Example 2.15 we have observed that some won game could be in some context favorable to one player more then to the other. Motivated by this observation, we attempt to order the Abelian group of games by the *favorability* towards Left with respect to the *partial ordering of outcome classes*, presented as a Hasse diagram in Figure 2.4.

Definition 2.30. The game G **is greater than** H, denoted

$$G \geqslant H, \text{ if } o(G+X) \geqslant o(H+X) \text{ for every game } X.$$

By the definition of the partial order on the outcome (Figure 2.4), in order to show that $G \geqslant H$ it suffices to show that Left has a winning strategy as a second.

Observation 2.31. $\geqslant$ is a partial order on games.

Proposition 2.32. If $G \geqslant H$ then for any game J is $G + J \geqslant H + J$.

Proof. Identical to the proof of Proposition 2.20. $\square$

Definition 2.33. We say that G is **incomparable** with H, denoted

$$G \parallel H, \text{ if } G - H \in \mathcal{N}.$$

We say that is G is **greater** (**smaller**) than H or **confused** with H, denoted

$$\begin{aligned} G \mathrel{\Vert\!\rhd} H \quad &\text{if} \quad G > H \vee G \parallel H, \text{ and} \\ G \mathrel{\lhd\!\Vert} H \quad &\text{if} \quad G < H \vee G \parallel H, \text{ respectively.} \end{aligned}$$

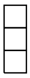
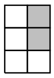
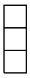
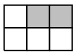
Definition 2.34. We say that G is **strictly greater** (**smaller**) than H, denoted

$$\begin{aligned} G > H \quad &\text{if} \quad G \geqslant H \wedge G \neq H \quad \text{and} \\ G < H \quad &\text{if} \quad G \leqslant H \wedge G \neq H, \text{ respectively.} \end{aligned}$$

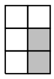
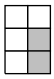
Corollary 2.35. The following table summarizes the terms and properties listed above:

$$\begin{array}{lcllcl} G = 0 & \Leftrightarrow & G \in \mathcal{P} & G = H & \Leftrightarrow & G - H \in \mathcal{P} \\ G > 0 & \Leftrightarrow & G \in \mathcal{L} & G > H & \Leftrightarrow & G - H \in \mathcal{L} \\ G < 0 & \Leftrightarrow & G \in \mathcal{R} & G < H & \Leftrightarrow & G - H \in \mathcal{R} \\ G \parallel 0 & \Leftrightarrow & G \in \mathcal{N} & G \parallel H & \Leftrightarrow & G - H \in \mathcal{N} \end{array}$$

Note that based on this table we can construct an algorithm to determine the relationship between any two games.

Example 2.36. To show that  $>$ , it suffices to show that in the game  $+$  Left wins as a second. But Right has only a single move to the second component, so Left's option in the first component can be his winning move.

2.2.3 Canonical Form

As we have seen in the example of empty games which equals to any lost game, no matter how complicated. Precise analysis will also show that the following positions are equal:  $+$  $=$ . When analyzing some game, it might be useful to replace its components with smaller games that are equal in order to simplify the analysis. It is fortunate that for any game exists a unique *smallest* game that is equal to it. This game is called a *canonical form* a of given game.

Definition 2.37. Let G be a game. Then for typical options of Left (G^{L_1}, G^{L_2}) and Right (G^{R_1}, G^{R_2}):

(a) Left move G^{L_1} is **dominated** (**by** G^{L_2}) if $G^{L_2} \geqslant G^{L_1}$.

(b) Right move G^{R_1} is **dominated** (**by** G^{R_2}) if $G^{R_2} \leqslant G^{R_1}$.

(c) Left move G^{L_1} is **reversible** (**through** $G^{L_1R_1}$) if $G^{L_1R_1} \leqslant G$.

(d) Right move G^{R_1} is **reversible** (**through** $G^{R_1L_1}$) if $G^{R_1L_1} \geqslant G$.

Theorem 2.38 (About dominated moves)**.** Let $G = \{A, B, \ldots \mid H, I, \ldots\}$, $B \geqslant A$. Then $G = \{B, \ldots \mid H, I, \ldots\}$.

Proof. We want to show whether $\{A, B, \ldots \mid H, I, \ldots\} - \{B, \ldots \mid H, I, \ldots\} \in \mathcal{P}$. Consider the following cases:

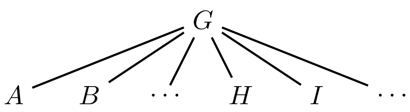


Figure 2.5: Dominated moves.

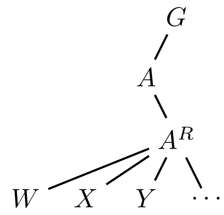


Figure 2.6: Reversible moves.

(a) Left begins into A. Then the second player plays into $-B$ (he can, because of $-\{B, \ldots \mid H, I, \ldots\} = \{-H, -I, \ldots \mid -B, \ldots\}$). This leads to game $A - B$. But because $A \leqslant B$, the second player wins.

(b) The first player moves outside of A. Then the second player can mirror the moves of the first player.

□

Theorem 2.39 (About reversible moves)**.** Let $G = \{A, B, \ldots \mid H, I, \ldots\}$ and let there exist a Right's move A^R such that $G \geqslant A^R$. Denote $A^R = \{X, Y, \ldots \mid \ldots\}$. Now consider a game

$$\overline{G} = \{X, Y, \ldots, B, \ldots \mid H, I, \ldots\}.$$

Then $G = \overline{G}$.

Proof. We need to show that $G - \overline{G} = 0$.

$$G - \overline{G} = \{A, B, \ldots \mid H, I, \ldots\} + \{-H, -I, \ldots \mid -X, -Y, \ldots, -B, \ldots\}.$$

Every opening move can be mirrored except for Left's move to A and Right's move to $-X, -Y, \ldots$. Let us consider these cases:

(a) Let Right move to $-X$ (or $-Y, \ldots$). But X is originally a move in A^R, and since $G \geqslant A^R$, it follows that $G - A^R \geqslant 0$, so Left wins here as a second.

(b) Let Left move to A, we get $A - \overline{G}$. We can show a winning strategy for Right: he moves into A^R, yielding $A^R - \overline{G}$. Now without loss of generality Left can play to:

  (i) $A^R - H$: But $A^R - G \leqslant 0$, so Right wins here as a second.

  (ii) $X - \overline{G}$: Then Right can mirror in $-\overline{G}$ yielding $X - X = 0$.

□

Example 2.40. Consider the following sum $1+(-1)$. We show that it is equal to 0 using Theorem about reversible moves: $1+(-1) \cong \{-1 \mid 1\} \cong \{\mid 0 \parallel 0 \mid\}$. Then, let $A = -1 = \{\mid 0\}$, so $A^R = 0$. But because $1-1-0$ is a lost game, $1-1 \geqslant 0$. So Left's move -1 is reversible through $(-1)^R = 0$. By the previous theorem, we get $\{-1 \mid 1\} = \{\mid 1\}$. Using symmetrical argument, Left's move to 1 reverses through $1^L = 0$, yielding $\{\mid 1\} = \{\mid\} \cong 0$.

Definition 2.41. Game G is in **canonical form**, if G does not have any dominated and reversible moves and all of its subpositions are in canonical form.

Theorem 2.42 (About canonical form)**.** Let G, H be games in canonical form. Then $G = H \Leftrightarrow G \cong H$.

Proof. ($\Leftarrow$) Trivially, if two games are isomorphic, of course they have the same outcome.

($\Rightarrow$) Let $G = H$. This means that Left wins as a second in $G - H$. So he has a reply for any Right's move into $G^R - H$:

Let the reply be $G^{RL} - H$. But that would mean $G^{RL} - H \geqslant 0$ which means $G^{RL} \geqslant H = G$. But then G has a reversible move which is a contradiction with the fact that it is in canonical form.

So we may assume that the reply is in $-H$. This leads to $G^R - H^R \geqslant 0$, so $G^R \geqslant H^R$. So $(\forall G^R \in g^R)(\exists H^R \in h^R) : G^R \geqslant H^R$.

Also, this argument can be done symmetrically for H, so $(\forall H^R \in h^R)$ $(\exists G^{R'} \in g^R) : G^{R'} \geqslant H^R$ We get $G^R \geqslant H^R \geqslant G^{R'}$. But because G does not have dominated moves and $G^R \geqslant G^{R'}$, $G^R \cong G^{R'}$ must hold. However, this means that $G^R \cong H^R$, meaning that for any right move in G, there is an equivalent move in H, so $g^R \cong h^R$.

Because Right also wins second in $G - H$, the same argument can be done symmetrically, so for any Left's move in G there is an identical one in H, so $g^L \cong h^L$. This implies $G \cong H$.

□

Observation 2.43. We can design a *canonization algorithm* based on the proofs of Theorems 2.38 and 2.39, since by Theorem 2.42 the order of reductions does not affect the resulting canonical form of G.

Definition 2.44. We will denote the function $\mathcal{C}$: *games* $\to \mathbb{G}$ as the **canonization function** which maps each combinatorial game to its unique canonical form.

2.2.4 Numbers

In the previous section, we have encountered three numeric games $0, 1$ and -1. We can now see that $1 - 1 = 0$ because the second has a mirroring strategy. In fact, there are many other games that alongside with disjunctive sum behave as integers and some rationals.

Note: To avoid confusion between numbers as games and regular numbers, when it will be necessary, we will put number games inside a box, e.g. $\boxed{0}$, $\boxed{1}$, and $\boxed{-1}$.

Definition 2.45. An **integer game** for integer $n \in \mathbb{N}_0$ is:

$$\boxed{n} := \left\{ \boxed{n-1} \mid \right\}$$
$$\boxed{-n} := -\boxed{n}$$

Definition 2.46. For all $j = 0, 1, \ldots, m \geqslant 0$ we define **dyadic rationals** as follows:

$$\boxed{\frac{m}{2^j}} := \begin{cases} \left\{ \boxed{\frac{m-1}{2^j}} \,\middle|\, \boxed{\frac{m+1}{2^j}} \right\} & \text{for odd } m; \\ \boxed{\frac{m/2}{2^{j-1}}} & \text{for even } m; \\ \boxed{a} + \boxed{\frac{b}{c}} & \text{for } m > 2^j \text{ and } m/2^j = a + b/c \text{ and an} \\ \textit{integer game} & \text{for } m = 0, m = 2^j. \end{cases}$$

We dennote the group of all dyadic rational by $\mathbb{D}$.

Observation 2.47. For all a, b, c be dyadic rational numbers. Then the following holds:

(a) $a + b + c \geqslant 0$ if and only $\boxed{a} + \boxed{b} + \boxed{c} \geqslant \boxed{0}$,

(b) $\boxed{a} + \boxed{b} = \boxed{c} \Leftrightarrow a + b = c$ and

(c) $\boxed{a} \geqslant \boxed{b} \Leftrightarrow a \geqslant b$

Proof of these statements are trivial and are left to the reader.

Definition 2.48. Let x^L, x^R be numbers. Then the **simplest number** between x^L and x^R is such a number x: $x^L < x < x^R$ that has minimal birthday of all numbers in the given interval.

Lemma 2.49 (About birthday of a dyadic game)**.** Let $n \in \mathbb{Z}, n \neq 0$. Then

(a) $b(\boxed{n}) = |n|$.

(b) $b\left(\boxed{n + \dfrac{i}{2^j}}\right) = b\left(-\boxed{n + \dfrac{i}{2^j}}\right) = |n + j + 1|$.

Note: A whole number $\boxed{n'}$ in the case (a) can be converted to dyadic: Let $n = n' - 1$, $i = 1$, $j = 0$. Then $(n-1) + \frac{1}{2^0} = n'$.

Proof. (By induction on sum $n + j$).

(a) $j = 0$: Then $(n-1) + \frac{1}{2^0}$ is an integer with birthday $n + 1$.

(b) $j > 0$: Then:

$$n + \frac{i}{2^j} = n + \left\{ \frac{i-1}{2^j} \;\middle|\; \frac{i+1}{2^j} \right\} = \{n - 1 \mid\} + \left\{ \frac{i-1}{2^j} \;\middle|\; \frac{i+1}{2^j} \right\}$$

$$\overset{\text{IH}}{=} \left\{ (n-1) + \frac{i}{2^j}, n + \frac{i-1}{2^j} \;\middle|\; n + \frac{i+1}{2^j} \right\}.$$

(i) $b((n-1) + \frac{i}{2^j}) \overset{\text{IH}}{=} n - 1 + j + 1 = n + j$

(ii) $n + \frac{i-1}{2^j}$: because i is odd, we get $n + \frac{x}{2^{j'}}$ for some x and $j' < j$. Then $b(n + \frac{i-1}{2^j}) \leqslant n + j$.

(iii) Similarly $b(n + \frac{i+1}{2^j}) \leqslant n + j$.

Then $b(n + \frac{i}{2^j}) = n + j + 1$.

□

Lemma 2.50. Let x_1, x_2 be numbers such that $x_1 < x_2$ and $b(x_1) = b(x_2)$. Then there exists such number x that $x_1 < x < x_2$ and $b(x) < b(x_1) = b(x_2)$.

Proof. Consider the following cases:

(a) Both x_1 and x_2 are integer games. Then $x_1 = \boxed{-n}$, $x_2 = \boxed{n}$ and $x = \boxed{0}$ with $b(x) = 0$.

(b) Either x_1 or x_2 is not an integer game. If $x_1 < 0$ and $x_2 > 0$, $x = 0$ as above. Without loss of generality, suppose that $x_1, x_2 > 0$.

$$\text{Let } x_1 = n_1 + \frac{k_1}{2^{j_1}}, x_2 = n_2 + \frac{k_2}{2^{j_2}}.$$

Without loss of generality, also suppose that x_1 not an integer game. Then by definition of dyadic game we have

$$x_1^R = n_1 + \frac{k_1 + 1}{2^{j_1}} = n_1 + \frac{k_1}{2^{j_1}} + \frac{1}{2^{j_1}}.$$

We claim that $x = x_1^R$. It has definitely smaller birthday than x_1 (it is its right move). Remains to show that $x_1^R < x_2$. Consider the following cases:

(i) $n_2 > n_1$ implies $x_1^R < x_2$ trivially.

(ii) $n_1 = n_2$. Because $b(x_1) = b(x_2)$, by Lemma 2.49 follows that $j_1 = j_2$. But $x_1 < x_2$ so that implies $k_1 < k_2$. Above that, because k_1, k_2 are odd, $k_1 + 2 \leqslant k_2$. This implies $x_1^R < x_2$.

□

Lemma 2.51. The simplest number between any $x_1 < x_2$ is unique.

Proof. (By contradiction). Let there be two simplest numbers s_1, s_2, such that $x_1 < s_1 < s_2 < x_2$. As s_1, s_2 are both the simplest, it follows that $b(s_1) = b(s_2)$. By Lemma 2.50, we know that there is another simpler number between s_1 and s_2 which is a contradiction. □

Theorem 2.52 (The Simplicity Rule)**.** Let G is a game and all its subpositions be numbers, $g^L < g^R$. Then, G is equal to the simplest number between $\mathit{max}(g^L)$ and $\mathit{min}(g^R)$.

Proof. Let $G = \left\{g^L \mid g^R\right\}$. Because all numbers are comparable, it follows from theorem about dominated moves that there are positions G^L, G^R, such that $G = \left\{G^L \mid G^R\right\}$. Let x be the simplest number between G^L and G^R. We show that $G = x$ by showing that $x - G = 0$, a lost game. Consider the following cases:

(a) Right moves first:

  (i) Moves into $x - G^L$. Because $G^L < x$, $x - G^L > 0$ and Left wins.

  (ii) Moves into $x^R - G$ For contradiction, suppose that Right wins. So even when Left moves to $x^R - G^R$, Right wins. This implies that $x^R < G^R$. Also, by definition of a dyadic games $x^R > x > G^L$, so x^R is still between G^L and G^R. Furthermore $b(x^R) < b(x)$, so x^R is simpler than x which is the simplest, a contradiction, so Left wins.

(b) Left moves first: symmetrically, Right wins.

□

Example 2.53.

$$\left\{\frac{1}{2} \,\middle|\, 2\right\} = 1 \qquad \left\{\frac{1}{8} \,\middle|\, \frac{5}{8}\right\} = \frac{1}{2} \qquad \{1 \mid 2\} = \frac{3}{2}$$

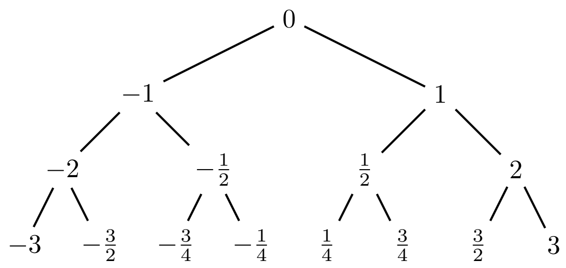


Figure 2.7: An evolutionary tree of dyadic games born by day 3.

Corollary 2.54. By Observation 2.47 and Theorem 2.52 we get an important observation that *all generalized numbers are comparable*, in fact, they form a totally ordered set. Figure 2.7 shows all numbers born by day 3 in a form of the *evolutionary tree*: any subposition of a game in this tree is some ancestor of the corresponding vertex.

2.2.5 Stops and Switches

Let us take a look at the games which are not numbers. So there $(\exists G^L \in g^L)\ (\exists G^R \in g^R): \quad G^L > G^R$. The simplest of them are games having only numeric options:

Definition 2.55. We say that $\{y \mid z\}$ is a **switch** if y, z are numbers and $y > z$.

Theorem 2.56 (Weak number avoidance)**.** Let x be a number, and G be any game that is not equal to a number. Then if L wins as the first player in $G + X$ then L wins by the first move in G.

Proof. We will show the following implication: $G + x^L \geqslant 0 \Rightarrow G^L + x \geqslant 0$. Let $G + x^L \geqslant 0$ and G is not a number. So $G \neq -x$ which implies that $G + x^L > 0$. This means that Left wins as a first in $G + x^L$. By induction on birthday, we get $G^L + x^L \geqslant 0$ for some G^L. But because x is a number, surely $x > x^L$, so the following implication holds $G^L + x^L \geqslant 0 \Rightarrow G^L + x \geqslant 0$. □

We will show that when comparing non-numeric games with numbers, the concept of *stops* which denotes the maximal and minimal numbers that Left and Right players can achieve by playing the game in isolation, is fundamental:

Definition 2.57. The **Left stop** $L(G)$ and the **Right stop** $R(G)$ are defined recursively by

$$L(G) = \begin{cases} G & \text{if } G \text{ is equal to a number;} \\ \max_{G^L}(R(G^L)) & \text{otherwise;} \end{cases}$$

$$R(G) = \begin{cases} G & \text{if } G \text{ is equal to a number;} \\ \min_{G^R}(R(G^R)) & \text{otherwise.} \end{cases}$$

Lemma 2.58. Let G and H be any games. The Left and Right stops have the following useful properties

(a) $L(G) \geqslant R(G)$.

(b) $L(-G) = -R(G)$ and $R(-G) = -L(G)$.

(c) If $L(G) > x$, then $G \mathrel{|\!\triangleright} x$. Likewise, if $L(G) \mathrel{\triangleleft\!|} x$, then $G < x$.

(d) If $R(G) > x$, then $G > x$. Likewise, if $R(G) < x$, then $G < x$.

(e) $R(G) + R(H) \leqslant R(G + H) \leqslant R(G) + L(H)$ and $L(G) + L(H) \geqslant L(G + H) \geqslant R(G) + L(H)$.

Proof. (a) If G is a number then clearly $L(G) = R(G)$. If G is not a number then for contradiction assume that $L(G) < R(G)$. But the simplicity rule (2.52) implies that there is a number X such that $L(G) < X < R(G)$. We will show that the game $G - X \in \mathcal{P}$. By weak number avoidance (2.56) the winning move of the first player would have to be in the component G. But if Left begins, by definition of Left stop he ends up with $L - X < 0$ so he loses. If Right begins, he ends up with $R - X > 0$ so he loses as well. Thus $G - X$ is lost and $G = X$, a contradiction.

(b) Clear from the definition and symmetry of the negation.

(c) Let $R(G) > x$. If G is equal to a number, $G = R(G)$ and we are done. Otherwise, for each G^R we have $L(G^R) \geqslant L(G) > X$, so by induction on (d), $G^R \mathrel{|\!\triangleright} X$ and by the number avoidance, Right will always play into G so $G \geqslant X$. But $G \neq X$, so $G > X$.

(d) Let $L(G) > x$. If G is equal to a number, $G = L(G)$ and we are done. Otherwise, by definition of stops, there must be some G^L such that $R(G^L) = L(G) > x$. By induction on (c) we get $G^L > x$, so $G \mathrel{|\!\triangleright} X$.

(e) Can be found in [71, CG, p. 77]

$\square$

Definition 2.59. The **confusion interval** of G is defined by

$$\operatorname{conf}(X) := \{X \in \mathbb{D} : X \parallel G\}.$$

Lemma 2.60 (Replacement Lemma)**.** Suppose that G' is obtained from G by replacing some Left option G^L with a new option $G^{L'}$. Then

- If $G^{L'} \geqslant G^L$, then $G' \geqslant G$, and
- if $G^{L'} \leqslant G^L$, then $G' \leqslant G$.

Proof. Let $G^{L'} \geqslant G^L$; we will show that Left can win $G' - G$ playing second. If Right moves to $G' - G^L$, Left wins by responding to $G^{L'} - G^L$; if Right makes any other move, then Left has mirroring response. The rest follows by symmetry. □

Theorem 2.61 (Number Translation Theorem, [71, CG, p. 78])**.** Let X is equal to a number and G is not. Then

$$G + X = \left\{G^L + X \mid G^R + X\right\}.$$

2.2.6 Infinitesimals

The total ordering of Numbers hints that sometimes the problem of finding the best move among options might be simply deduced from the canonical forms of these options: if they are totally ordered, Left can always move to the maximal option, while Right can always choose the minimal one and they can be sure they play optimally. In this section, we will see that things are not always so simple. We can see that $\uparrow < 1$. But in a moment we will show that $\uparrow < x$ for any positive number x. So the scale of numbers is insufficient for "measuring" all games.

Definition 2.62. Game G is **infinitesimal** if for all numbers $x > 0$:

$$-x < G < x.$$

Example 2.63. The games $*, \uparrow$ and $\downarrow$ are infinitesimals.

Proof. To give an idea, how to observe these facts, we will prove that the game $\uparrow$ is infinitesimal. We will show that $x - \uparrow > 0$ by induction on birthday of x. Observe that $x - \uparrow = x + \downarrow = \left\{x^L \mid x^R\right\} + \{* \mid 0\}$. Now consider the following courses of this game:

(a) The Right begins.

  (i) Moves to $x + 0 = x > 0$.

$\Downarrow := \downarrow + \downarrow$ $\quad \Uparrow := \uparrow + \uparrow$ $\quad \frac{1}{2} := \{0 \mid *\}$ $\quad \frac{1}{4} := \left\{0 \mid \frac{1}{2}\right\}$ $\quad \frac{3}{2} := \{1 \mid 2\}$

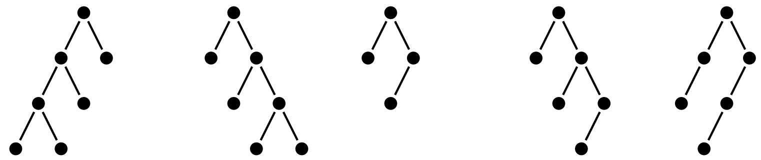


Figure 2.8: Game trees of some infinitesimals and dyadic numbers.

(ii) Moves to $x^R + \downarrow$. But $x^R > x$ and $b(x^R) < b(x)$, so by induction hypothesis, $x^R + \downarrow > 0$.

(b) The Left begins. Then $x > 0 \Rightarrow x^L \geqslant 0$.

(i) If $x^L > 0$, then by induction hypothesis, $x^L + \downarrow > 0$.

(ii) If $x^L = 0$, then we have game $\left\{0 \mid x^R\right\} + \downarrow$. Game $\left\{0 \mid x^R\right\} \in L$ which means $\left\{0 \mid x^R\right\} > 0$. Then $x > * \Rightarrow x + * > 0$.

$\square$

Definition 2.64. We define the games $\Uparrow, \Downarrow, \Uuparrow, \Ddownarrow$, $\Uparrow\!\!\Uparrow$ and $\Downarrow\!\!\Downarrow$ as:

$$\begin{array}{ll}
\Uparrow := \uparrow + \uparrow & \Downarrow := \downarrow + \downarrow \\
\Uuparrow := \uparrow + \uparrow + \uparrow & \Ddownarrow := \downarrow + \downarrow + \downarrow \\
\Uparrow\!\!\Uparrow := \Uparrow + \Uparrow & \Downarrow\!\!\Downarrow := \Downarrow + \Downarrow \\
\uparrow n := \underbrace{\uparrow + \uparrow + \ldots + \uparrow}_{n} & \uparrow n* := \underbrace{\uparrow + \uparrow + \ldots + \uparrow}_{n} + *
\end{array}$$

In Figure 2.8 are drawn a game trees of several infinitesimals and dyadic games for comparison. Notice that $\Uparrow$ and $\frac{1}{4}$ have quite similar game trees, even though one is infinitesimal and second is number. In fact, this similarity holds for any pair $\uparrow n$ and 2^{-n}.

Observation 2.65. Observe the following properties of $\uparrow, \downarrow, \Uparrow$, $\Downarrow$ and $*$:

- $\uparrow \equiv -\downarrow$
- $\downarrow < 0 < \uparrow$, so also $\Downarrow < \downarrow$ and $\Downarrow* < \downarrow*$.
- $\downarrow \parallel * \parallel \uparrow$

  Proof. We will show that $\uparrow + * \in \mathcal{N}$.
  $\uparrow + * \equiv \{0 \mid *\} + \{0 \mid 0\}$. If Left starts, she moves to $\{0 \mid *\}$, Right then to $*$ and Left wins. If Right starts, she moves to $0 + 0 = 0$ and wins. $\square$

- $\Downarrow < * < \Uparrow$

Definition 2.66. Games G and H are **infinitesimally close**, if $G - H$ is infinitesimal. In this case we say that G is **G-ish**. The games infinitesimally close to numbers are called **numberish**.

Definition 2.67. Game $G > 0$ is an **infinitesimal relative** to $H > 0$, if:

$$(\forall n \in \mathbb{Z}): \quad n \cdot G < H.$$

Definition 2.68. We define a **scalar multiplication** as

$$n \cdot G := \begin{cases} 0 & \text{for } n = 0 \\ G + G + \ldots + G & \text{for } n > 0 \\ (-n) \cdot (-G) & \text{for } n < 0 \end{cases}$$

The following proposition shows that even though we are not able to measure the infinitesimals on the scale of numbers, the scale of multiples of ups and downs has the desired granularity.

Proposition 2.69. [71, CG, p. 85]. For any infinitesimal G, there is some integer n such that

$$n \cdot \uparrow > G > n \cdot \downarrow.$$

Proposition 2.70. [36] The game G is infinitesimal if and only if $L(G) = R(G) = 0$.

Proof. ($\Leftarrow$) Let $L(G) = R(G) = 0$. So by Lemma 2.58, for every game X we have $-X \leqslant G \leqslant X$, therefore for numbers as well, so G is infinitesimal.

($\Rightarrow$) By the same Lemma, if $L(G) < 0$ then $L(G) < -x$ for some positive number x, so $G \mathrel{\|\!\!\triangleright} X$. The same argument can be applied to $R(G)$ by symmetry.

□

Corollary 2.71. If G is All-small then it is infinitesimal. Also, if game G is infinitesimal, then all its subpositions are All-small

Note: The all-small positions such as $\uparrow$, $\Downarrow$ often occur in games in disjunctive sums with $*$. It is a standard practice to omit the $+$ when writing such a disjunctive with $*$. For example, instead of $\uparrow + *$ we write $\uparrow*$ and $\downarrow n + * = \downarrow n*$ [53].

2.2.7 Reduced Canonical Form

Even though for each game there is a unique smallest game that is equal to it (its canonical form), sometimes this smallest game is still too complicated and does not help us much with the analysis of the game. One of the sources of these complications are small infinitesimal perturbances that can complicate the canonical form of some game, even though most games will compare to this game the same. More precisely, the infinitesimal perturbances does not affect the final score of the game (its Left and Right stop), but can have an effect on who has the move when that score is reached.

The idea to ignore such small changes is captured by the *reduced canonical form*: it turns out that similarly as for canonical form, for each game G there exists the *simplest* game that is infinitesimally close to G.

Definition 2.72. Let Inf be some infinitesimal. We write

$$\begin{aligned} G &\equiv H \pmod{\mathrm{Inf}} \quad \text{if} \quad G - H = \mathrm{Inf}, \text{ and} \\ G &\geqq H \pmod{\mathrm{Inf}} \quad \text{if} \quad G - H \geqslant \mathrm{Inf}. \end{aligned}$$

We will write simply $G \equiv H$ if we do not care how much infinitesimally close the games are.

Definition 2.73. Let G be a game.

(a) Left move G^{L_1} is Inf-**dominated** (**by** G^{L_2}) if $G^{L_2} \geqq G^{L_1}$.

(b) Right move G^{R_1} is Inf-**dominated** (**by** G^{R_2}) if $G^{R_2} \leqq G^{R_1}$.

(c) Left move G^{L_1} is Inf-**reversible** (**through** $G^{L_1R_1}$) if $G^{L_1R_1} \leqq G$.

(d) Right move G^{R_1} is Inf-**reversible** (**through** $G^{R_1L_1}$) if $G^{R_1L_1} \geqq G$.

Definition 2.74. The game K is in **reduced canonical form** if, for every subposition H of K, H is a number or contains no Inf-dominated options. Furthermore, if H is not infinitesimally close to a number, H also does not contain any Inf-reversible options. The **reduced canonical form** of G is the game K such that K is in reduced canonical form. We write $K = \mathit{rcf}(G)$.

Theorem 2.75. [36]. For each game G the reduced canonical form $\mathit{rcf}(G)$ is uniquely defined.

2.3 Impartial Theory

A game G is impartial if Left and Right have exactly the same moves available from every subposition H of G: if $H = \{h^L \mid h^R\}$ then $h^L = h^R$. Since there is no point in distinguishing Left and Right moves, we will simplify the notation and in this section, any game G will be denoted as a (possibly empty) set of options $G = \{G'_1, \ldots, G'_k\}$ and $G' \in G$ will denote a typical option of both (indistinguishable) players.

All impartial games are all-small, therefore their value always will be an infinitesimal. Also by the symmetry of moves, every impartial game is its own negative. That is, for any impartial game G we have $G + G = 0$.

Figure 2.9: Ruleset of Nim.

position:	Piles of tokens of sizes $a_1, a_2, \dots, a_k$, denoted $\text{Nim}(a_1, a_2, \dots, a_k)$.
ruleset:	Choose single pile and remove any non-zero number of tokens.

Since the players are indistinguishable, no game can belong either to $\mathcal{L}$ or $\mathcal{R}$. So the game tree of impartial game has the following property:

Observation 2.76. Let G be impartial game and Z are all subpositions of G (all states in its game tree). Also, let (A, B) be a disjoint partitioning of Z into two partitions $\mathcal{A}$ and $\mathcal{B}$, such that:

$$(\forall G \in \mathcal{A})(\forall G' \in G) : G' \in \mathcal{B}, \quad \text{and} \quad (\forall H \in \mathcal{B})(\exists H' \in H) : H' \in \mathcal{A}.$$

Then, $\mathcal{A} \subseteq \mathcal{P}$ and $\mathcal{B} \subseteq \mathcal{N}$.

The simplest impartial games are those, which underlying graph is bipartite:

$$(\forall H \in \mathcal{B})(\forall H' \in H) : H' \in \mathcal{A},$$

so we call them the **bipartite games**. The simplest such game is well known under the name She-Loves-Me, She-Loves-Me-Not. In this game a players alternate in taking a single token from a heap of n tokens. The outcome obviously depends only on the parity of n, so the game is bipartite. Notice that in these games there is no strategy, since no matter what the winner does, he cannot lose his winning position.

The impartial games have under normal form a remarkably simple structure. This can be demonstrated, for instance, by the following fact: For general games the number of distinct game values increases quite steeply: $|\mathbb{G}_2| = 22, |\mathbb{G}_3| = 1474$ and $|\mathbb{G}_4| \cong 3 \cdot 10^{12}$, so the canonical theory does not always provide much simplification. On the other hand, the number of distinct *impartial* game values born by the day n is only $n + 1$, so each day only one new impartial game is born.

Observation 2.77. Let G,H be impartial games. Since G=$-G$, we have

$$G = H \text{ if and only if } G + H \in \mathcal{P}.$$

The complete structure theory for impartial games was given already in the 1935 by the *Sprague-Grundy Theorem*: every impartial game is equal some single-pile game of Nim:

Definition 2.78. For $n \in \mathbb{N}_0$ we define a **nimber** as:

$$*n := \begin{cases} \{*0, *1, \dots, *(n-1)\} & \text{for } n \geqslant 1, \\ 0 & \text{for } n = 0. \end{cases}$$

Lemma 2.79. The canonical form of $*n$ is $\{*0, *1, \ldots, *(n-1)\}$.

Proof. (By induction on n).

(a) If $n = 0, 1$, then $*0 = 0, *1 = \{0\}$ are in canonical form.

(b) If $n > 1$, then $*n = \{*0, *1, \ldots, *(n-1)\}$.

We show that for any $i \neq j$, $i, j < n$ holds $*i \parallel *j$ (so there are no dominated or reversible moves).

Without loss of generality, let $i < j$. We will show that a game $*j - *i \parallel 0$: First player moves $*j$ to $*i$, so we get $*i - *i$. Then he can mirror all the opponent's moves and win.

□

The nimbers play a key role in the analysis of impartial games. But they also play a great role in the analysis of partizan games. Santos and Silva [65] have, for instance, shown that all nimbers occur in the game of Konane. The question what nimbers do appear in some particular game has been asked already by Berlekamp in 1996 [55, GONC1, problem 45] and motivates the following term.

Definition 2.80. A (possibly partizan) game has a **nim-dimension** k if it contains a position $*2^{k-1}$ but not $*2^k$. If no such k exists then the nim-dimension is said to be infinite.

Note that if the game has nim-dimension k then for any $j \leqslant 2^k$ it automatically contains a position $*j$. Also, if it does not contain 2^k, it does not contain any higher nimber.

Definition 2.81. For a proper subset of non-negative integers $M, M \subset \mathbb{N}_0$ we define its **mex value** (the **m**inimal **ex**cluded value) as

$$\mathit{mex}(M) := \mathit{min}(\mathbb{N}_0 \backslash M).$$

So it is the smallest non-negative integer missing in the set M.

The importance of nimbers in impartial games will be clear after the next two famous theorems. Let us start by showing that if the options of a game are all nimbers, the game is a nimber as well.

Theorem 2.82 (The Mex Rule)**.** Let $G = \{*l_1, \ldots, *l_k \mid *r_1, \ldots, *r_j\}$ and

$$\mathit{mex}\{l_1, \ldots, l_k\} = \mathit{mex}\{r_1, \ldots, r_k\} = n.$$

Then, $G = *n$.

Proof. We will show that $G - *n = 0$:

(a) First player moves one of the components to some $*k, k < n$. Because $n > k$, $*k$ is a viable move in both G and $*n$, so the second player can move to the other $*k$, yielding $*k + *k = 0$.

(b) First player moves G to some $*k, k > n$ getting $*k - *n$. Then, since $k > n$, the second player can move $*k$ to $*n$, yielding $*n + *n = 0$.

□

Now we will show that this can be easily generalized to any impartial game.

Theorem 2.83 (Sprague-Grundy Theorem)**.** Let G is an impartial game. Then $G = *m$ for some non-negative integer m.

Proof. Let $G = \{G_1, \dots, G_k\}$. Then by induction on birthday, $G = \{*n_1, \dots, *n_k\}$. Let $m = \mathit{mex}(n_1, \dots, n_k)$. Then by the mex rule we have that $G = *m$. □

Definition 2.84. The non-negative integer m in Theorem 2.83 such that $G = *m$ is known as the **nim-value** or the $\mathcal{G}$-value, denoted $\mathcal{G}(G) = m$. The positions

$$\{\mathcal{G}(G') : G' \in G\}$$

are called the **excludents** of G, so the nim-value of G is the mex of its excludents.

We have shown that any impartial game is equal to some pile of Nim. Thus it might be useful to show how to solve and how to play this game.

Definition 2.85. Let $a, b \in \mathbb{N}_0$. We define their **nim-sum** $a \oplus b$ to be:

$$a \oplus b := R(a) \text{ XOR } R(b),$$

where $R(x)$ denotes the binary string representation of x and XOR denotes the bitwise xor, also knowns as the operation "addition without carrying".

Theorem 2.86 (Analysis of Nim)**.**

$$*a_1 + \ldots + *a_k \in \mathcal{P} \text{ if and only if } \bigoplus_{i=1}^{k} a_i = 0.$$

Proof. Let $G = \text{Nim}(a_1, \dots, a_k)$. By Observation 2.76, it suffices to show that: (a) from each $\mathcal{P}$ position each move leads to an $\mathcal{N}$ position and (b) from each $\mathcal{N}$ position there exists a move to an $\mathcal{P}$ position.

(a) Let $\bigoplus_{i=1}^{k} a_i = 0$. We show that every move in G leads to non-zero nim-sum. Without loss of generality let us suppose that in a move some r stones were removed from the pile a_i. Then by property of nim-sum: $(a_1 - r) \oplus a_2 \oplus \dots a_k \neq 0$.

(b) Let $\bigoplus_{i=1}^{k} a_i \neq 0$. We show that there is a move that leads to zero nim-sum. Let $g = a_1 \oplus \dots \oplus a_k, R(g) = (g_j, g_{j-1}, \dots, g_0)_2$ and $g_j = 1$. Then there must exist a pile whose binary representation of stone count has j-th bit turned on. Without loss of generality suppose it is a_1. Consider a move that takes such number of stones from this pile, that after it we have $a_1' = a_1 \oplus g$. Notice that this is a valid move, since $a_1 > (a_1 \oplus g)$, because the leftmost non-zero bit in $(a_1 \oplus g)$ has to be at smaller position than j (leftmost on bit in g is j-th and a_1 switches it in xor to 0). After the move, the nim-sum of all piles is $(g + a_1) + a_2 + \dots + a_k = g + (a_1 + \dots + a_k) = g + g = 0$.

At the end of the game, the nim-sum is equal to 0. It follows from Observation 2.76, that positions with nim-sum equal to 0 are in $\mathcal{P}$, and the remaining games are in $\mathcal{N}$. $\square$

Observation 2.87. The proof of Theorem 2.86 gives us a polynomial winning strategy for Nim.

Theorem 2.88 (Nim-Sum of Impartial Games)**.** Let $k, j \in \mathbb{N}_0$. Then

$$*k + *j = *(k \oplus j).$$

For impartial games G and H then

$$G + H = \mathcal{G}(G) \oplus \mathcal{G}(H).$$

Proof. We can observe, that $*k \equiv \{*0, \dots, *(k-1)\} \cong \mathrm{Nim}(k)$. Then, $*k + *j + *(k \oplus j) = \mathrm{Nim}(k) + \mathrm{Nim}(j) + Nim(k,j) = Nim(k,j,k,j) = 0$, because $k \oplus j \oplus k \oplus j = 0$. The rest follows from Theorem 2.82. $\square$

Observation 2.89. The proof of Theorem 2.86 alongside with Theorem 2.88 gives us a winning strategy for any impartial combinatorial game G. This strategy is polynomial if the calculation of $\mathcal{G}(G)$ is polynomial.

2.4 Algorithmic Combinatorial Game Theory

The *algorithmic combinatorial game theory*[4] studies the computational complexity of solving combinatorial games in order to achieve one of the following:

(a) **Proof of Intractability**. The aim is to rule out the possibility of some ruleset Γ to admit a complete theory. The usual technique is to identify a subclass of Γ-positions that have such a special structure that some question about the game (usually to tell the *outcome*) can be reduced to a known computationally difficult problem. We will see many examples later in this section.

(b) **Efficient Exhaustive Search**. The objective is to design a *game solving algorithm* that will answer questions about games, which are too complex to admit a concise theory that would answer them in a closed form, but are simple enough that the computation of these answers is tractable. Among other objectives also belong constructing efficient algorithms for *computer-assisted proofs* about structure of these games. One of the most interesting examples of this approach is the solution to the well-known game of CHECKERS. In 2007, Schaeffer et al. finished the analysis over 10^{14} of its subpositions (which took over 18 years) with a result that the game is a first-player draw [67].

In a sense, the algorithmic approach complements the algebraic theory. Both approaches analyze the structure of combinatorial games and can provide important insights. The combinatorial approach seeks to show that the structure is *simple* enough to admit a complete theory. The algorithmic approach attempts to show that the structure is *too complex* so that we can prove that no complete theory is possible. But sometimes we believe that the structure should admit a complete theory, but the structure is too deep to comprehend. Then the algorithmic approach might helps us in building the solution by constructing computer-assisted proofs.

The definitions of a game, ruleset and a solution from the beginning of this chapter might be sufficient for the combinatorial analysis. However, for the algorithmic approach, more precise definitions are required. To be able to formulate the question about the *outcome* of a game as a decision problem, we will drop the distinction between Left and Right and will only distinguish player I, which starts the game, and player II, which plays as a second. Stockmeyer and Chandra then define a game as follows [78].

[4] Not to be confused with the *Algorithmic Game Theory* which studies effective algorithms in the formal models of strategic environments between rational decision-makers, arising in the context of economics.

Definition 2.90. [78, ch. 2] A two-person perfect-information game is a triple (P_1, P_2, R) where P_1 and P_2 are sets, $P_1 \cap P_2 = \varnothing$, and $R \subseteq P_1 \times P_2 \cup P_2 \times P_1$.

The sets P_1 and P_2 will represent the positions in which player I and player II have the initiative, respectively. R will denote the set of available moves: if $(a, b) \in R$ for $a \in P_1, b \in P_2$, then player I can move from position a to position b. We will consider only the normal convention, where the player who is unable to move is declared the loser.

If the game is impartial, we have $P_1 = P_2$, so we can represent the game by an acyclic digraph.

Definition 2.91. Let $\Gamma = (P_1, P_2, R)$ be a game and $W_{i-1}(\Gamma) = \varnothing$. Then for any integer $i \geqslant 0$ define

$$\begin{aligned} W_i(\Gamma) = W_{i-1} &\cup \{a \in P_1 : (\exists b \in P_2) : (a, b) \in R \text{ and } b \in W_{i-1}(\Gamma)\}, \\ &\cup \{b \in P_2 : (\forall a \in p_1) : (b, a) \in R \text{ implies } a \in W_{i-1}(\Gamma)\}. \end{aligned}$$

Then the set of **winning positions** is defined as

$$W(\Gamma) = \bigcup_{i \geqslant 0} W_i.$$

Note that this definition slightly differs from the one of Schaefer [66]. He distinguishes between *general games*, which may be played on any of a large class of inputs, and a *specific game*, which is a general game with these inputs provided. If Γ is a general game, he denotes $\mathrm{Inp}(\Gamma)$ as the set of possible inputs to Γ. The winning positions are then defined as $W(\Gamma) = \{A \in \mathrm{Inp}(\Gamma) : \text{ Player I has a winning strategy for } \Gamma \text{ on input } A.\}$.

Let us now formulate the *outcome* problem as a decision problem.

Definition 2.92. Let $\Gamma = (P_1, P_2, R)$ be a ruleset and suppose there exist a finite alphabet Σ such that $P_1, P_2 \subseteq \Sigma^*$. Then the Outcome problem is defined as follows:

	Outcome
Input:	A position $x \in \Sigma^*$.
Output:	True if and only if $x \in W(\Gamma)$.

Then, the most common definition of the *solution* of a game from the algorithmic point of view, is the following:

Definition 2.93. A solution to a ruleset Γ is a polynomial-time algorithm for computing the outcome $o(G)$ for any position G of Γ.

So in order to answer the question "What is the complexity of solving the game Γ?", we need to know how to encode the positions in this game and then answer the question if there is a polynomial algorithm for computing the outcome under this encoding.

Note that one of possible encodings is the game tree, which has a linear algorithm described in Corollary 2.12, so all games are in a sense solved.

Siegel sidesteps the issue of declaring the encoding of positions by putting the measure of input complexity directly into the definition of ruleset [71, CG, p. 36].

Definition 2.94. A **ruleset** is a pair (Γ, N), where Γ represents an infinite set of position and the mapping $N : \Gamma \to \mathbb{N}$ describes the **input complexity** of each position $G \in \Gamma$.

A *solution* of Γ seeks a polynomial algorithm for computing $o(G)$ for all G which runs in polynomial time in $N(G)$.

Notice that by the Definition 2.94 are all games that have bounded size *a priori* solved. For instance, we know that the size of the game tree of a game of Chess is estimated to be around 10^{123} [69], so even though the game is in theoretical sense solved, it is unlikely that we will know the outcome of starting position anytime soon. For this reason we require from the definition 2.94 that Γ represents an *infinite* set of positions.

In order to find out the complexity of Chess, Fraenkel and Lichtenstein extended Chess to a suitable generalization of $n \times n$ Chess which they proved to be EXPTIME-complete [29]. Even though this result is about a different game, it strongly suggests that there is some intrinsic complexity embedded in the rules of Chess.

For completeness, we define the *winning move* problem as follows.

Definition 2.95. Let $\Gamma = (P_1, P_2, R)$ be a ruleset, $P_1, P_2 \subseteq \Sigma^*$. Then the Winning Move problem is defined as follows:

	Winning Move
Input:	A position $x \in \Sigma^*$ such that $x \in W(\Gamma)$.
Output:	A position $y \in \Sigma^*$ such that $(x, y) \in R$ and $y \notin W(\Gamma)$.

Notice that while Outcome is a decision problem, Winning Move is a search problem. In some cases, the search problem clearly reduces to the decision problem, e.g. when the number of moves is small.

Guo and Miller [38] describe a new way of looking at solving games. They consider a solution for a game a *data structure* that can answer the Winning Move problem to any position that will arise during the play as a response of the other player to a previous winning move, until the end of the game (it serves as a flawless advisor, or an algorithm that is able to play the game itself).

Definition 2.96. Let $\Gamma = (P_1, P_2, R)$ be a ruleset, $P_1, P_2 \subseteq \Sigma^*$. Then the Winning Move Data Structure problem is defined as follows:

Winning Move Data Structure

Input: A position $x \in \Sigma^*$ such that $x \in W(\Gamma)$.

Output: A data structure that efficiently answers the following *query*: Given a position x' such that there exists a sequence of moves $x = x_1, y_1, x_2, y_2, \ldots, x_k = x'$ with $k \geqslant 1$ such that for all i, y_i is the answer of this structure to the query x_i and $(x_i, y_i) \in R$, efficiently calculate a winning move from x'.

2.4.1 Complexity of Some Combinatorial Games

Combinatorial games usually tend to be complete in complexity classes higher than NP, typically they are complete in PSPACE, EXPTIME or even EXPSPACE [23]. This fact should be not surprising: Many classical decision problems that can be formulated as "Does there exist a solution that satisfies ...?" are NP-complete. On the other hand, the question if player I has a winning strategy takes the form: "Does there exist a move of player I such that for every move of player II there exists a move of player I such that, ..., there exists a winning move of player I?" The problems with an unbounded number of alternating quantifiers are much harder.

This can be demonstrated by the notion of non-deterministic computation described in [8]: In classical non-deterministic Turing machines there are only existential quantifiers. In a generalization of nondeterminism called the *alternation*, the *Alternating Turing Machine* can alternate the quantifiers in the course of the computation. It can be shown that the alternating polynomial time is equivalent to the deterministic polynomial space, however, it is widely believed that NP $\neq$ PSPACE.

Stockmeyer and Chandra identify three different types of games based on the resource bounds for games and the length of their plays [78]. These types correspond to three levels of complexity: P, PSPACE and EXPTIME.

Definition 2.97. Let $G \in \Gamma$. Then **reach** of G, denoted $\mathit{reach}(G)$, is defined as the number of distinct positions that can be reached from G. For the ruleset Γ and a non-negative integer S let us define $\mathit{reach}(\Gamma, S)$ be the maximum of $\mathit{reach}(G)$ across all $G \in \Gamma$ such that $|g| = S$.

Then most common rulesets Γ belong to one of the following categories:

I. The $\mathit{reach}(\Gamma, S)$ is bounded by a polynomial in S. The game-tree algorithm solves these games in polynomial time, so these games are in P.

II. For each $G \in \Gamma$ such that $|G| = S$, the length of play of G is bounded by a polynomial in S, but $reach(\Gamma, S)$ can be exponential in S. The recursive evaluation of the game-tree algorithm can be designed so that it uses only polynomial space, so these games are in PSPACE.

III. Both the length of play and the reach can be exponential. The game-tree algorithm runs in exponential time, so these games are in EXPTIME.

The games of the first category can be described as games on graph where a single marker is pushed from node to node. Each position in this game is determined by the position of the vertex, so the reach is polynomial. Here belong for example the single-pile SUBTRACTION GAMES.

Into the second category belong games on graph where players permanently mark the vertices. A position is determined by a subset of vertices, so the reach is exponential, but the length is polynomial. A typical game of this type is, for instance, the game HEX.

Some games in the third category can be described as games on graph where the position of markers can change through the play. Since positions can not repeat, the reach is still exponential, but the length can be exponential as well. Examples of this type are GO and CHESS.

Here we list several complexity classes in which a complete combinatorial game is known. When choosing an example game for a given class, we focused on the simplicity of their rulesets.

NP-complete game. Fraenkel and Yesha described an impartial game called ANNIHILATION which is played on digraph $G = (V, E)$ [31, ch. 2]. Tokens of several types are placed on distinct vertices and each edge has specified a set of token types that can be passed over it. A move consists of choosing some token and moving it over some edge that allows it. If any two tokens appear on a single vertex, they get annihilated and are removed from the game.

PSPACE-complete game. Schaefer [66, ch. 1] explains the significance of the most common completeness in games as follows. The PSPACE-completeness of some games rests ultimately on the fact that the language of these positions is rich enough to describe Turing machine computations. This language is also succinct enough that any problem that is decidable by a polynomial-tape bounded Turing machine is reducible efficiently to instances of these games.

The best-known PSPACE-complete game is probably the game of NODE KAYLES. This impartial game is played on a graph and players in their move choose vertices to remove from the graph, together with their immediate neighbors. The black and white variant of this game, where players can choose only vertices of their colors, is PSPACE-complete as well. [66, ch. 1]

EXPTIME-complete game. Sockmeyer and Chandra show in [78] several games played on propositional formulas that are EXPTIME-complete. One of them is the game G_4 played on a formula $F(X, Y)$ in 13 DNF, where X and Y are disjoint sets of variables. Each variable is assigned some starting value. Then player I (II) moves by changing the value assigned to exactly one variable in $X(Y)$, passing is allowed. A player wins if his moves causes the formula F to become true. The same game can be played under condition when only I player wins if F ever becomes true (G_6), where F is any formula in CNF.

EXPSPACE-complete game. In order to find a game with complexity greater than exponential time, the rules of the type III games need to be changed so that the outcome of a position depends on the way in which the position arose. This can be in some unimportant ways seen in Chess (rules of castling, *en passant* move, draw rule) but this does not affect the EXPTIME-completeness of $n \times n$ Chess.

Robson describes a game on propositional formula, which is complete in exponential space. The game is played on a boolean formula $F(X, Y, \{t\})$ where $X, Y, \{t\}$ is a partition of variables that appear in F. Player I (II) moves by assigning $t = 1$ ($t = 0$) and any values to the variables X (Y). If a formula is false after player's move, he loses. The EXPTIME-completeness is achieved by an additional "no repetition rule" which states that if player assigns his set the same values as some earlier move, provided that at least one variable's assignment has been changed since this move.

Undecidable game. There are few examples of a game for which the Outcome problem is undecidable. One of them is a game played on d ordered heaps of tokens. The position can be represented as a non-negative vector of length d. Moves are described by a finite list M of (possibly negative) integer vectors of length d, player can from the current position subatract any vector from M provided that each heap size will stay non-negative. Larsson and Wästlund showed that it is undecidable to tell if the outcome of two such games equal for all heap sizes [51].

Constant time solvable game. We can simply design a game that is always won by the first player (for instance the single-pile Nim). A more interesting game where such condition occurs only for the starting position, is the game Chomp. Two players take turns in eating a square from a rectangular chocolate bar of dimensions $n \times m$. Each move they choose a single square and alongside with it they have to eat all the squares that are below it and to its right. The top left square is poisoned – who will eat this, dies and looses.

The starting position of this game is a first-player win by a famous "strategy-stealing" argument: Consider two cases: if removing the single square (n, m) is a winning move, we are done. Otherwise, for contradiction, suppose the first player opens by this move and th e second player answers with some winning move to G'. But observe that the first player also has a move to G' from the starting position, so he could have won by moving there initially, a contradiction.

The proof of the existence of a first player's winning strategy for the game of Chomp has an interesting feature: it gives us absolutely no insight into how to play the game. Also, it tells the outcome only for the starting position, while for other positions is the outcome still an open problem. We call such games **weakly solved**.

Fenner and Rogers list more examples of combinatorial games that are complete in a few other complexity classes [16].

2.4.2 Tractability of Combinatorial Games

In the previous section, we have defined the solution of a game as an algorithm that efficiently finds the outcome of all its positions. This definition is the most widely used. However, if we would actually play the game, say with an *oracle* that answers the Outcome problem for any position, it would not always suffice to play the game efficiently.

Let us demonstrate this issue on an example of already mentioned game Chomp. The input complexity of the starting position of this game is $\log(n+m)$. We informally say that such input is **succinct** because it is logarithmic in the size of the object that it represents. The outcome oracle tells us that our position is a first-player win, so there should exist a winning move. Even if the oracle would be able to tell us the outcome of any of our moves in constant time, there are $n \cdot m$ of them which is exponential in the size of the input, thus inefficient. Another issue is that the size of the description increases through the play. Should that affect the notion of tractability of the game?

The main issue with this reasoning is in the discrepancy of concepts of using algorithms to understand the structure of games compared to the idea of algorithms actually playing games. Already in 1957, Rabin noticed that the inherent finiteness of algorithms imposes limitations on the game playing abilities of a machine [62]. Rabin has designed a game that demonstrates an interesting paradox: we know the outcome, but for the winner, it is *undecidable* to compute his winning move. The game is known as Rabin's Game and is played as follows: Let S be a simple set of integers. Player I chooses an integer a. Then player II fixes an integer b. Player I wins if $a + b \in \mathcal{S}$. Simple set is defined as follows: S is recursively enumerable and its complement is infinite and does not contain any infinite recursively enumerable subsets (it is *immune*). Jones has generalized these games into so called *arithmetic games* [46].

Fraenkel attempts to answer what actually is a *tractable* game in the articles [26, 23, 27, 25]. He notices that in contrast with the classical definition of tractability for existential questions, where we can usually linearly order the problems from polynomial to exponential, the situation is far more complex for games. He asks the following rhetorical question while putting in contrast Rabin's Game and Nimania:

> "Which is more tractable: a game that ends after four moves, but it is undecidable who wins, or a game that takes an Ackermann number of moves to finish but the winner can play randomly, having to pay attention only near the end?"

He proposes the following hierarchy of tractability and complexity of games through the definition of tractable, polynomial and efficient strategies:

Definition 2.98. Let $\mathcal{A}$ be a set of combinatorial games (rulesets), Γ be a ruleset in $\mathcal{A}$ and G be a position in Γ. We say that a $\mathcal{A}$ has a **tractable strategy** if it has the following properties:

(a) The *outcome* of G can be computed in polynomial time in $N(G)$.

(b) The *next optimal move* can be computed in polynomial time in $N(G)$.

(c) The *length of play* is at most exponential in $N(G)$.

Furthermore, we say that the set $\mathcal{A}$ has a **polynomial strategy** if the properties (a)–(c) are *closed under sum*: for any two positions G, H in any two rulesets Γ, Λ in $\mathcal{A}$ they do hold also for the disjoint sum $G + H$.

We say that the set $\mathcal{A}$ has an *efficient strategy* if the properties (a)–(d) hold also for the **misère games**.

Chapter **3**

Taking and Breaking Games

The Bouton's analysis of Nim from 1901 motivated many researches to study games from the mathematical perspective. Since then, many Nim variants have been devised, which can be grouped under the class of *Heap games*. The most prominent category among them is *Taking and Breaking games*, which is, after Nim, probably the earliest and most studied class of combinatorial games [1, LIP, ch. 7].

Although most of these games are originally designed to be played under impartial setting, many partizan generalizations have been considered. So we will define them under general context.

3.1 Heap Games

Any **heap game** Γ is played on a finite number of heaps of tokens[5], where each position can be expressed as a disjunctive sum of its **single-heap** components, denoted

$$\mathcal{H} = \{H_0, H_1, H_2, \ldots\}, \text{ where } H_i \text{ denotes heap of } i \text{ tokens.}$$

A typical position in a heap game G has the following form

$$G = H_{a_1} + H_{a_2} + \ldots + H_{a_k}, \text{ with } a_1 < a_2 < \ldots.$$

Each of these components $H_i \in \mathcal{H}$ has to follow the ruleset of the game Γ so we will call them **Γ-heaps** and we will call Γ a **heap ruleset**. In their move, a player can choose a single-heap and replace it with a (possibly empty) set of heaps given by the specific ruleset of the game. So a typical move on a *single-heap* will have the form

$$H_n' = H_{a_1} + H_{a_2} + \ldots + H_{a_k} \quad (\text{where each } a_i < n).$$

[5] Also called *piles of beans*, *piles of coins*, *heaps of chips* or *rows of counters*.

Figure 3.1: Ruleset of HEAP GAMES.

position:	Piles of tokens of sizes $a_1, a_2, \ldots, a_k$, denoted $H^{\prime}_{a_1}+H_{a_2}+\ldots+H_{a_k}$.
ruleset:	Choose some heap, say H_{a_1}, and replace it with heaps $H_{b_1}, H_{b_2}, \ldots H_{b_{k'}}$ with $b_1 + \ldots + b_{k'} \leqslant a_1$ given by the specific ruleset of the heap game.

Figure 3.2: Ruleset of a NIMANIA.

position:	Heaps $H_{a_1} + H_{a_2} + \ldots + H_{a_k}$ and the counter k indicating the number of the current move (starting at $k = 1$).
ruleset:	Choose some heap, say H_{a_1}, and replace it with k copies of H_{a_1-1}; increment k.

Even stronger restriction on such move (which follow most of such games) is that the *total* size of newly created heaps must be smaller than the size of the heap component moved into, so we cannot introduce new tokens into the game. When this restriction is followed, it is clear that these games are short. An example of a famous game that does not follow this restriction is in Figure 3.2.

If we do not put any restriction on the ruleset Γ, the above definition of a heap game actually allows us to construct any possible impartial game. Clearly, given an acyclic digraph $G = (V, E)$ and a topological order of vertices $v_1, v_2, \ldots$, we can define an equivalent heap game as

$$H_n = \{H_i : (n, i) \in E\}.$$

The following restriction is essential to push the analysis of heap games further:

Definition 3.1. We say that the heap game Γ is **invariant** for all $j \geqslant 0$ and $a_1, \ldots, a_k$ such that $1 \leqslant a_1 + \ldots + a_k \leqslant n$ and $a_i \geqslant 1$ for all i:

$$\begin{gathered} H_{a_1} + H_{a_2} + \ldots + H_{a_k} \text{ is an option of } H_n \\ \text{if and only if} \\ H_{a_1} + H_{a_2} + \ldots + H_{a_k+j} \text{ is an option of } H_{n+j}. \end{gathered}$$

3.1.1 Heap Games and Sequences

Definition 3.2. We will call the set of all single-heap positions

$$\mathcal{H} = \{H_0, H_1, H_2, \ldots\}$$

the **heap alphabet**.

Definition 3.3. Let $\mathcal{A}$ the set of all positions of some heap game Γ. We say that $\mathcal{A}$ is the **heap algebra** on the heap alphabet $\mathcal{H}$. Each position $\mathcal{G} \in \mathcal{A}$ can be represented as a finite multiset of elements from $\mathcal{H}$.

Let Γ be a heap ruleset and Φ_Γ a *game mapping function* on the heap alphabet $\mathcal{A}$ (see Definition 2.17), $\Phi : H_n \to \mathcal{Q}$. We will write $\Phi(n) = \Phi(H_n)$ for $n \in \mathbb{N}_0$. It is useful to analyze the values $\Phi(n)$ as a sequence enumerated by the heap size n.

Definition 3.4. Let Φ be a game mapping function on a heap alphabet and Γ a heap ruleset. We define the Φ-sequence(Γ) as

$$\Phi\text{-sequence}(\Gamma) := \big(\Phi_\Gamma(H_n)\big)_{n=0}^{\infty} = \big(\Phi_\Gamma(n)\big)_{n=0}^{\infty}.$$

Note: We will sometimes slightly abuse the notation of the Φ-sequences in order to formulate them more succinctly. For instance, consider the following Φ-sequence:

$$\big(\Phi_\Gamma(n)\big)_{n=1}^{\infty} = 0, 1, 2, 0, 1, 2, 0, 1, 0, 1, 0, 1, \ldots.$$

Since it has all nimbers smaller than 10, we will omit the commas, introduce spaces to separate logical blocks and use the so-called overbar notation for periodicity. Together with the above notation for Φ-sequences, we write

$$\Phi\text{-sequence}\big(\Gamma\big) = 012\ 012\ \overline{01}.$$

For more complicated sequences we will also use the superscript to denote repetition, e.g. $(012)^3 4$ denotes the sequence 012 012 012 4.

3.1.2 Periodicity of Φ-sequences

One very important notion while solving heap games is **periodicity**. Many heap games have Φ-sequences which exhibit an interesting behavior: after some irregularities at the beginning of the sequence, all values $\Phi(n)$ are structured in blocks of fixed size p. For these blocks there exists a function $f : \mathbb{N}_0 \to \mathbb{N}_0$ such that each block is a function of the previous one. Let us define this property more precisely.

Definition 3.5. Let n_0, p be integers with $n_0 \geqslant 0, p > 0$. We say that a Φ-sequence is ***f*-periodic** with **period** p and **pre-period** n_0, if for all integers n such that $n \geqslant n_0$ we have

$$\Phi(n + p) = f(\Phi(n)).$$

We are usually interested in minimal such n_0 and p. We will often interchange between the terms period and period length or pre-period and pre-period length for sake of brevity. By saying a sequence has period $\overline{012}$ we mean that after some irregular values, 012 repeat indefinitely, and by saying the sequence has period 3, we mean the length of the (usually shortest) repeated block.

Furthermore if $n_0 = 0$, we say that the sequence is **strictly**[6] f-periodic, otherwise it is **ultimately** f-periodic.

By representing a heap game Γ as a periodic Φ-sequence, we can compute the value of any position efficiently thereby solving given heap game.

Simple periodicity or just periodicity denotes the Φ-sequences which are f-periodic having f an identity function $f(g) := g$. When we say a Φ-sequence is **periodic**, we will mean it meets the criteria a simple periodicity. All finite subtraction games exhibit this type of periodicity. It is conjectured that all octal games are simple periodic as well. We will denote the simple periodicity with the following overbar notation: 123 $\overline{012}$ denotes the infinite sequence (0123 012 012 012 0...) with period 3 and pre-period 4.

Arithmetic periodicity We say that Φequence is **arithmetic periodic** with **saltus** $s \in \mathcal{A}$ if it is f-periodic with $f(g) := g \cdot s$. All *all-but* subtraction games and many hexadecimal games are arithmetic periodic. Also, Lasker's Nim ($4{\cdot}\overline{3}$) is arithmetic periodic with $p = 4$ and $s = 4$. Note that the saltus $s = 4$ here denotes the *nimber* $*4$, but since this game is impartial, we use commutative monoid of $\mathcal{G}$ values. However, this type of periodicity is not limited to impartial games. For instance, a particular game of Partizan Splittles, introduced in [54], is arithmetic periodic with a *switch* saltus $s = \{a \mid 0\}$.

We will denote the arithmetic periodicity by the same notation as the simple periodicity, appended with a saltus in a parenthesis. If the sequence has pre-period $a_1 a_2 \ldots a_{n_0-1}$ and period $a_{n_0} \ldots a_{n_0+p-1}$ such that for all $n \geqslant n_0$ is $a_{n+p} = a_n \cdot s$, we will write the sequence as

$$a_1 a_2 \ldots a_{n_0-1}\ \overline{a_{n_0} \ldots a_{n_0+p-1}} \quad (+s).$$

Split arithmetic periodicity, also called *periodic regularity* or *sapp regularity* for short, is a periodicity which regularly switches between simple and arithmetic periodicity. More precisely a Φ-sequence is sapp regular with saltus $s \in \mathcal{Q}$ if there is a set $S \subseteq \{0, 1, \ldots, p-1\}$ such that the sequence is f-periodic with

$$f(g) := \begin{cases} g & \text{if } (n \bmod p) \in S \text{ and} \\ g \cdot s & \text{otherwise.} \end{cases}$$

Some octal games with a pass exhibit this type of periodicity.

[6] Also called *purely periodic.*

3.1.3 Impartial Heap Games

in Section 2.3 we have seen that for the class of impartial games of impartial games, things get much simpler.

Since for any impartial game G we have $G + G = 0$, we can denote each single-heap position H_n as a set of its options, so a *subset* of the heap alphabet $H_n \subseteq \mathcal{H}$.

Furthermore, the Sprague-Grundy theorem (2.83) tells us, that the desired game mapping function on any heap algebra can be formulated using only the values of game mapping function on the heap alphabet.

For any element of the heap algebra $G \in \mathcal{A}$, denoted

$$G = H_{a_1} + H_{a_2} + \ldots + H_{a_k},$$

we get from the nim-sum property of the sum of impartial games (Theorem 2.88) that

$$\mathcal{G}(G) = \mathcal{G}_\Gamma(H_{a_1}) \oplus \mathcal{G}_\Gamma(H_{a_2}) \oplus \ldots \oplus \mathcal{G}_\Gamma(H_{a_k}).$$

Furthermore, the $\mathcal{G}$-value for each Γ-heap H_a can be obtained using the mex rule (Theorem 2.82):

$$\mathcal{G}_\Gamma(H_a) = \mathit{mex}\left\{\mathcal{G}_\Gamma(H_i) : H_i \in H_a\right\}.$$

Therefore the solution to any impartial heap game Γ can be expressed in terms of the function

$$n \mapsto \mathcal{G}_\Gamma(H_n),$$

and the $\mathcal{G}$-sequence(Γ) is a sequence of non-negative integers. The properties of these sequences are essential in the understanding of these games. We will call this sequence the **nim-sequence** of the heap ruleset Γ.

For brevity, let us denote the $\mathcal{G}$-value of a single Γ-heap $\mathcal{G}_\Gamma(H_n)$ as $\mathcal{G}_\Gamma(n)$. If the ruleset will be clear from the context, we will write only $\mathcal{G}(n)$.

An essential property of impartial games is its set of $\mathcal{P}$-positions, since detecting a $\mathcal{P}$ position solves the OUTCOME problem for single-heap positions. The following definition narrows this term for the heap games.

Definition 3.6. The **kernel** of a heap game Γ is the set of sizes of all lost Γ-heaps. Formally,

$$\text{kernel}(\Gamma) = \{n : H_n \in \mathcal{P}\}.$$

In this text the main focus will be on impartial games. Further not, if it will not be stated otherwise, you can always assume the setting of the impartial games.

3.1.4 Partizan Heap Games

We will denote a **partizan heap game** as

$$H_n = \{H_n^L \mid H_n^R\},$$

where H_n^L (H_n^R) range over all options of H_n for Left (Right) player.

The most common partizan heap games are created using impartial rulesets. Given a impartial rulesets Γ and Λ, we define

- **Duel Heap Game**, denoted Γ **versus** Λ, is the partizan heap game $H_n = \{H_n^L \mid H_n^R\}$, where H_n^L range over the Γ-heap $H_n^L \in H_n^\Gamma$ and H_n^R range over the Λ-heap $H_n^R \in H_n^\Lambda$.

  These type of games were introduced by Fraenkel and Kotzig in 1987 [28] and applied to some more taking and breaking games by Mesdal in 2009 [54].

- **Black and White Heap Hame** with (an impartial) ruleset Γ and partition of integers $(\mathscr{B}, \mathscr{W})$ such that $\mathscr{B} \cup \mathscr{W} = \mathbb{N}_0, \mathscr{B} \cap \mathscr{W} = \varnothing$. Let us denote H_n^Γ the impartial Γ-heap. Then its black and white game is the partizan heap game $H_n = \{H_n^L \mid H_n^R\}$, where for the option $H_n^L = H_{a_1} + H_{a_2} + \ldots + H_{a_1}$ range over $H_n^L \in H_n^\Gamma$ with $a_i \in \mathscr{B}$ for all i, and for the option $H_n^R = H_{b_1} + H_{b_2} + \ldots + H_{b_1}$ range over $H_n^R \in H_n^\Gamma$ with $b_j \in \mathscr{W}$ for all j.

  This variant of black and white games was first introduced by Schaefer, who has in 1977 showed that the black and white Node Kayles are PSPACE-complete [66]. Other games in the black and white variant were studied by Fenner et al. [17].

Similarly as for the impartial games, we will be usually interested in the properties of the canonical sequences $\mathcal{C}$-sequence(Γ **vs** Λ) and $\mathcal{C}$-sequence(Γ) and the outcome sequences of these games. Note that in order to be able to determine the outcome of a sum $H_a + H_b$ for some $a, b \in \mathbb{N}_0$, the $\mathcal{C}$-sequence must consist only of some very limited class of game values (defined by the monoid Q of this game-mapping function for a specific game). Sometimes we will not be able to determine the outcome of any sum, however the $\mathcal{C}$-sequence might be still interesting for the analysis of single-pile games. Thus we will not always require for $\mathcal{C}$ to be a proper game-mapping function.

Fraenkel and Kotzig defined another interesting property that can be analyzed for the duel heap games.

Definition 3.7. An impartial heap game Γ **dominates** an impartial heap game Λ (denoted $\Gamma \prec \Lambda$), if there exists a non-negative integer n_0 such that every heap of size $n \geqslant n_0$ has the property that L can win both as a first and as a second player in the *duel heap game* Γ **versus** Λ.

Figure 3.3: Ruleset of Grundy's Game.

position:	Heaps $H_1 + H_2 + \ldots + H_k$.
ruleset:	Choose a single heap and split it into two *non-empty* heaps.

Analogously we define $\Gamma \succ \Lambda$ if every heap of size $n \geqslant n_0$ has the property that R can win both as a first and as a second player. If $\Gamma \nprec \Lambda$ and $\Gamma \nsucc \Lambda$, we say that Γ and Λ are **incomparable** (denoted $\Gamma \parallel \Lambda$).

3.2 Taking and Breaking Games

Taking and Breaking Games are heap games, such that the move in these games can be described as follows:

(1) Chosing a heap to move into.

(2) **Taking** some tokens from this heap.

(3) **Breaking** the remainder into several heaps.

The specific rules usually put restrictions on the *taking* and *breaking* parts of the move. Sometimes the taking is not allowed at all. These are Breaking Games and most famous among them is probably the Grundy's Game described in Figure 3.3. Here the *taking* is not allowed and the *breaking* can be done only into two heaps of unequal size. The analysis of this game, introduced by Grundy in 1939, is still open [42, p. 111].

When the breaking move is not allowed, we call the game a Take-Away Game. A famous example of a take-away game is the Silver Dollar Game. It is played like Nim with one exception, that the sizes of heaps must stay unique all the time. This complicates, for instance, the analysis of two-pile game, since the strategy for two-pile nim is based on this move.

3.2.1 Code-Digit Games

In the famous paper from 1956, Guy and Smith [39, ch. 3] devised a code classifying a broad range of taking and breaking games:

Definition 3.8. The rules of a **code-digit game** D are described as a sequence

$$\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots, \quad \text{with } d_i \geqslant 0 \text{ for all } i,$$

and are defined as follows: if the binary expansion of a *code digit* d_k is:

$$\mathsf{d}_k = 2^{a_k} + 2^{b_k} + 2^{c_k} + \ldots, \qquad \text{with } 0 \leqslant a_k \leqslant b_k \leqslant c_k \leqslant \ldots,$$

then in the game D player can remove k tokens from any heap, provided that the rest of the heap is divided into exactly a_k or b_k or c_k or ... *non-empty* heaps.

Theorem 3.9. Code-digit games are invariant.

Proof. Let i be the number of tokens removed in a move into

$$H_{a_1} + H_{a_2} + \ldots + H_{a_k} \text{ is an option of } H_n,$$

where all a_i are positive integers. Since we could do this move, this implies that 2^k must be a part of the binary expansion of the digit d_i. Now consider some position H_{n+j}. Observe that by using the same digit d_i, we can achieve the move into

$$H_{a_1} + H_{a_2} + \ldots + H_{a_k+j} \text{ is an option of } H_{n+j}.$$

Clearly, this is an equivalence, so the game must be invariant. □

Definition 3.10. The **length** of a code-digit game $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ is the largest k such that $\mathsf{d}_k \neq 0$ or ∞ if no such k exists.

Although the definition of the class of code-digit games might look quite restrictive, it actually contains a lot of games that have been studied separately, as Nim ($\cdot\overline{3}$), Subtraction Games ($\cdot\mathsf{s}_1\mathsf{s}_2\mathsf{s}_3\ldots$ with $\mathsf{s}_i = 3$ if i is in subtraction set, zero otherwise), Kayles ($\cdot 77$), Dawson's Kayles ($\cdot 07$), Dawson's Chess ($\cdot 137$), Guiles ($\cdot 15$), Treblecross ($\cdot 007$), Officers ($\cdot 6$), Lasker's Nim ($4{\cdot}\overline{3}$) and many others.

3.2.2 Equivalences Among Code-Digit Games

One may notice that the notation of code digit games allows us to describe some games that are the same using two different code digits. For example consider a game $\cdot 4$, where the only move is to remove single token and partition the rest into two non-empty piles. Then the game $\cdot 42$ will be the same, because any move that is done using $\mathsf{d}_2 = 2$ that takes 2 tokens from pile of size greater than 2 can be modeled using a move $\mathsf{d}_1 = 4$ by taking single token and splitting the remainder into sizes 1 and the rest. Since $\mathcal{G}(1) = 0$ for this game, the excludents will be the same. Guy noticed these equivalences and described them in [39, ch. 4] without proof. We will provide the proofs of these equivalences in Chapter 4.

Definition 3.11. Code-digit games A and B are **equivalent**, denoted $\mathsf{A} \equiv \mathsf{B}$ if $\mathcal{G}_{\mathsf{A}}(n) = \mathcal{G}_{\mathsf{B}}(n)$ for all n.

Definition 3.12. Code-digit game B is ***t*-th cousin** of game A, denoted $\mathsf{A} \equiv_t \mathsf{B}$ if $\mathcal{G}_{\mathsf{A}}(n) = \mathcal{G}_{\mathsf{B}}(n+t)$ for all n and $\mathcal{G}_{\mathsf{B}}(n) = 0$ for $n < t$.

Definition 3.13. We say that code digit d_j **contains** a code digit d_k, denoted $\mathsf{d}_k \in \mathsf{d}_j$, if $\mathsf{d}_j \,\&\, \mathsf{d}_k$ equals d_k, where & represents binary "and" operation. In other words, the allowed divisions into heaps of d_k is a subset of allowed divisions of d_j. (For instance, $\mathsf{3}$ contains code digits $\mathsf{0}, \mathsf{1}, \mathsf{2}$ and $\mathsf{3}$).

Each value of the form 2^i contained in some digit of code-digit game D is called a **digit-bit**.

Proposition 3.14 (Code-digit equivalence, [39, ch. 4])**.** Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ be code digit game with d_1 being even. Now suppose that d_k contains $\mathsf{2}^\ell$ for some $k \geqslant 1$ and $\ell > 0$. Then it makes no difference to the game if code-digit d_{k+1} contains $\mathsf{2}^{j-1}$ or not.

Example 3.15. $\mathsf{4{\cdot}4} \equiv \mathsf{4{\cdot}42} \equiv \mathsf{4{\cdot}421} \equiv \mathsf{4{\cdot}6} \equiv \mathsf{4{\cdot}62} \equiv \mathsf{4{\cdot}621} \equiv \mathsf{4{\cdot}61} \equiv \mathsf{4{\cdot}63} \equiv \mathsf{4{\cdot}631}$.

Proposition 3.16 (Redundant ones)**.** Let $\mathsf{2}^\ell \in \mathsf{d}_k$ for $\ell > 0, even$. Then for any $u > 0$ it makes no difference if $\mathsf{d}_{k+\ell u}$ contains $\mathsf{1}$ or not.

Theorem 3.17 (About standard form)**.** Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ with d_1 even. Consider a game $\mathsf{E} = {\cdot}\mathsf{e}_1\mathsf{e}_2\mathsf{e}_3\ldots$ constructed using the following rule:

$$\mathsf{e}_r \text{ contains } \mathsf{2}^{h+1}{-}\mathsf{1} \quad \text{whenever} \quad \mathsf{d}_{r-h+1} \text{ contains } \mathsf{2}^h.$$

Then $\mathsf{E} \equiv_1 \mathsf{D}$ (D is first cousin of E).

Example 3.18. We will show that $\mathsf{{\cdot}137} \equiv_2 \mathsf{{\cdot}4}$: First by Theorem 3.14 the game $\mathsf{{\cdot}4} \equiv \mathsf{{\cdot}421}$. Then by rule 3.17 we get $\mathsf{{\cdot}07} \equiv_1 \mathsf{{\cdot}421}$. Again by Theorem 3.14 we get $\mathsf{{\cdot}07} \equiv \mathsf{{\cdot}0731}$ and then by rule 3.17 follows $\mathsf{{\cdot}137} \equiv_1 \mathsf{{\cdot}0731} \equiv \mathsf{{\cdot}07} \equiv_1 \mathsf{{\cdot}321} \equiv \mathsf{{\cdot}4}$.

Definition 3.19. A game $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ is in a *standard form* if no further application of 3.17 does not change its representation.

Lemma 3.20. If a game $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ is in standard form then d_1 is odd.

Proof. Application of Theorem 3.14 ensures that there will exist $k > 0$ such that $1 \in \mathsf{d}_k$. Then a single application of rule 3.14 ensures that $1 \in \mathsf{d}_{k-1}$. So when no further application of this rule is not possible, it has to $\mathsf{1} \in \mathsf{d}_1$. □

Theorem 3.21 (The m-plicate games, [39, ch. 4])**.** Let $\mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ be a game with $d_i \in \{\mathsf{0}, \mathsf{7}\}$ for all i with no isolated 7s (if $\mathsf{d}_i = \mathsf{7}$ then either $\mathsf{d}_{i-1} = \mathsf{7}$ or $\mathsf{d}_{i+1} = \mathsf{7}$). Now consider the m-plicate game $\mathsf{M} = {\cdot}\mathsf{m}_1\mathsf{m}_2\mathsf{m}_3\ldots$ where each run of 7s of length ℓ is replaced by a run of 7s of length $m\ell$. Then $\mathcal{G}_{\mathsf{M}}(n) = \mathcal{G}_{\mathsf{D}}\big(\lfloor \frac{n}{m} \rfloor\big)$.

Example 3.22. A 3-plicate of the game KAYLES (·77) is ·777777. Kayles has nim-sequence (0 1 2 3 1 4 3 . . .), so we get the following nim-sequence:

$$\mathcal{G}\text{-sequence}({\cdot}777777) = (000\ 111\ 222\ 333\ 111\ 444\ 333\ldots).$$

There is not much more known about the code-digit games in general. However, there are many subclasses of code-digit games, usually restricted by a range of allowed types of digits, that can be used. These games have received much higher attention and we will dedicate them the rest of this chapter.

3.3 Subtraction Games

In this section, we will survey the known results on the most widely studied subclass of taking games, the **subtraction games**.

The ruleset of this generalization of the game Nim is defined by a set of integers called the **subtraction set** $\mathcal{S} = \{s_1, s_2, s_3, \ldots\}$. A move is to choose a heap and remove any number of tokens, provided that the number is in s. Formally, an $\mathcal{S}$-heap is defined as

$$H_n = \{H_{n-s} : s \in S, s \leqslant n\}.$$

The following open problem has been stated on the first place of all the publications of GONC [55, 58, 56, 57] and even before in the original Guy's lists of unsolved problems in combinatorial games [40, 9] back in 1991:

***Open Problem* 1.** *[40, problem 1] Investigate the relationship between the subtraction set and the length and structure of its period.*

We will denote such games $S(s_1, s_2, s_3, \ldots)$ and we will assume that $s_1 < s_2 < s_3 < \ldots$. For brevity, we will freely interchange between the terms subtraction games and subtraction set based on the context.

Clearly, these games are code-digit games, since any subtraction set can be translated into the code-digit game $\mathsf{S} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ with

$$\mathsf{d}_0 = 0, \quad \text{and} \quad \mathsf{d}_i = \begin{cases} 3 & \text{if } i \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$$

If the cardinality of $\mathcal{S}$ (and also the length of the code-digit game) is finite, we say that the subtraction game is **finite**, otherwise the subtraction game is **infinite**.

3.3.1 Structure of Subtraction Games

There is still little known about the structure of the nim-sequences of subtraction games in general. Even the relationship of the subtraction sets and the lengths of the pre-period and period is so far eluding.

T. S. Ferguson [18] has in 1974 observed and proved a surprising property of perfect pairing between the nim-values 0 and 1 in *any* subtraction game: for each zero in the nim-sequence there is always a nimber one in the distance of the smallest subtraction number on the right. More formally,

Proposition 3.23 (Ferguson's Pairing Property)**.** Let $S(s_1, s_2, s_3, \ldots)$ be a subtraction game. Then

$$\mathcal{G}(n) = 1 \text{ if and only if } G(n - s_1) = 0.$$

Proof. By contradiction. Let n is the least number for which the above statement fails. The following cases could occur

(a) $\mathcal{G}(n) = 1$ and $\mathcal{G}(n-s_1) \neq 0$. So for some move $t \in \mathcal{S}$ must $\mathcal{G}(n-s_1-t) = 0$ which by minimality of n implies $\mathcal{G}(n-t) = 1$, so $\mathcal{G}(n) \neq 1$.

(b) $\mathcal{G}(n) \neq 1$ and $\mathcal{G}(n - s_1) = 0$. But this means that $G(n) > 1$ so for some $t \in \mathcal{S}$ we have $\mathcal{G}(n - t) = 1$. By minimality of n this implies $\mathcal{G}(n - s_1 - t) = \mathcal{G}(n - t - s_1) = 0$, so $G(n - s_1) \neq 0$.

□

Nathan Fox, in search of aperiodic bounded nim-sequence, proved a generalized version of this proposition:

Proposition 3.24 (Generalized Ferguson [22, th. 1])**.** Let k be the smallest positive multiple of s_1 not in $\mathcal{S}$. Then for any integer n:

(1) For all $i \in \{0, \ldots, k-2\}$, if $\mathcal{G}(n) = i$ then $\mathcal{G}(n + s_1) = i + 1$, and

(2) for all $i \in \{1, \ldots, k-1\}$, if $\mathcal{G}(n) = i$ then $\mathcal{G}(n - s_1) = i - 1$.

Proof. Similarly as in Proposition 3.23, we will proceed by contradiction, assuming n is the least number for which the statement fails. Then either

(a) $\mathcal{G}(n) = i$ for some $i \in \{0, \ldots, k-2\}$, while $\mathcal{G}(n + s_1) \neq i + 1$. But since n is the first value that failed (1), for all $h \in \{0, \ldots, i\}$ we have $\mathcal{G}(n - h \cdot s_1) = i - h$. Since $h \cdot s_1 \in \mathcal{S}$ and also $\mathcal{G}(n + s_1) \neq i + 1$, it must $\mathcal{G}(n + s_1) > i + 1$. Therefore there must exist a $t \in \mathcal{S}$ such that $\mathcal{G}(n + s_1 - t) = i + 1$. But again by minimality of n we have $\mathcal{G}(n - t) = i$, a contradiction.

(b) $\mathcal{G}(n) = i$ for some $i \in \{1, \ldots, k-1\}$, while $\mathcal{G}(n - s_1) < i - 1$. Let $\mathcal{G}(n - s_1) = i - h$ for some $h \geqslant 2$. Then there exists a move $t \in \mathcal{S}$ such that $\mathcal{G}(n - t) = i - h + 1$. But then $\mathcal{G}(n - t) = i$, a contradiction.

(c) $\mathcal{G}(n) = i$ for some $i \in \{1, \ldots, k-1\}$, while $\mathcal{G}(n-s_1) > i-1$. Then there exists a move $t \in \mathcal{S}$ such that $\mathcal{G}(n-s-t) = i-1$, so $\mathcal{G}(n-t) = i$, a contradiction.

$\square$

The Ferguson's property tells us that from 1 the move s_1 always leads to 0. Similar fact can be shown about a move from 1 to 0:

Proposition 3.25. [68, WW, p. 442] Let $\mathcal{S}(s_1, s_2, \ldots)$ be a subtraction game and $n \geqslant s_1$ integer. Then

$$\mathcal{G}(n) = 0 \text{ implies that there exists } s \in \mathcal{S} \text{ such that } \mathcal{G}(n-s) = 1.$$

Proof. For contradiction, let n be the smallest position violating this rule. So for all $s \in \mathcal{S}$ is $\mathcal{G}(n-s) > 1$. Let $\mathcal{G}(n-s_1) = k$ for some $k > 1$. Then there exists $t \in \mathcal{S}$ such that $\mathcal{G}(n-s_1-t) = 0$. But by Ferguson we have that $\mathcal{G}(n-t) = 1$ which contradicts the assumption that there is no move to 1. $\square$

The next proposition gives us an insight that for any finite subtraction set, there exists a lost pile that is arbitrarily large (unlike for Nim).

Proposition 3.26. [6, l. 3.1] Let $S(s_1, s_2, \ldots, s_k)$ be a finite subtraction game. Then its kernel is infinite.

Proof. Let $K = \text{kernel}(\mathcal{S})$ is finite and let $m = \max(K)$. Then for all $n > m + s_k$ there is no move to K, so $\mathcal{G}(n) = 0$, a contradiction. $\square$

3.3.2 Periodicity of Subtraction Games

The following theorem is key in the analysis of finite subtraction games.

Theorem 3.27 (Finite periodicity)**.** Let $S(s_1, s_2, \ldots, s_k)$ be a *finite* subtraction game. Then its nim-sequence is (ultimately) periodic.

Proof. Note that by Proposition 3.44, the sequence is bounded by m. Observe that there are finitely many blocks of length s_k of non-negative integers smaller than m. So there must exist integers $a, b, a < b$ such that the blocks of length s_k beginning at a and b are the same, more precisely $\mathcal{G}(a+i) = \mathcal{G}(b+i)$ for all $i \in [0, s_k)$. Since the value $\mathcal{G}(n)$ depends only on preceding s_k $\mathcal{G}$-values, by induction on the number of blocks of length s_k we can easily show that $\mathcal{G}(n)$ is periodic with period $b-a$ and pre-period a. $\square$

Corollary 3.28. Both the pre-period n_0 and period p of a subtraction game $\mathcal{S}$ is bounded by $s_m \cdot m^{s_m}$, where $s_m = \max(\mathcal{S})$ and $m = |\mathcal{S}|$.

Theorem 3.29. (Subtraction Periodicity Theorem) Let $S(s_1, s_2, \ldots, s_k)$ be subtraction game and suppose that there exist $p \geqslant 1$ and $m_0 \geqslant 0$ such that

$$\mathcal{G}(n) = \mathcal{G}(n+p) \qquad \text{for all} \quad n \in [m_0, m_0 + s_k).$$

Then its nim-sequence is periodic with a period of length p and some pre-period $n_0 \leqslant m_0$.

Proof. It is enough to show that $\mathcal{G}(n) = \mathcal{G}(n+p)$ for all $n \geqslant m_0 + s_m$. By induction on n let us suppose this holds for all nim-values less than n and consider a value of $\mathcal{G}(n)$. By definition $\mathcal{G}(n) = \mathit{mex}\{\mathcal{G}(n-s) : s \in \mathcal{S}\}$. Equally $\mathcal{G}(n+p) = \mathit{mex}\{\mathcal{G}(n+p-s) : s \in \mathcal{S}\}$. But because $n - s < n$ for all $s \in \mathcal{S}$, from induction hypothesis we get that $\mathcal{G}(n-s) = \mathcal{G}(n+p-s)$ for all $s \in \mathcal{S}$ so the mex value is calculated from same set, therefore $\mathcal{G}(n) = \mathcal{G}(n+p)$. $\square$

Corollary 3.30. This theorem provides us an efficient algorithm for testing whether the nim-sequence of subtraction game is periodic with given period p and some pre-period $n_0 \leqslant m_0$.

The periodicity of finite subtraction games assures us, that the problem OUTCOME will always be polynomially solvable. First we pre-compute all the values $\mathcal{G}(0), \mathcal{G}(1), \ldots, \mathcal{G}(n_0 + p - 1)$ which takes a constant time and space relatively to the size of the game description (not relatively to the size of the ruleset, but that is a different problem). Then the nimber of given a position $G = H_{a_1} + H_{a_2} + \ldots + H_{a_k} +$ can be computed as

$$\mathcal{G}\big(G\big) = \mathcal{G}\big(n_0 + |a_1 - n_0|_p\big) \oplus \ldots \oplus \mathcal{G}\big(n_0 + |a_k - n_0|_p\big).$$

Clearly, this computation can be done in polynomial time.

Note that no algorithm whose complexity does not depend on the period length is known, which unfortunately does not have a better bound than stated in Corollary 3.28. This bound on the length of period is rather enormous. Unfortunately no closer bound is known.

Nevertheless, the computation results show that the period is usually much shorter. One of the longest known classes of games with long period were discovered by Althöfer and Bültermann. They have shown the structure of an infinite number of subtraction games having the superlinear pre-period length and also games having even cubic period lengths, measured in the size of the smallest subtracted number.

Note that Althöfer and Bültermann [2, p. 118] use a slightly different definition of the subtraction games: “the winner is the player who makes the size of a heap non-positive”. However, the results have shown not to depend on this small change of rules.

Observation 3.31 (A superlinear pre-period length, [2, p. 118])**.** The game $\mathcal{S}(2s, 4s+1, 22s+2)$ has for $2 \leqslant s \leqslant 20$ pre-period length $n_0 = 24s^2 - 4s + 1$.

Observation 3.32 (A cubic period length, [2, ch. 3])**.** The game $\mathcal{S}(s, 4s, 12s+1, 16s+1)$ has for $1 \leqslant s \leqslant 26$ period length $p = 56s^3 + 52s^2 + 9s + 1$.

Guy conjectures the following bound on the length of the period of a finite subtraction game:

***Conjecture* 1.** *[56, CONG, A1] Let $S(s_1, s_2, \ldots, s_k)$ be any finite subtraction game. Then its period is bounded by a polynomial of degree $\binom{k}{2}$ in s_k.*

Another open problem regarding the period of subtraction games was asked by Althö fer and Flammenkamp at 2002 Dagstuhl Seminar on Algorithmic Combinatorial Game Theory:

***Open Problem* 2.** *[12, p. 2] For a fixed $c > 0$ is it possible to find a subtraction game $S(s_1, s_2, \ldots, s_k)$ with period length $2^{c \cdot m_k}$?*

In [2] Althö fer also noticed that for any periodic subtraction game $\mathcal{S}$, the period of the -sequence($\mathcal{S}$) is always longer than the period of the outcome sequence($\mathcal{S}$) (the outcome sequence). This is due to the following observation:

Observation 3.33. If the nim-sequence is periodic with period p, then the outcome sequence is also periodic with period p (but might admit even shorter period).

He then asks the following question:

***Question* 2.** *[2, q. 4] The period of nim-sequence might be longer than the period of the outcome sequence. What are the extremal examples?*

3.3.3 Equivalences among Subtraction Games

Here we will discuss situations when a completed analysis for some game might help with solving another game, similarly as in Section 3.2.2.

Definition 3.34. Let $a_0 a_1 a_2 \ldots$ be some sequence. We define the ***m*-plicate** sequence of (a_n) as the sequence $a_0^m a_1^m a_2^m \ldots$, where each value is repeated m-times.

Theorem 3.35 (The g.c.d Property)**.** Let us denote the *greatest common divisor* of the elements of the subtraction $S(s_1, s_2, s_3, \ldots)$ as $g = \mathit{gcd}(\mathcal{S})$. Then nim-sequence of $\mathcal{S}$ is the g-plicate of the nim-sequence of the subtraction game $\mathcal{S}/g$ defined as $\mathcal{S}/g := \{\frac{s_1}{g}, \frac{s_2}{g}, \frac{s_3}{g}, \ldots\}$.

Proof. Consider any $\mathcal{S}$-heap of n tokens. Let q, r be maximal integers such that $n = q \cdot g + r$ and $r \in [0, g)$. Because each move subtracts some multiple of g, the size of the heap modulo g will not change during the game and will always be equal to r. □

Corollary 3.36. Let the nim-sequence of $S(s_1, s_2, s_3, \ldots)$ has period p and pre-period n_0. Then the nim-sequence of the game $k\mathcal{S}$ defined as $k\mathcal{S} := \{k \cdot s_1, k \cdot s_2, \ldots\}$ has period $k \cdot p$ and pre-period $k \cdot n_0$ for any $k \in \mathbb{N}$.

Thus, it suffices to consider subtraction sets whose members are relatively prime.

In Winning ways, Guy and Conway describes an interesting property of subtraction games:

Proposition 3.37. [42, WW, p. 84] Let $\mathcal{S}$ be a subtraction game. Then any number s, such that $\mathcal{G}(n) \neq \mathcal{G}(n+s)$ for all n, can be added to $\mathcal{S}$ without changing the sequence.

They listed all these **optional extras** for all subtraction games having members of subtraction set up to 7.

The following analysis of this property is due to Ho who has explored optional extras in more depth while solving several three-element subtraction games [43].

Definition 3.38. Let $\mathcal{S}$ be a subtraction game. We define its **expansion set**[7] or simply its **expansion** as

$$\mathit{exp}(\mathcal{S}) := \{s : \mathcal{G}(n) \neq \mathcal{G}(n+s), n \in \mathbb{N}_0\}.$$

We say that such $s \in \mathit{exp}(\mathcal{S})$ can be **adjoined** to $\mathcal{S}$.

Definition 3.39. Let S be a set of non-negative integers, $p > 1$ an integer. We define a ***p*-expansion** of S as

$$S^{*p} := \{s + k \cdot p : s \in S, k \in \mathbb{N}\}.$$

Proposition 3.40 (Partition of Expansion Set, [43, th. 1])**.** Let $\mathcal{S}$ be a *periodic* subtraction game with pre-period n_0 and period p. Then there exist *finite* sets S_1 and S_2 such that the expansion set $\mathit{exp}(\mathcal{S})$ can be finitely presented by

$$\mathit{exp}(\mathcal{S}) = S_1 \cup S_2^{*p}.$$

[7] Ho denotes the expansion set of $\mathcal{S}$ as $\mathcal{S}^{ex}$.

Proof. First we show that a number $s \geqslant n_0 + p$ can be adjoined to $\mathcal{S}$ if and only if the number $s - p$ can be. But that is trivial, since for all $n \geqslant n_0 + p$ is $\mathcal{G}(n-s) = \mathcal{G}(n-p-s)$ so the property in Definition 3.38 will be satisfied for s whenever it will be satisfied for $s - p$. Now define

$$S_1 = \{s : \mathcal{G}(n) \neq \mathcal{G}(n+s) : 0 \leqslant s < n_0\} \text{ and}$$
$$S_2 = \{s : \mathcal{G}(n) \neq \mathcal{G}(n+s) : n_0 \leqslant s < n_0 + p\}.$$

From the above observation follows that $\mathit{exp}(\mathcal{S}) = S_1 \cup S_2^{*p}$. $\square$

Observation 3.41. Notice that if $s_k = \mathit{max}(\mathcal{S})$ than the sets S_1 and S_2 can be obtained by checking the property of Definition 3.38 only for n with $0 \leqslant n < n_0 + p + s_k$.

Corollary 3.42. If $\mathcal{S}$ has expansion set $\mathit{exp}(\mathcal{S})$ then for the expansion set of the game $k\mathcal{S}$ holds $\{ks : s \in \mathit{exp}(\mathcal{S})\} \subseteq \mathit{exp}(k\mathcal{S})$ for any $k \in \mathbb{N}$.

3.3.4 Bound of Subtraction Games

Now we could ask ourselves the following question: what kind of nim values can occur in the nim-sequence? If the set is $\mathcal{S} = (1, 3, 5, \ldots)$ (all odds), the periodic sequence is $\overline{01}$, so we have only two values. On the other hand, the set $\mathcal{S}(1, 2, 3, \ldots) = \mathbb{N}_0$ has the arithmetic-periodic sequence $01234\ldots = \overline{0}\ (+1)$, so the set of nim-values is infinite (this actually is exactly the game of Nim).

Definition 3.43. We say that the nim-sequence of the subtraction game $\mathcal{S}$ is **bounded** (**by** m) if for all $n \geqslant 0$ is $\mathcal{G}(n) < m$. If no such m exists, we say it is **unbounded**. The number m is called the **bound** of the nim-sequence.

Note: Let $\mathcal{S}$ be a bounded subtraction game and let k be a minimal integer such that $\mathcal{S}$ is bounded by 2^k. Then the game has has a *nim-dimension* k.

Because the nim-sequence is calculated using a mex function, clearly if a nimber n occurs in the sequence then the nimbers $0, 1, \ldots, n-1$ must have occurred before it. The following proposition follows trivially.

Proposition 3.44. Let $S(s_1, s_2, \ldots, s_k)$ be a finite subtraction game. Then the range of its nim-sequence is bounded by k.

Proof. Each value of nim-sequence $\mathcal{G}(n)$ is determined by at most k previous values. Since always $\mathit{mex}(\mathcal{S}) \leqslant |\mathcal{S}|$, it follows that the value is bounded by k. $\square$

Observation 3.45. [6, p. 6] There exists a ultimately periodic subtraction set $\mathcal{S}$ such that the pre-period range is smaller, equal and greater than the range of the period.

Proof. Smaller: $\mathcal{S}(1,4,10)$, equal: $\mathcal{S}(1,8,11)$, greater: $\mathcal{S}(1,2,6,11)$, easily verifiable by hand or computer. □

3.3.5 Infinite Subtraction Games

Theorem 3.29 tells us that all finite subtraction games are periodic. We might ask what kind of periodicity can be observed in infinite subtraction games. Later on in this chapter will see that there are many periodic nim-sequences which only extend the finite ones (simplest example of these is $\mathcal{S}(odds)$ which extends the game $\mathcal{S}(1)$). Conversely, the game of Nim ($\mathcal{S} = \mathbb{N}_0$) is the simplest example of a game with aperiodic and unbounded nim-sequence. But the following question has attracted many researchers:

***Question* 3.** *Is there a subtraction game with aperiodic and bounded nim-sequence?*

In 2014, Fox presented a positive answer to this question [22]. Using a technique of *combinatorics of words* he showed an example of infinite subtraction game with aperiodic nim-sequence bounded by 3. Both, the description of the set and the proofs are rather complicated. This motivated Larsson [50], who has in 2015 shown a much simpler game (and simpler analysis) with the same properties. The game is defined by the following subtraction set

$$\mathcal{S} = \{F_{2n+1} - 1 : n \in \mathbb{N}\},$$

where F_n is n-th Fibonacci number defined by $F_n = F_{n-1} + F_{n-2}$ with $F_0 = 0$ and $F_1 = 1$. [74].

Blackham in his master thesis [6, th. 4. 2] pushes analysis of this problem further by introducing *uncountably* many of aperiodic subtraction games of nim-dimension 2.

> Suppose $S(s_1, s_2, s_3, \ldots)$ be a subtraction game with $s_1 = 1, s_2 = 4$, and for each $n > 2$ with $s_n = 2s_{n-1} - s_{n-2} + k_{n-2}(s_{n-1} + 1)$ where $(k_1, k_2, \ldots)$ is a sequence of non-negative integers. Let us define a mapping g from these "k-sequences" to a nim-sequence as follows: $g(k_1, k_2, \ldots)$ in the following way: $g(\varnothing) = 01012$, $g'(\varnothing) = 012$ [8] and for $q > 0$ by
>
> $$g(k_1, \ldots, k_q) = g(k_1, \ldots, k_{q-1})^{k_q+1}\, g'(k_1, \ldots, k_{q-1}),$$
> $$g'(k_1, \ldots, k_q) = g(k_1, \ldots, k_{q-1})^{k_q}\, g'\{k_1, \ldots, k_{q-1}).$$

[8] $\varnothing$ here denotes an empty sequence.

Then $\mathcal{G}$-sequence$(\mathcal{S}) = \overline{g(k_1, \ldots, k_{m-1})}$ and any infinite k-sequence, such that for any n_0 exists $n \geqslant n_0$ with $k_n > 0$, generates a different aperiodic nim-sequence bounded by 4. So they are uncountably many of those.

A more general class of infinite subtraction games have been described by Angela Siegel in 2005 [73]. Motivated by the results on finite subtraction games she focused on their complements:

Definition 3.46. [71] Let $\hat{\mathcal{S}}$ be a *finite* set. We define **All-but subtraction game**[9] $\hat{\mathcal{S}}(\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k)$ as a subtraction game with the infinite subtraction set $\mathbb{N}\backslash\hat{\mathcal{S}}$.

It shows that the analysis of these games is similarly simple as of the finite ones.

Theorem 3.47. [1, LIP, p. 149] Every All-but subtraction game is arithmetic periodic.

3.3.6 Solutions for Various Subtraction Games

In this section, we survey various solutions for particular subtraction games published in the literature. As we already mentioned, in Winning Ways are listed solutions for all subtraction games having members of subtraction set smaller than 8. By using Theorem 3.29 and enough computation power, we are able to find a solution for any finite subtraction game. So among finite subtraction games, only a general solutions or solutions of games having unusual properties are interesting. (Say some infinite subset of subtraction games or a game with maximal known period.)

Symmetric Subtraction Games

Blackham noticed a subclass of subtraction games which are “somewhat well-behaved”.

Definition 3.48. Finite subtraction game $\mathcal{S}$ is **symmetric** if there exists some number r so that $r - s \in \mathcal{S}$ whenever $s \in \mathcal{S}$. Number r is then called the **modulus** of $\mathcal{S}$.

Theorem 3.49. [6, th. 2. 1] Let $\mathcal{S}$ be a symmetric set with modulus r. Then the game is strictly periodic with period dividing r.

[9] Originally called *Finite excluded subtraction game.*

Proof. We will show that for any $n \in \mathbb{N}_0$ is $\mathcal{G}(n) = \mathcal{G}(n+r)$. For contradiction, assume that for some n we have $\mathcal{G}(n) \neq \mathcal{G}(n+r)$. Let $g = \mathcal{G}(n)$ and consider the following cases:

(a) $\mathcal{G}(n+r) > \mathcal{G}(n)$. Then by definition of $\mathcal{G}(n+r)$ there must exist $s \in \mathcal{S}$ such that $\mathcal{G}(n+r-s) = g$. But the symmetry of $\mathcal{S}$ implies that $r-s \in S$, so we have that $\mathcal{G}(n+r-s) \neq g$ for any $s \in S$. So $\mathcal{G}(n+r) \leqslant g$.

(b) $\mathcal{G}(n+r) < \mathcal{G}(n)$. Let $h = \mathcal{G}(n+r), h < g$. Then there exists $t \in \mathcal{S}$ such that $\mathcal{G}(n-t) = h$. But periodicity implies that $\mathcal{G}(n-t+r) = h$, so $\mathcal{G}(n+r) \neq h$.

Both cases led to a contradiction, so $\mathcal{G}(n) = \mathcal{G}(n+r)$. □

Corollary 3.50. If the elements of the subtraction set $S(s_1, s_2, \ldots, s_k)$ form an arithmetic progression then the game $\mathcal{S}$ is pure periodic with period p dividing $s_1 + s_k$.

Corollary 3.51. Any finite subtraction game $\mathcal{S}$ having a nim-sequence with period p and pre-period n_0. Then the game with subtraction set $\mathcal{S}' = \mathcal{S} \cup \{p - s : s \in \mathcal{S}\}$ is *strictly* periodic with period dividing p.

Two and Three-element Subtraction Games

All two-element subtraction games are solved, the structure of the nim-sequence has been presented in Winning Ways and the expansion was given by Ho as follows:

Theorem 3.52. [5, p. 530], [43, th. 2] Consider two-element subtraction games $\mathcal{S}(a,b), a < b$, where a, b are relatively prime. We assume that if $a = 1$ than b is not odd, since this is simple bipartite game. Let $b = ka + r$ for some $r < a$. This game is purely periodic with period $p = a + b$ and nim sequence

$$\mathcal{G}(n) = \begin{cases} (0^a 1^a)^{\frac{k}{2}} 0^r 2^{a-r} 1^r & k \text{ is even;} \\ (0^a 1^a)^{\frac{k+1}{2}} 2^r & k \text{ is odd.} \end{cases}$$

Furthermore the game $\mathcal{S}(a,b)$ has non-empty expansion set if and only if $a + 1 < b \leqslant 2a$

$$exp(\mathcal{S}) = \{a, a+1, \ldots, b\}^{*(a+b)}.$$

Otherwise (either $a = 1$, $b = a + 1$ or $b > 2a$) the game is non-expandable.

Three-element subtraction games are known to be deceptively simple but stubborn to solve, and have not yet been generally solved. Guy and Nowakovski in the list of unsolved problems published the second GONC write:

> *"It would now seem feasible to give the complete analysis for games whose subtraction sets have just three members, but the detail has so far eluded those who have looked at the problem."* [58, p. 1]

Nevertheless, there are known some results on subclasses of these games:

Proposition 3.53. [43, ch. 3–4] Consider $\mathcal{S}(1, a, b)$-games. Ho [43] gives nim-sequences and expansions for all cases when a is odd and for $\mathcal{S}(a, k{\cdot}a{+}r)$-games where $k \in \{1, 2, 3\}, r < a$ when a is even.

Consider $\mathcal{S}(a, b, a{+}b)$-games where $2 \leqslant a < b$. Ho [43] gives nim-sequences and expansions for cases when $b = ka + r, r < a, k$ is odd.

***Conjecture* 4.** *Consider $\mathcal{S}(a, b, a + b)$-games where $b = ka + r, r < a, k$ is even. In [5, WW, p. 531] is claimed that the game is periodic with period $(2b + r)a$ without a proof. Ho in 2014 [43, ch. 2] confirms that this case is still unresolved.*

Ultimately Bipartite Subtraction Games

Let us now consider the simplest possible subtraction game. Those clearly are the games whose subtraction set has only a single element $\mathcal{S}(a)$ for some $a \geqslant 0$. By using the g.c.d theorem 3.35, the only interesting case is the parity game $\mathcal{S}(1)$ which is the game She-Loves-Me, She-Loves-Me-Not by definition. We will now show a bit stronger statement about this game.

Theorem 3.54. [43, ch. 3] Let $\mathcal{S}$ be a subtraction set with $\mathit{gcd}(\mathcal{S}) = 1$. Then $\mathcal{S}$ is bipartite if and only if $1 \in \mathcal{S}$ and all numbers in $\mathcal{S}$ are odd. Moreover, the $\mathcal{G}(n) = 0$ if and only if n is even. In other words, game $\mathcal{S}(1)$ has the following nim-sequence and expansion

$$\mathcal{G}(n) = \overline{01}; \qquad \mathit{exp}(\mathcal{S}) = \{2k + 1 : k \geqslant 0\}$$

Proof. (By induction on n) Let $1 \in \mathcal{S}$ and all elements of $\mathcal{S}$ are odd. Trivially $\mathcal{G}(0) = 0$. Suppose $n > 0$. If n is odd then all $\{n - s : s \in \mathcal{S}\}$ are even so by induction hypothesis, for all $s \in \mathcal{S}$ values $\mathcal{G}(n - s) = 0$, so by mex-rule $\mathcal{G}(n) = 1$. Similarly, if n is odd then for all $s \in \mathcal{S}$ values $\mathcal{G}(n - s) = 1$, so by mex-rule $\mathcal{G}(n) = 0$.

Conversely, suppose that $\mathcal{S}$ is bipartite and let $s = \mathit{min}(\mathcal{S})$. Clearly, $\mathcal{G}(n) = 0$ for all $n \in \{0, 1, \ldots, n-1\}$. Since the game is bipartite, it inductively follows that $\mathcal{G}(n) = 0$ if and only if $\left\lfloor \frac{n}{s} \right\rfloor$ is even. Now consider any $t \in \mathcal{S}$. Clearly, $\mathcal{G}(t + i)$ is won for all $0 \leqslant i < s$, thus $\left\lfloor \frac{t+i}{s} \right\rfloor$ is odd. It follows that t is an odd multiple of s. The rest follows from the fact that $\mathit{gcd}(\mathcal{S}) = 1$. $\square$

The above Theorem tells us that all the bipartite subtraction games are easily recognizable. Cairns and Ho studied their generalization which allows small heaps of size less then some n_0 behave without any restriction, but all heaps of size $n \geqslant 0$ to be bipartite.

Definition 3.55. [7, ch. 2] We say that subtraction game $\mathcal{S}$ is **ultimately bipartite** if it is ultimately periodic with the period $p = 2$.

Observation 3.56. Notice that the ultimate period length 2 by definition of $\mathcal{G}(n)$ clearly admits only the period $\overline{01}$.

Observation 3.57. If $\mathcal{S}$ is ultimately bipartite than all elements of $\mathcal{S}$ are odd.

The ultimately bipartite subtraction games exhibit an interesting property: the optimal move on large enough heap is actually any move at all. Thus, the solution to the problem Strategy is trivial. Similar property is true, for instance, for Nimania and its generalizations studied by Fraenkel, Loebl and Nešetřil in [30], fittingly named as “Games with a Dozing Yet Winning Player”.

In [7, th. 3] is presented another peculiar property of these games: for a game with large enough heaps we can always easily say if it is a winning or losing (and actually solving the problem Outcome in constant time as well as Strategy).

Theorem 3.58. Let $S(s_1, s_2, \ldots, s_k)$ be a ultimately bipartite subtraction game. Then for sufficiently large n, the game is winning if and only if n is odd.

Proof. If $1 \in \mathcal{S}$, the game is bipartite and the answer follows by Proposition 3.54. Otherwise, for contradiction assume that the converse holds and for large enough n is $\mathcal{G}(n) = 0$ if and only if n is odd. Let n be some odd lost position. Since it is lost, the game will end after an even number of moves in some $\mathcal{G}(m)$ with $m < s_1$. Notice that now m must be odd as well, so $m > 0$. Consider now the game on a heap of size $n - 1$. Since $m > 0$, the second player can use exactly the same winning strategy, ending after even number of moves at $\mathcal{G}(m-1)$, so $\mathcal{G}(n) = \mathcal{G}(n-1) = 0$, a contradiction. □

Corollary 3.59. If $\mathcal{S}$ is ultimately bipartite than the pre-period length is even.

Ho has also observed the following properties about the structure of the pre-period of bipartite subtraction games:

Proposition 3.60. [43, l. 4] Let $\mathcal{S}$ be an ultimately bipartite subtraction game. Then $n_0 \geqslant s_1 + s_m$.

Proposition 3.61. [43, l. 5] Let $\mathcal{S}$ be an ultimately bipartite subtraction game. Then

$$\mathcal{G}(n_0) = 0,\ \mathcal{G}(n_0 - 1) \geqslant 2 \quad \text{and} \quad \mathcal{G}(n_0 - 1 - s_m) = 1.$$

Theorem 3.62. [43, p. 17] Let $\mathcal{S}$ be an ultimately bipartite subtraction game. If $s \in \mathbb{N}$ can be *adjoined* to $\mathcal{S}$ then $s \leqslant s_m$.

And finally, Ho also identified several infinite classes of these games. The proofs are all based on identifying the set kernel or explaining the whole nim-sequence of the game and are all rather technical.

Theorem 3.63. [7, th. 2], [43, th. 5] The following subtraction games are ultimately bipartite:

- $\mathcal{S}(3, 5, 9, ..., 2^k + 1)$ for $k \geqslant 3$,
- $\mathcal{S}(3, 5, 2^k + 1)$ for $k \geqslant 3$,
- $\mathcal{S}(a, a + 2, 2a + 3)$ for odd $a \geqslant 3$,
- $\mathcal{S}(a, 2a + 1, 3a)$ for odd $a \geqslant 5$ (with pre-period length $n_0 = 2a^2 - a - 1$).

***Conjecture* 5.** *[43, p. 18] Ho conjectured that an ultimately bipartite subtraction game is non-expandable.*

Solutions for Infinite Subtraction Games

As we have seen, the finite subtraction games are periodic and the finite excluded subtraction games are arithmetic periodic. However, there is little known about the games which do not fit to either of these categories: the subtraction games for which both the subtraction set and its complement are infinite.

Many of these games trivially reduce to the finite ones, because each finite game has an infinite expansion set (see Section 3.3.3).

Pherwani has solved several subtraction games that involve only powers.

Proposition 3.64. [60, th. 3.3] Consider any subtraction game of powers of any base b: $\mathcal{S}(1, b, b^{r_1}, b^{r_2}, \ldots)$, where for all $i \geqslant 1$ is r_i any positive integer. Assume that b is even since odd b gives simple bipartite game. This game is purely periodic with period $(01)^{b/2}2$ of length $b/2 + 1$.

Proof. By showing that powers of even base b are subset of expansion set of a game $\mathcal{S}(1, b)$. For details see [60, th. 3.3]. $\square$

Proposition 3.65. [60, th. 3.7] The expansion set of the game $\mathcal{S}(1, b^2, b^3)$ contains all higher powers of b: $\{b^k : k > 3\} \subset \mathit{exp}(\mathcal{S})$.

The game Take-A-Square was first studied by Golumb [33]. He observed that its kernel is a set with the following property: it is *square-difference-free*, meaning that there are not any two elements that differ by some positive square. Eppstein [15] notes that the square-difference-free set generated this set is maximal: every positive integer outside kernel has a move to zero position, therefore differs by a square from some element in kernel. In other words, no other element can be added to the set without destroying the square-difference-free property. Golumb also notes that each maximal square-difference-free set on interval $[0, n]$ needs to have at least $\Omega(\sqrt{n})$ elements [33, th. 4.1].

***Question* 6.** *Vajda in his book "Mathematical Games and How to Play Them" lists several infinite subtraction games that, even though they have simple descriptions, it is not much known about the structure of their nim-sequence nor their strategy [79, p. 24]:*

$$\mathcal{S}(\textit{primes}),\ \mathcal{S}(\textit{odd primes}),\ \mathcal{S}(\textit{Fibonacci numbers})\ \textit{and}\ \mathcal{S}(\textit{Lucas numbers}).$$

Note that $\mathcal{S}(\textit{primes} \cup \{1\}) = \mathcal{S}(\textit{odds} \cup \{1\}) = \mathcal{S}(1)$.

Theorem 3.66. [57, A1] Every all-but subtraction game $\hat{\mathcal{S}} = \{\hat{a}, \hat{b}, \widehat{a+b}\}$ is strictly periodic with a period of length $3(a+b)$.

3.3.7 Aperiodic Subtraction Games

Fraenkel in his article "Aperiodic Subtraction Games" from 2011 [24] describes a different type of aperiodic subtraction games then those described in Section 3.3.1. These games have been generalized by Sopena in [77] as follows.

Definition 3.67. For some sets of integers S and D we define a **subtraction-division game** $\mathcal{SD}(S, D)$ as a heap game where players can in their move either subtract $s \in S$ tokens or divide size of some heap by $d \in D$. There are several variants of these games in terms of understanding the division move:

- $\mathcal{SD}(S, D)$ — There is a move into $\left\lfloor \frac{n}{d} \right\rfloor$ for each $d \in D, n \geqslant 1$. [24]
- i-Mark(S, D) — Player can use $d \in D$ on H_n only if d divides n. [77]
- UpMark(S, D) — There is a move into $\left\lceil \frac{n}{d} \right\rceil$ for each $d \in D, n \geqslant 1$. [24]

We denote $\mathcal{M}i\mathcal{SD}$, i-MiMark and MiUpMark as the misère variants of these games.

Note that these games are not invariant. Let us start with a proof that these games truly are aperiodic.

Proposition 3.68. [24, th. 9] The games $\mathcal{SD}(S, D)$m UpMark(S, D) and i-Mark(S, D) are aperiodic for any sets S, D with $D \neq \varnothing$.

Proof. For contradiction, assume that these games are ultimately periodic with a period p and pre-period n_0. Let d be some move $d \in D$ and consider a single-heap position H_n with $nk \cdot p \geqslant n_0$ for some k. Then in any of these games there exists a move $d \cdot n \to n$ which implies that $\mathcal{G}(d \cdot n) \neq \mathcal{G}(n)$. But the difference $dn - n = (d-1) \cdot n = (d-1) \cdot k \cdot p$ which is a multiple of p so it should $\mathcal{G}(d \cdot n) = \mathcal{G}(n)$, a contradiction. □

Since these games are aperiodic, polynomial solutions of these games need to somehow exploit some other regularities of the nim-sequence.

Definition 3.69. Let n be a non-negative integer. We write $R(n)$ as the binary representation of the number n. We say that n is:

- **evil**, if the number has even number of 1's in $R(n)$.
- **odious**, if the number has odd number of 1's in $R(n)$.
- **vile**, if the number has even number of 0's at the end of $R(n)$.
- **dopey**, if the number has odd number of 0's at the end of $R(n)$.

Theorem 3.70. [24, ch. 3] The game Mark $= \mathcal{SD}(\{1\}, \{2\})$ has its nim-sequence defined as follows:

$$\mathcal{G}(n) = \begin{cases} 0 & n \text{ is } \textit{dopey}, \\ 1 & n \text{ is } \textit{vile} \text{ and } \textit{odious}, \\ 2 & n \text{ is } \textit{vile} \text{ and } \textit{evil}. \end{cases}$$

To prove this property, it suffices to show that all the moves required by the definition of the nimbers all available, e.g. a move from vile odious position to a dopey one. Also, one needs show that no moves are possible inside these partitions. The proof is rather technical, so we omit it in this text and refer the reader to [24, ch. 3].

We summarize the other results on the aperiodic subtraction game in Table 3.1. We list the time complexities to compute the $o(H_n)$ and $\mathcal{G}(H_n)$ since these are the complexities of solving the problem Outcome on single-pile and multi-pile games, respectively.

Wythoff Games and Invariant Subtraction Games

Here we shortly introduce several generalizations of Nim and subtraction games that have been more widely studied.

Table 3.1: Known results on the aperiodic subtraction games.

Name	Game	$o(H_n)$	$\mathcal{G}(H_n)$	Citation
Mark	$\mathcal{SD}(\{1\},\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[24, ch. 2–3]
MiMark	$\mathcal{M}i\mathcal{SD}(\{1\},\{2\})$	$\mathcal{O}(\log n)$	-	[24, ch. 4]
UpMark	UpMark$(\{1\},\{2\})$	$\mathcal{O}(\log n)$	-	[24, ch. 5]
Mark-**t**	$\mathcal{SD}([1,t-1],\{t\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log^2 n)$	[37, th. 2.3]
MiMark-**t**	$\mathcal{M}i\mathcal{SD}([1,t-1],\{t\})$	$\mathcal{O}(\log n)$	-	[37, th. 3.1]
i-Mark	i-Mark$(\{1\},\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 2]
i-Mark-**t**	i-Mark$([1,t-1],\{t\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 3]
-	i-Mark$([1,t-1],\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 3]
-	i-Mark$(\{1,2\},\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 4]
-	i-Mark$(\{2,4\},\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 4]
-	i-Mark$(\{4,8\},\{2\})$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	[77, ch. 4]

Figure 3.4: Ruleset of a game Wythoff.

position: Two heaps of tokens, denoted as an ordered pair $(a,b) \in \mathbb{N}^2$.
ruleset: Remove any number of tokens from a *single* heap as in Nim or remove the *same* number of tokens from *both* heaps.

Figure 3.4 explains the ruleset of a game Wythoff which was invented by a Dutch mathematician Willem Wythoff in 1907. The alternative description of the game, which is known as Corner-The-Lady, is played on a semi-infinite chess board with a single queen. Players can in their move shift the queen only in one of three directions: up, left, and up-left diagonal. This way the Queen is getting closer and closer to the single corner, labeled $(0,0)$. The equivalence of this game with Wythoff is obvious.

The $\mathcal{P}$-positions of Wythoff display a striking geometric regularity: they cluster around the lines with slopes φ and $1/\varphi$, where $\varphi = (1+\sqrt{5})/2$ is the *golden ratio*.

Theorem 3.71. [14, ch. 1] A position (a,b) in Wythoff is a $\mathcal{P}$-position if and only if it has a form $(\lfloor \varphi n \rfloor, \lfloor \varphi^2 n \rfloor)$ for some non-negative integer n.

This regularity was discovered already by Wythoff itself. However for determining the $\mathcal{G}$-value of an arbitrary position is still an open problem. This has motivated many researchers to study various modifications and generalizations of Wythoff that would bridge the complexity gap between Nim and Wythoff.

The first modification is based on the following observation: we define a **star operator** applied on a heap game G as a game $G^\star$ which has the same moves as G plus new taking moves of sizes corresponding to the $\mathcal{P}$-positions of G, except 0. Notice that since a position in a Two-Pile-Nim is lost if and only if the sizes of piles equal, Two-Pile-Nim* =Wythoff. This operator was invented by Larsson, who has studied the game Wythoff* and showed nice properties of repeated application of $\star$ [49].

A well studied generalizations of Wythoff are Vector Subtraction Games[10] introduced by Golumb in 1966 [33]. A *t-Vector Subtraction Game* is played on t ordered heaps of tokens, so a position of such game is a t-tuple of non-negative integers. Moves are also a t-tuples corresponding to the number of tokens that are removed from each heap. For instance, the moves in Wythoff, as for 2-Vector subtraction game, are defined as $\mathcal{M} = \{(0,i),(i,0),(i,i) : i > 0\}$.

The vector subtraction games have achieved some results in answering the following general problem:

***Open Problem* 3.** *Let $\mathcal{A} \subset \mathbb{N}_0$. Is there a invariant normal play impartial heap-game with A as its set of $\mathcal{P}$-positions?*

Clearly, this will not be possible for classical subtraction game:

Observation 3.72. Let $A \subset \mathbb{N}_0$. The following statements are equivalent:

(a) There is an $a \notin \mathcal{A}$ such that for all $b \in \mathcal{A}$ with $b < a$ there is a pair of integers $x, y \in \mathcal{A}$ such that $y - x = a - b$.

(b) There is no subtraction game with $\mathcal{A}$ as its set of $\mathcal{P}$-positions.

Larsson therefore generalized subtraction games to so called *comply* subtraction games[11]) and are played on a single-heap H_n and defined using a family of finite subsets of $\mathbb{N}$, denoted $\mathscr{S}$. The move of a player goes as follows: first, the player to move proposes a set $\mathcal{S} \in \mathscr{S}$ satisfying $n \geqslant \mathit{max}(S)$. Then the player not to move chooses some $s \in \mathcal{S}$ and moves to $n - s$. Larsson showed that it is possible to find such game, that its set of $\mathcal{N}$-positions *avoids arithmetic progression*, meaning that there is no triple $\{x, x+d, x+2d\}$ in this set for any $x \geqslant 0, d > 0$.

For many other modifications of these games, see [14].

[10] Sometimes called also Take-Away Invariant Games, see [13].

[11] These games have been also known as the Blocking Nim, see [21]

3.4 Octal and Hexadecimal Games

In this section, we will discuss the taking and breaking games that have also the *breaking* move allowed. Are focus will be primarily on the subsets of code-digit games that have a restricted set of allowed digits in their code sequence.

As well as for the subtraction games, the most common approach in solving a code-digit game is to resolve its periodicity.

In general, it appears to be hard to find a function f such that given taking and breaking game Γ is f-periodic. However, by using a simple trick, we can generalize Theorem 3.29 to be able to resolve the simple periodicity of any code-digit game.

Theorem 3.73. (Simple periodicity theorem) Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots\mathsf{d}_\mathsf{k}$ be a finite code-digit game. Let $t \geqslant 0$ be maximal number of heaps that are created in a single move ($\mathsf{d}_\mathsf{i} < 2^{t+1}$ for all i). Suppose there exist integers $n_0 \geqslant 0$ and $p \geqslant 0$ such that the following holds:

$$\mathcal{G}(n+p) = \mathcal{G}(n) \quad \text{for} \quad n_0 \leqslant n \leqslant tn_0 + (t-1)p + k.$$

Then the nim-sequence(D) is periodic with pre-period n_0 and period p.

Proof. (By induction on n.) Let $n \geqslant tn_0 + (t-1)p + k$. Each option of H_{n+p} that takes i tokens from a heap has form $H_{a_1} + H_{a_2} + \ldots H_{a_\ell}$ with $a_1 + a_2 + \ldots + a_\ell = n + p - i$, such that $i \leqslant k$ and $\ell \leqslant t$ Since $n + p \geqslant tn_0 + tp + k$, we have $a_1 + a_2 + \ldots + a_\ell \geqslant tn_0 + tp$. Without loss of generality then $a_1 \geqslant n_0 + p$, because $\ell \leqslant t$. Thus $a_1 - p \geqslant n_0$ and by induction hypothesis, we have $\mathcal{G}(a_1 - p) = \mathcal{G}(a_1)$. Therefore $\mathcal{G}(a_1 - p) \oplus \mathcal{G}(a_2) \oplus \ldots \oplus \mathcal{G}(a_\ell) = \mathcal{G}(a_1) \oplus \mathcal{G}(a_2) \oplus \ldots \oplus \mathcal{G}(a_\ell)$. Now $a_1 - p > 0$ ($n_0 > 0$) so by using Observation 3.9 we get that $H_{a_1-p} + H_{a_2} + \ldots H_{a_\ell}$ is an option of H_n. Similarly we can show that every option of H_n is an option of H_{n+p}, so we conclude that $\mathcal{G}(n+p) = \mathcal{G}(n)$ for all $n \geqslant n_0$. □

Corollary 3.74. Each periodic code-digit game with pre-period n_0 and period p can be solved in time $\mathcal{O}(N^t \cdot k)$, where $N := tn_0 + (t-1)p + k$.

Proof. We can calculate first N terms of the nim-sequence by starting at some small length and then doubling its length and testing its periodicity until it is found. The periodicity testing can be done in linear time. The bottleneck of this calculation is therefore in the generation of the nim-sequence. We can generate it from definition by iterating on all viable moves and calculating their mex. □

3.4.1 Octal Games

The most famous family of code-digit games with allowed breaking are **octal games**, introduced by Guy and Smith as the first code-digit games in the famous paper from 1956 [39]. In the octal games, a move consists of removing some number of tokens from a heap and then optionally splitting the remaining tokens into *two* heaps. Thus, for each code digit $\mathsf{d}_i < 8$ holds, hence the name.

We have seen in the proof of Theorem 3.27 that there is an easy argument for the fact that all subtraction games are periodic. Whether the same is true for octal games is an important open question, that has been listed as the second problem (after the subtraction games analysis) in all Guy's lists of unsolved problems in combinatorial games since 1991:

***Open Problem* 4.** *[40, 9, problem 2] Is every finite octal game ultimately periodic?*

Guy described the behavior of the nim-sequences of octal games as "a considerable mystery" [41, E27]. Indeed, we seem to be far to understanding the behavior of these games in general. For instance, J.P. Grossman analyzed the game $\mathsf{6}$ up to heaps of size 2^{47} and has not yet confirmed any periodicity.

The following Theorem ensures us that we cannot answer the above open problem by finding an arithmetic periodic octal game.

Theorem 3.75. [71, CG, exercise 2.12] Octal game of finite length cannot be arithmetic periodic with non-zero saltus.

Proof. For $n \in \mathbb{N}$, denote by $f(n)$ the cardinality of the set

$$\{a \oplus b : a + b = n, a, b \in \mathbb{N}_0\}.$$

Observe that for each $n \in \mathbb{N}$ is $f(2n+1) = f(n)$. Indeed, since $2n+1$ is odd, any value $a \oplus b$ must be odd as well, because only one of a and b can be odd. So if $a \oplus b = x$ with $a + b = n$ then both $(2a+1) \oplus 2b = 2a \oplus (2b+1) = 2x \oplus 1$. So each of the $n+1$ possibilities of $a+b$ such that $a+b = n$, each maps to 2 unique pairs that have the same nim-sum, so none of the $2n+2$ possibilities of values a, b which give $a + b = 2n+1$ can achieve a new nim-sum.

Also notice that $f(2n) = f(n) + f(n-1)$. That is due to the fact that we can map each nim-sum $a \oplus b$ with $a + b = 2n$ and a and b being *even* to a nim-sum $\frac{a}{2} \oplus \frac{b}{2}$ with $\frac{a}{2} + \frac{b}{2} = n$ and thus is counted in $f(n)$. Also, we can map each $a \oplus b$ with $a + b = 2n$ and a and b being *odd* to a nim-sum $\frac{a-1}{2} \oplus \frac{b-1}{2}$ with $\frac{a-1}{2} + \frac{b-1}{2} = n - 1$ and thus is counted in $f(n-1)$. Since the nim-sums calculated in $f(n)$ are odd and the nim-sums calculated in $f(n-1)$ are even, it follows that these two sets are disjunct.

Since one of n and $n-1$ is odd, we have either $f(n) = f(a)$ for some $a \leqslant \frac{1}{2}n$ or else $f(n) = f(a) + f(b)$ for some $a \leqslant \frac{1}{2}n$ and $b \leqslant \frac{1}{4}n$.

Now we will prove by induction, that $f(n) \leqslant \frac{5}{4}n^\alpha$, where $(\frac{1}{2})^\alpha = \varphi$ for φ being the inverse of a golden ratio (the unique solution for $\varphi^2 + \varphi = 1$). After verifying $n = 1$ and $n = 2$ we can write

$$\begin{aligned} f(n) \leqslant f(a) + f(b) &\leqslant \frac{5}{4}\big((\frac{1}{2}n)^\alpha + (\frac{1}{4}n)^\alpha\big) \\ &= \frac{5}{4}(\varphi n^\alpha + \varphi^2 n^\alpha) = \frac{5}{4}n^\alpha. \end{aligned}$$

Now, if the nim-values of the arithmetic periodic game had a nonzero saltus, the number of distinct $\mathcal{G}$-values of heaps up to n would need to be greater than $\lambda n + \mu$ for some non-zero λ. But that is not possible, since we have shown that this value is less than $\frac{5}{4}n^\alpha$ with $\alpha = 0.694\ldots$. □

3.4.2 The Sparse Space Phenomenon

From theorem 3.73 we get that to solve a finite octal game $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots\mathsf{d}_\mathsf{k}$, one needs to calculate $N = 2n_0 + p + k$ its $\mathcal{G}$-values. If we calculate them by definition, this takes $\mathcal{O}(N^2)$ operations. Trying to solve a game such as $\cdot 16$ with $n_0 = 105,350$ and $p = 149,459$ does not seem quite feasible and a more advanced technique is required.

While analyzing the game of KAYLES, Berlekamp noticed an interesting property. If we closely look only at the $\mathcal{G}$-values that occur only in pre-period, we notice that these numbers $0, 3, 5$, and 6 all have *even* number of ones in their binary expansions. The rest of $\mathcal{G}$-values which appear in the period, so they occur infinitely (namely $1, 3, 4, 7$, and 8) have *odd* number of ones. We have seen that this distinction played an important role for aperiodic subtraction games (Theorem 3.70). Conway named them **evil** and **odious**, respectively [42, p. 109]. For GRUNDY'S GAME we can notice similar property if we ignore the first bit of binary expansions. This property generalizes for some octal games in so called **Sparse Space Phenomenon**.

Definition 3.76. A partition $\mathcal{R}, \mathcal{C}$ of $\mathbb{N}$ is called a **sparse space decomposition** if $\mathcal{R} \cup \mathcal{C} = \mathbb{N}$, $\mathcal{R} \cap \mathcal{C} = \varnothing$ and for each $c_1, c_2 \in \mathcal{C}, r_1, r_2 \in \mathcal{R}$:

$$\begin{aligned} r_1 \oplus r_2 &\in \mathcal{R}; \qquad & r_1 \oplus c_2 &\in \mathcal{C}; \\ c_1 \oplus c_2 &\in \mathcal{R}; \qquad & c_1 \oplus r_2 &\in \mathcal{C}. \end{aligned}$$

The integers $r \in \mathcal{R}$ are then called **rare**, while the integers $s \in \mathcal{C}$ are **common**.

Definition 3.77. A taking and breaking game Γ has a **sparse space** $\mathcal{R}$ if there exists a sparse space decomposition $\mathcal{R}, \mathcal{C}$ such that the set

$$\{n : \mathcal{G}(n) \in \mathcal{R}, n \in \mathbb{N}\} \quad \text{is finite.}$$

Theorem 3.78. [71, CG, p. 193] If a finite octal game $\mathsf{d_0{\cdot}d_1d_2d_3\dots d_k}$ has sparse space then it is ultimately periodic.

Proof. The definition of sparse space implies that there exists an integer n_0 such that for all $n \geqslant n_0 : \mathcal{G}(n) \in \mathcal{C}$. Then for all $n \geqslant 2n_0 + k$ the excludents used to calculate $\mathfrak{mex}$ for $\mathcal{G}(n)$ and are *common* will all have form:

$$\mathcal{G}(a) \oplus \mathcal{G}(b),$$

where $\mathcal{G}(a)$ is rare which implies that $b \geqslant n - n_0 - k$. Hence the value of $\mathcal{G}(n)$ depends only on last $n_0 + k$ values, so the sequence is ultimately periodic. □

Theorem 3.79. There exists a bijection $b : \mathbb{S} \to \mathbb{N}$, where $\mathbb{S}$ represents set of all sparse spaces, $\mathbb{S} \subseteq 2^{\mathbb{N}}$.

Proof. For a sparse space $\mathcal{R}$ the bijection is given by

$$b(\mathcal{R}) = \bigvee_{x \in \mathcal{R} \cap 2^{\mathbb{N}}} x,$$

where $2^{\mathbb{N}}$ denotes set of all non-negative powers of two. This of course deterministically maps an integer to each sparse space. It remains to proof that each integer there is a unique sparse space. Let us denote

$$\mathcal{A} = \left\{ 2^n : n \in \mathbb{N}, (2^n \,\&\, m) \neq 0 \right\}$$

and let $\mathcal{B} = 2^{\mathbb{N}} \backslash \mathcal{A}$. Because for any rare r_1, r_2 we have $r_1 \oplus r_2$ rare, all numbers generated using powers of two from $\mathcal{A}$ must be rare and all numbers generated using elements of $\mathcal{B}$ must be common. Also since for any common c_1, c_2 we have $c_1 \oplus c_2$ rare, all numbers generated using combination of any two elements of $\mathcal{B}$ must be rare. This directly implies the structure of rare numbers:

$$\mathcal{R} = \langle \mathcal{A} \cup \{a \oplus b : a, b \in \mathcal{B}\} \rangle.$$

Now for contradiction let $\mathcal{R}'$ be a different sparse space: $\mathcal{R} \neq \mathcal{R}'$ and $b(\mathcal{R}) = b(\mathcal{R}')$. Then $\mathcal{R} \cap 2^{\mathbb{N}} = \mathcal{R}' \cap 2^{\mathbb{N}}$. Also for any $a, b \in \mathcal{B}$ we have $a \oplus b \in \mathcal{R}'$, otherwise $a \oplus \mathcal{R}'$ and $b \oplus \mathcal{R}'$ would be distinct cosets of $\mathcal{R}'$, a contradiction. □

Definition 3.80. The bijection b from previous theorem we will call **the ignore mask** as it describes which bits to ignore while counting the one bits to determine if given $\mathcal{G}$-value is rare or common:

Theorem 3.81. Let D be an octal game, $\mathcal{R}$ its sparse space and $m = b(\mathcal{R})$ its ignore mask. Then

$$\mathcal{G}(n) \in \mathcal{R} \quad \Leftrightarrow \quad \text{populationCount}\left(\mathcal{G}(n) \,\&\, \neg m\right) \ \text{ is } \textit{even},$$

where populationCount(x) represents a number of ones and $\neg$ the logical negation on each bit in the binary expansion.

Proof. From proof of th. 3.79 we have that there is a partition $\mathcal{A}, \mathcal{B}$ of non-negative powers of two such that the the rare space can be described by $\mathcal{R} = \langle \mathcal{A} \cup \{a \oplus b : a, b \in \mathcal{B}\} \rangle$. In the formula above, the bits that correspond to elements in $\mathcal{A}$ are ignored by ignore mask and the only relevant ones that can change the population count are the ones that come from the set $\mathcal{B}$. We can see that since we start with nim-sum of pair of these elements, all rare elements have even number of one bits that correspond to elements of $\mathcal{B}$. □

Lemma 3.82. Let $\mathsf{D} = \mathsf{d}_0 \cdot \mathsf{d}_1 \mathsf{d}_2 \mathsf{d}_3 \ldots \mathsf{d}_\mathsf{k}$ is a finite octal game with a sparse space. Then its kernel is finite.

Proof. This comes directly from the fact that 0 is rare. If 0 was not rare, than for any common c we have $c \oplus 0 = c$ is rare, a contradiction. □

Observation 3.83. Until now, there has been no assumption made on the frequencies of rare and common values. It is worth noticing that if the rare values would occur rarely in the first n values of nim-sequence then, because common $\oplus$ common value is rare, typically many excludents will be rare, so $\mathcal{G}(n)$ itself is quite likely to be common. Hence when the distinction between rare and common value is established, it tends to persist.

Now, if the rare values really occur rarely, we can compute any common $\mathcal{G}$-value only by taking into account the moves that are made up of a rare and a common value. Berlekamp et al. confirm that this approach really speeds up the computation [42, WW, p. 102]. For instance, when calculating the Grundy's game value $\mathcal{G}(250, 001)$ this way, it is required to calculate only $\mathcal{G}(1, 273)$ nim-sums. They estimate that the computation on average takes only a few thousand operations.

3.4.3 Computational Results on Octal Games

The original paper introducing octal games from 1956 lists results on several single and two digit octal games (probably computed by hand) [39]. Among them, the longest period has the game Guiles of the code $\cdot\mathsf{15}$ with $n_0 = 1$ and $p = 10$. They also describe several infinite octal games and their arithmetic periodicity, e.g. the Duplicate Nim $\cdot\overline{\mathsf{03}}$ with period $\overline{00}$ $(+1)$, the Triplicate Nim $\cdot\overline{\mathsf{003}}$ with period $\overline{000}$ $(+1)$ and many others.

First computational results directed to octal games dates to 1967 when Kenyon [47] assembled first data on at most 3 code digits octal games. His most notable result finding is prime period 349 for $\cdot\mathsf{156}$. Austin extended his tables in 1976, especially by finding period of length 1550 for $\cdot\mathsf{165}$ [4].

In 1982 Berlekamp, Conway and Guy published an extensive compendium of three-digit octal games in [42, WW, p. 102] alongside with the introduction to sparse spaces.

Table 3.2: Non-trivial octal games with known structure [20].

Game	Period	Pre-period	Solved
·45	20	498	1956
·156	349	3,479	1967
·055	148	259	1976
·644	442	3,256	1976
·356	142	7,315	1976
·165	1,550	5,181	1976
·127	4	46,578	1988
·56	144	326,640	1988
·16	149,459	105,351	1988
·376	4	2,268,248	1988
·454	60,620,715	160,949,019	2000
·104	11,770,282	197,769,598	2001
·106	328,226,140,474	465,384,263,797	2002
·054	10,015,179	193,235,616	2002
·354	1,180	10,061,916	2002

In 1989 Gangolli and Plambeck [32] analyzed using Berlekamp's sparse spaces method the game ·16 with pre-period 105,350 and period length 149,459.

In 2000 Flammenkamp discovered game ·454 with period 60,620,715 and pre-period 160,949,019, and in 2002 the game ·106 with period 328,226,140,474 and pre-period 465,384,263,797 which still holds a record.

Flammenkamp also has a website [20] that lists the whole up-to date compendium alongside with the latest computation results concerning octal games. Some results on non-trivial games from this lists are showed in Table 3.2.

In 2011 J. P. Grossman analyzed the only unsolved single digit octal game ·6, computing its $\mathcal{G}$-values up to 2^{47} by several low-level optimizations and parallel computation, still finding no periodicity [34].

Some of these games are also listed in Nowakowski's Unsolved problems in combinatorial games in Games of no chance from 2015 [57].

***Open Problem* 5.** *[57, A2] Resolve any number of the following 74 unsolved 3-digit octal games:* ·6 *(*Officers*),* ·04 , ·06 , ·14 , ·36 , ·37 , ·64 , ·74 , ·76 , ·004, ·005, ·006, ·007 *(*Treblecross*),* ·014, ·015, ·016, ·024, ·026, ·034, ·064, ·114, ·125, ·126, ·135, ·136, ·142, ·143, ·146, ·162, ·163, ·164, ·166, ·167, ·172, ·174, ·204, ·205, ·206, ·207, ·224, ·244, ·245, ·264, ·314, ·324, ·334, ·336, ·342, ·344, ·346, ·362, ·364, ·366, ·371, ·374, ·404, ·414, ·416, ·444, ·564, ·604, ·606, ·744, ·764, ·774, ·776, 4·004, 4·007, 4·026, 4·044, 4·045, 4·324, 4·327 *and* 4·367.

***Open Problem* 6.** *[71, CG, p. 197] Identify an octal game with period longer than the game* $\cdot$106 *with a period of length* 328, 226, 140, 474.

3.5 Hexadecimal games

Hexadecimal games are code-digit games where we are allowed to split the remainder into at most three piles, so for each code digit $\mathsf{d}_i < 16$ holds. To keep the game notation concise, we will use letters A, B, C, D, E and F for the numbers 10 through 15, e.g. game $\cdot$A8F is a code-digit game with $d_1 = 10, d_2 = 8$ and $d_3 = 15$.

Not surprisingly, the question to extend the analysis of hexadecimal games listed as the third problem in all Guy's lists of unsolved problems in combinatorial games since 1991.

***Open Problem* 7.** *[40, 9, problem 2] Examine some hexadecimal games. Obtain conditions for arithmetic periodicity.*

It is believed that all octal games are eventually periodic, so in other words, we believe that a relatively simple solution always exists in the form of the finite description of the periodic sequence. By allowing splitting the remainder into not two, but three piles, things get unexpectedly complicated.

Many of these games exhibit the *arithmetic periodicity*: there exists integers n_0, p and s (the pre-period and period lengths and the saltus) such that for all $n \geqslant n_0$ is $\mathcal{G}(n+p) = \mathcal{G}(n) + s$.

The following proposition explains why is this periodicity common for a subclass of hexadecimal games.

Proposition 3.84. [45, l. 1] Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3 \ldots \mathsf{d}_\mathsf{k}$ be a hexadecimal game for which exist odd i and even j such that $\mathsf{d}_i, \mathsf{d}_j \geqslant 8$. Then

$$\lim_{n\to\infty} \mathcal{G}(n) = \infty.$$

Proof. Let $m = max(i, j)$ and without loss of generality assume that i is odd. We will show that for all $n \geqslant m + 2$ there is a move to all positions with $\mathcal{G}$-value $\mathcal{G}(m)$ such that $1 \leqslant m \leqslant n - m - 2$. Specifically, we prove that there always exists a positive number k such that $H_k + H_k + H_m$ is an option of H_n. Notice that this option has $\mathcal{G}$-value $\mathcal{G}(k) \oplus \mathcal{G}(k) \oplus \mathcal{G}(m) = \mathcal{G}(m)$.

Let n is odd and consider the following cases.

(a) Let m is odd. Then since j is even, $n-j-m$ is even and also $n-j-m \geqslant 2$, so we can subtract j and then put $k = n - m - j$ to get the required move.

(b) Let m is even. Then since i is odd, $n-i-m$ is even and also $n-i-m \geqslant 2$, so we can subtract i and then put $k = n - m - i$ to get the required move

If n is even, the argument is the same by symmetry. □

The above Proposition has the consequence that we know for sure that the attempt at using Theorem 3.73 for periodicity testing will always fail. In 1976, Austin developed a systematic method for detecting the arithmetic periodicity [4]. His approach is similar to the simple periodicity testing method described at the beginning of this section.

However, his test was able to detect only the periodicity with the saltus of the form 2^k. It would be sufficient if Guy's conjecture that all hexadecimal games exhibit only a saltus of power of two was true. This conjecture was disproved by Jack Kenyon in 1967 [40, p. 43] with a *Kenyon's game* ·3F which is arithmetic periodic with $p = 6$ and odd saltus $s = 3$:

$$\mathcal{G}\text{-sequence}(\cdot\mathsf{3F}) = 012\ 012\ 345\ 345\ 678\ 678\ldots = \overline{(012)^2}\ (+3).$$

Howse and Nowakowski were able to slightly extend the arithmetic periodicity test to games with odd saltus [45]. The test still applies only to certain restricted circumstances and involve computing an enormous number of $\mathcal{G}$-values to verify relatively small periods.

Theorem 3.85 (Arithmetic-periodicity theorem, [45])**.** Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots\mathsf{d}_\mathsf{k}$ be a finite hexadecimal game. Let exist integers $n_0, 3p \geqslant u{+}2$ and $s = 2^{\gamma-1}{+}j$, where $j < 2^{\gamma-1}$ such that

(a) $\mathcal{G}(n+p) - \mathcal{G}(n+s)$ for all n in range:

$$n_0 < n < t + max\big(3n_0 + p(20 + 2^{\gamma+1} + 2^{2\gamma+1-k} + 2^{3\gamma+3-2k}),\\ n_0 + p(1 + 2^{j+k+3-\gamma})\big),$$

(b) $\mathcal{G}(n) < s$ for all $n \leqslant n_0$,

(c) $\mathcal{G}(n) < 2s$ for all $n \leqslant n_0 + p$,

(d) either there exist d_{2v+1}, d_{2v}, both of which contain 8, and for each $g, 0 \leqslant g < 2s$, there exists $n > 0$ such that $\mathcal{G}(n) = g$, or there exists d_t which contains 8, and for each g, $0 \leqslant g < 2s$, there exists $2v+1$, $2w \geqslant 0$ such that $\mathcal{G}(2v+1) = \mathcal{G}(2w) = g$.

Then for all $n > n_0$, $\mathcal{G}(n+p) = \mathcal{G}(n) + s$.

3.5.1 Computational Results on Hexadecimal Games

Guy et al. states a few single and two digit hexadecimal games in [42, WW, p. 117]. In [55, GONC1, p. 162], several trivial examples are listed: ·8 is a first cousin of the TRIPLICATE NIM: the infinite subtraction game $\cdot\overline{003}\ldots$ with the nim-sequence 000 (+1). ·A, ·B, ·D and ·F all are SHE-LOVES-ME, SHE-LOVES-ME-NOT and ·1A and ·1B are ultimately arithmetic periodic with the period 2 and saltus 1.

Howse has calculated the first 1,500 $\mathcal{G}$-values for each of the 1-, 2- and 3-digit games []. For example, ·XY, where X and Y are each A, B, E or F and ·E8, ·E9, ·EC and ·ED are each equivalent to Nim. ·0A, ·0B, ·0E, ·0F, ·1A, ·1B, ·48, ·4A, ·4C, ·4E, ·82, ·8A, ·8E and ·CZ, where Z is any even digit, are equivalent to Duplicate Nim, while ·0C, ·80, ·84, ·88, and ·8C are like Triplicate Nim. Some games displayed ordinary periodicity; ·A2, ·A3, ·A6, ·A7, ·B2, ·B3, ·B7 have period 4, and ·81, ·85, ·A0, ·A1, ·A4, ·A5, ·B0, ·B1, ·B5, ·D0, ·F0, ·F1 are all SHE-LOVES-ME, SHE-LOVES-ME-NOT. ·9E, ·9F, ·BC, ·C9, ·CB, ·CD and ·CF have (apparent ultimate) period 3 and saltus 2; ·89, ·8D, ·A8, ·A9, ·AC, ·AD each have period 4 and saltus 2, while ·8B, ·8F and ·9B have period 7 and saltus 4.

With the new arithmetic periodicity testing theorem, Howse and Nowakowski have showed many interesting hexadecimal games with longer periods [45]. Some interesting examples are ·28=·29 with the longest known period 53 and saltus 16 and the only exceptional value being the zero $\mathcal{G}(0) = 0$. The longest known pre-period has the game ·9C with $n_0 = 28$, $p = 36$ and $s = 16$. The compendium of other interesting arithmetic periodic games can be found in [45, table 3].

The general analysis of hexadecimal games would seem feasible if we could be sure that all of them are arithmetic-periodic. However, this is not the case. There are a lot of of examples that exhibit completely different kinds of periodicity [57, A3].

For instance, the game ·123456789 is arithmetic periodic with $n_0 = 15, p = 2$ and $s = 1$ with infinite number of exceptional values that occur in a geometric fashion: $\mathcal{G}(2^k + 6) = 2^k - 1$. The game ·205200C is arithmetic periodic with $n_0 = 4$, $p = 40$ and saltus 16 except that $40k + 19$ has nim-value 6 and $40k + 39$ has nim-value 14. Horrocks and Nowakowski reported that the game ·660060008 exhibit similar regularity with period length of approximately 300,000 [57, A3].

Grossman and Nowakowski [35] presented a new type of regularity in the games ·200...0048 with an odd number of zero digits. This regularity can be described by so called "ruler function". We say that a sequence is **ruler regular** with saltus s and ruler parameter r if there is a finite set of integers

S such that $\mathcal{G}(n+p) = \mathcal{G}(n) + s$ except all n such that

$$n = P(q+1, m) + k, \quad \text{for all } k \in S,\ q, m \geqslant 0,$$
$$\text{where } P(q,m) := r(q \cdot 2^{m+1} + 2^m + 1).$$

For these values we have

$$\mathcal{G}\left(P(q+1,m)+k\right) \ = \ \mathcal{G}\left(P(q,m)+k\right) + 2^{m+3}.$$

Another interesting game is $\cdot\mathsf{9}$ which has not been completely analyzed, but the heaps up to $100,000$ exhibit a remarkable fractal-like set of nim-values [57, A3].

We end this section with a summary of unsolved two-digit hexadecimal games.

***Open Problem* 8.** *[48, GONC5, A3] Resolve any of the following unsolved hexadecimal games:*

$$\begin{array}{ll}
\cdot\mathsf{1X},\ X \in \{\mathsf{8,9,C,D,E,F}\}, & \cdot\mathsf{2X},\ \mathsf{A} \leqslant \mathsf{X} \leqslant F, \\
\cdot\mathsf{3X},\ 8 \leqslant \mathsf{X} \leqslant \mathsf{E}, & \cdot\mathsf{4X},\ \mathsf{X} \in \{\mathsf{9,B,D,F}\}, \\
\cdot\mathsf{5X},\ 8 \leqslant \mathsf{X} \leqslant \mathsf{F}, & \cdot\mathsf{6X},\ 8 \leqslant \mathsf{X} \leqslant \mathsf{F}, \\
\cdot\mathsf{7X},\ 8 \leqslant \mathsf{X} \leqslant \mathsf{F}, & \cdot\mathsf{9X},\ 1 \leqslant \mathsf{X} \leqslant \mathsf{A}, \\
\cdot\mathsf{9D}, & \cdot\mathsf{BX},\ \mathsf{X} \in \{\mathsf{6,9,D}\}, \\
\cdot\mathsf{DX},\ 1 \leqslant \mathsf{X} \leqslant \mathsf{F}, & \cdot\mathsf{FX},\ \mathsf{X} \in \{\mathsf{4,6,7}\}.
\end{array}$$

3.6 Pure Breaking Games

In view of the above results, we can notice there is a great gap between understanding octal and hexadecimal games. It appears that the complexity of the nim sequences has a tendency to increase while we increase the number of possible splits of a heap. In this section, we will try examine this tendency in more detail. This can be done by disallowing the removing tokens from a heap in order to examine these games in an isolation.

This class of so called pure breaking games was first described and analyzed in separation by Dailly et al. in 2018 [11]. They define these games as follows.

Definition 3.86. Let A be a set of positive integers, called the *cut numbers*. We define a Pure Breaking Game $\mathcal{PB}(L)$ as the heap game with the following options

$$H_n := \{H_{a_1} + \ldots + H_{a_k}\},$$

such that $\ell \in L$ and $a_0 + a_1 + \ldots + a_\ell = n$ with $a_i > 0$ for all $i \geqslant 0$. In other words, in $\mathcal{PB}(L)$ a player's move consist of choosing a heap and splitting it into $\ell + 1$ non-empty heaps with $\ell \in L$ being his cut number. We will call this move an *ℓ-cut*.

Table 3.3: Solution for particular pure breaking games introduced in [11].

Game	**Sequence**	**Period**
$\{\ell_1, \ell_2, \ldots\}$ $(\ell_1 \geqslant 2)$	$(0)^{\ell_1}(+1)$	ℓ_1
$\{1, \ell_2, \ldots\}$ (ℓ_i odd)	$(0, 1)$	2
$\{1, 2, 3\ell_4, \ldots, \ell_k\}$	$(0)(+1)$	1
$\{1, 2, 3k\}$ $(k \geqslant 1)$	$(0, 1)^k(+2)$	$2k$

Lemma 3.87. Let $\mathcal{PB}(L)$ be a pure breaking game such that L contains some even integer and let us denote m the smallest one. Then for each $a, b \geqslant 0$ such that $\mathcal{G}(a) = \mathcal{G}(b)$ we have $a \not\equiv b \pmod{m}$.

Proof. For contradiction, let us assume $\mathcal{G}(a) = \mathcal{G}(b)$ and $a = k_1 m + d, b = k_2 m + d$ for some $k_1 > k_2 \geqslant 0$ and $0 \leqslant d < m$. Then from H_b there is a m-cut to $H_a + H_{k_1-k_2} + H_{k_1-k_2} + \ldots + H_{k_1-k_2}$. But since m is even, the terms $H_{k_1-k_2}$ will cancel each other and H_b has a move to $\mathcal{G}(a)$ which results in $\mathcal{G}(a) \neq \mathcal{G}(b)$ contradicting the assumption. □

Definition 3.88. The pure breaking game $\mathcal{PB}(\ell_1, \ldots, \ell_k)$ **verifies** the power-two AP-test(p, s) for some $p \geqslant 1$ and $s = 2^k$ for some $k \geqslant 1$ if the following holds.

(a) for $n \leqslant 3p, \mathcal{G}(n + p = \mathcal{G}(n) + s$,

(b) $\{\mathcal{G}(n) : 1 \leqslant n \leqslant p\} = \{0, \ldots, s - 1\}$, and

(c) for all n in $[3p + 1, \ldots, 4p]$ and for all g in $[0, \ldots, s - 1]$, there is a move from H_n to g using an m-cut with $m \geqslant 2, m \in L$.

Theorem 3.89 (Pure Breaking Power-two AP-test, [11, th. 9])**.** The pure breaking game $\mathcal{PB}(\ell_1, \ldots, \ell_k)$ which verifies the power-two AP-test(p, s) with $s = 2^k$ for some $k \geqslant 1$ is arithmetic periodic with pre-period 1, period p and saltus s.

3.7 Partizan Code-Digit Games

A duel heap game, as described in Section 3.1.4, are straightforward generalizations of impartial heap games in which each player is assigned an individual impartial ruleset [28]. We remind that for rulesets Γ and Λ we say that $\Gamma \prec \Lambda$ *(Γ dominates Λ)* if for all large enough n, Γ can always beat Λ (see Definition 3.7).

Here we will survey the results on duels of code-digit games.

3.7.1 Partizan Subtraction Games

Definition 3.90. Let $\mathcal{S}_L$ and $\mathcal{S}_R$ be sets of integers. We call the duel of these subtraction games $\mathcal{S}_L$ **versus** $\mathcal{S}_R$ a **partizan subtraction game**, denoted $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$.

The following proposition shows that the dominance property unfortunately does not define a partial order.

Proposition 3.91. [28, ch. 2] The relation $\prec$ on code-digit games is not transitive. In other words, there exist code-digit games Γ, Λ and Υ such that $\Gamma \prec \Lambda \prec \Upsilon \prec \Gamma$.

Proof. This is actually true already for subtraction games. The subtraction games that satisfy this property are $\Gamma = \mathcal{S}_1(1,2,6)$, $\Lambda = \mathcal{S}_2(1,3,5)$ and $\Upsilon = \mathcal{S}_3(2,3,4)$. The proof of the outcome for large enough heap is a simple exercise. □

In Theorem 3.27 we have seen that all impartial finite subtraction games have periodic nim-sequences. For partizan subtraction games holds only a restricted version of this Theorem, where we focus only on the periodicity of the outcome sequence. So this theorem is applicable only to single-heap games.

Theorem 3.92. [28, th. 1] Every finite partizan subtraction game has a periodic outcome sequence.

Proof. Let us denote the $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$-heap of size n as H_n. Let m be the maximal number in $\mathcal{S}_L \cup \mathcal{S}_R$. There are at most 4^m distinct blocks of length m. So there exist integers n_0 and p such that $o(H_n) = o(H_{n+p})$ for all $n \in \{n_0, n_0+1, \ldots, n_0+m-1\}$.

Now, since the outcome of any position depends solely on the sets of outcomes both player can move to and we can observe that any two positions $n_0+m-1+j$ and $n_0+m-1+j+c$ with $1 \leqslant j \leqslant m$ have these sets of outcomes identical, this implies that $o(H_n) = o(H_{n+p})$ for all $n \geqslant n_0$. □

Corollary 3.93. We can solve any single-heap finite subtraction game by calculating the periodicity of its outcome sequence.

Observation 3.94. Let $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$ be a partial subtraction game which has aperiodic outcome sequence, or the period is greater than one. Then $\mathcal{S}_L \parallel \mathcal{S}_R$.

Now let us consider two particular examples of partizan subtraction games: $\mathcal{S}(1, 2n+1 \mid 1, 2m)$ and $\mathcal{S}(1, 2n \mid 1, 2m)$.

Proposition 3.95. [28, th. 2] Let $\mathcal{S}_L = (1, 2n)$, $\mathcal{S}_R = (1, 2m+1)$, $1 \leqslant m, n$. Then $\mathcal{S}_L \prec \mathcal{S}_R$.

Proposition 3.96. [28, th. 3] Let $\mathcal{S}_L = (1, 2n)$, $\mathcal{S}_R = (1, 2m)$, $1 \leqslant n < m$. Then $\mathcal{S}_L \succ \mathcal{S}_R$.

The following theorem generalizes the property of impartial symmetric subtraction games described in Section 3.3.6.

Theorem 3.97 (Symmetric Partizan Subtraction Game, [28, th. 4])**.** Consider a partizan subtraction game with the following sets

$$\begin{aligned} \mathcal{S}_L &= (x_1, \ldots, x_n : 1 \leqslant x_1 < \ldots < x_n) \text{ and} \\ \mathcal{S}_R &= (y_1, \ldots, y_n : 1 \leqslant y_1 < \ldots < y_n). \end{aligned}$$

Suppose there exists a non-negative integer r (the modulus) such that $x_i + y_{n-i+1} = r$ for all $i = 1, \ldots, n$. Then the outcome sequence of partial game $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$ is purely periodic with period r and $\mathcal{S}_L \parallel \mathcal{S}_R$.

Fraenkel and Kotzig also describe interesting property of the subtraction games which are strictly periodic with period length being a sum of the minimal elements of both sets.

Theorem 3.98. Let $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$ be a purely periodic partizan subtraction game with period $p = l + r$ such that $l = \mathit{min}(\mathcal{S}_L)$ and $r = \mathit{min}(\mathcal{S}_R)$. Then for all $s \in \mathcal{S}_L$ holds $s \equiv l \bmod p$ and for all $s' \in \mathcal{S}_R$ holds $s' \equiv r \bmod p$.

Similarly, the expansion set defined in 3.38 can also be generalized to denote the set of moves that can be adjoined to either of the expansion sets without changing the outcome sequence of the single-heap game.

Canonical Forms of Partizan Subtraction Games

There are no published results on the canonical forms of these games. Even though these are one of the simplest partizan games, the canonical form sequences show a surprising complexity. In his notes on these games, Plambeck [61] reports canonical sequences of several partizan subtraction games that do contain some periodicity. He conjectures the following behavior of the game $\mathcal{S}(1, 2 \mid 1, 3)$:

Conjecture 7. *[61, ch. 3] The canonical forms of the game $\mathcal{S}(1, 2 \mid 1, 3)$ satisfies the recursion*

$$\mathcal{C}(n) = \{0 \mid \mathcal{C}(n-3)\}$$

with $\mathcal{C}(0) = 0, \mathcal{C}(1) = *$ and $\mathcal{C}(2) = \uparrow$.

3.7.2 Partizan Code-Digit Games

Definition 3.99. We define a **partizan code-digit** game using a pair of code-digit rulesets $(\mathsf{D}^L, \mathsf{D}^R)$. This game is played as a duel heap game D^L **versus** D^R.

Mesdal et al. gave some results on the $\mathcal{C}$-sequences of the game called PARTIZAN SPLITTLES. These are partizan code-digit (octal) games of form $0{\cdot}\mathsf{d}_1\mathsf{d}_2\ldots\mathsf{d}_\mathsf{k}$) with $\mathsf{d}_i \in \{0,7\}$. In these games a move consist of removing some tokens from one heap and optionally splitting the remainder into two heaps.

Partizan Splittles

Mesdal studied a subclass of code-digit games which he named the PARTIZAN SPLITTLES. In these games, players can in their moves remove some tokens from a pile and then *optionally* splitting the remaining heap into two. He investigated the canonical forms of positions in this game under a duel partizan setting.

Definition 3.100. The game of partizan splittles, denoted $\mathcal{SP}(\mathcal{S}_L \mid \mathcal{S}_R)$ is a partizan code-digit game $(\mathsf{D}^L, \mathsf{D}^R)$ with

$$\mathsf{d}_i^L = \begin{cases} 7 & \text{if } i \in \mathcal{S}_L \\ 0 & \text{otherwise.} \end{cases} \qquad \text{and} \qquad \mathsf{d}_i^R = \begin{cases} 7 & \text{if } i \in \mathcal{S}_R \\ 0 & \text{otherwise.} \end{cases}$$

The sets $\mathcal{S}_L, \mathcal{S}_R$ are called *subtraction sets* of Left and Right player, respectively.

Theorem 3.101. [54, th. 2] The game of partizan splittles $\mathcal{SP}(1 \mid k)$ has the the canonical form sequence defined by

$$H_n = \begin{cases} n & \text{if } n < k, \\ \{k-1 \mid 0\} & \text{if } n \geqslant k. \end{cases}$$

Definition 3.102. We define $\uparrow^2 = \{0 \mid \downarrow *\}$ and $\uparrow^{\rightarrow 2} = \uparrow + \uparrow^2 = \{\uparrow \mid *\}$.

Theorem 3.103. [54, th. 3] The game of partizan splittles $\mathcal{SP}(1, 2k \mid 1, 2k+1)$ has the the canonical form sequence defined by

$$H_{4jk+i} = \begin{cases} j \cdot \uparrow^{\rightarrow 2} + 0 & \text{if } i \text{ is even and } 0 \leqslant i < 2k; \\ j \cdot \uparrow^{\rightarrow 2} + * & \text{if } i \text{ is odd and } 0 \leqslant i < 2k; \\ j \cdot \uparrow^{\rightarrow 2} + \uparrow & \text{if } i \text{ is even and } 2k \leqslant i < 4k; \\ j \cdot \uparrow^{\rightarrow 2} + \uparrow * & \text{if } i \text{ is odd and } 2k \leqslant i < 4k, \end{cases}$$

where $j \in \mathbb{N}_0$ and $0 \leqslant i < 4k$. So the sequence is arithmetic periodic with $p = 4k$ and the saltus $s = \uparrow^{\rightarrow 2}$.

Next Mesdal investigated the partizan splittles games $\mathcal{SP}(1, a_1, a_2, \ldots \mid 2, 4)$, where a_i is *odd* for each i. He notes that the canonical forms of positions in these games are too complicated and strongly depend on the exact values a_i. However, he noticed that the complexity of the canonical forms is caused only by infinitesimal perturbances, so he analyzed the reduced canonical form sequence of these games (introduced in Section 2.2.7).

Theorem 3.104. [54, ch. 7] Let us define $f(n)$ to be a the arithmetic-periodic sequence with period 4 and saltus $\frac{3}{4}$ defined by

$$f(n) = \begin{cases} 0, & \text{if } n = 0; \\ 1, & \text{if } n = 1; \\ 1/2, & \text{if } n = 2; \\ 3/2, & \text{if } n = 3; \\ f(n-4) + \frac{3}{4}, & \text{if } n \geqslant 4. \end{cases}$$

Then the game has the following reduced canonical form sequence

$$rcf\text{-sequence}(n) = f(n) \cdot 1*,$$

where $1* = 1 + *$ and $\cdot$ represents the *Norton multiplication* defined for this purpose as follows: for any game G is $G \cdot 1* := \left\{ G^L \cdot 1* + 1 \,\middle|\, \mathcal{G}^R \cdot 1* - 1 \right\}$.

Up to our knowledge, the partizan subtraction games and the partizan splittles are the only code-digit games that have been studied under the partizan setting.

Chapter **4**

Our Results

In this section, we present our contribution to the theory presented in the previous chapter. First three sections of this Chapter describe our results in the study of subtraction games, impartial taking and breaking games and partizan taking and breaking games. The last section presents our results in algorithmic combinatorial game theory by providing new game solving algorithms and complexity results about a generalization of a subtraction game.

Usually, we used a computer-assisted method for achieving these results by utilizing game solving algorithms described in Section 4.4. We will describe this computer-assisted method on the following example:

Valla [64] asked the following question about the behavior of the outcome and nim-sequences of finite impartial subtraction games:

***Question* 8.** *[64] It is clear that if the nim-sequence of a subtraction game is periodic then the outcome sequence periodic as well. However, is there a subtraction game which has an aperiodic nim-sequence and a periodic outcome sequence?*

The answer to this natural question might not be as straightforward as it might seem at a first glance. Definitely, there is some limitation on the nim-sequence once we have fixed the periodicity of the outcome sequence. For instance, the aperiodicity of the nim-sequence can not be easily proven just by applying the usual argument that for any integer p we can adjoin some $\ell \cdot k$ to the subtraction set, because no multiple of the length of the outcome period can not appear in the subtraction set.

We performed an computing experiment where we examined all subtraction sets having maximal number less then 15. For each such subtraction game, we computed its nim-sequence and then we continued with the following procedure. We searched for any integer s such that there exists n such that $\mathcal{G}(n) = \mathcal{G}(n+s)$, but for each such n we have $\mathcal{G}(n) \neq 0$. This means that adjoining s will change the nim-sequence while the outcome sequence will stay the same. We then adjoined s and repeated this search while backtracking when no such s could be found. Then we looked for patterns in these consecutively adjoined numbers s. This experiment led to the result described in Theorem 4.4.

4.1 Results on Subtraction Games

4.1.0.1 Periodicity of the Outcome and Nim Sequences

***Conjecture* 9.** *Ho in [44, p.18] conjectured that for every subtraction game its nim-sequence and its outcome sequence have the same period p.*

Note: This is not true and there seems to be many counter-examples. For example, for the game $\mathcal{S}(1, 15, 23, 38, 39, 50, 81, 98)$ is

$$\mathcal{G}\text{-sequence}(\mathcal{S}) = \ldots \overline{01230132}$$

with the period length 8 (and pre-period $1,140$), while

$$o\text{-sequence}(\mathcal{S}) = \ldots \overline{\mathcal{N}\mathcal{N}\mathcal{N}\mathcal{P}}$$

with period length 4 and pre-period length $1,041$.

We remind Question 2 asked by Althöfer and discussed in Section 3.3.2:

Question. *[2, q. 4] The period of nim-sequence might be longer than the period of the outcome sequence. What are the extremal examples?*

This inspires also the following question:

***Question* 10.** *Is there an infinite number of examples of games, for which the nim-sequence and the outcome sequence have different period?*

Theorem 4.1. Let subtraction game $\mathcal{S}$ be periodic with (not necessarily least) period p. Let ℓ be the least period of $\mathcal{S}$. Then ℓ divides p.

Proof. Let $g = \mathit{gcd}(\ell, p)$, n_0 pre-period of period p, m_0 pre-period of period ℓ. We can rotate both periods to the right until they have the same pre-period N. Then the end positions of repeated shorter period will fall onto the following positions of the longer period: $\{k\ell \bmod p : k \in \mathbb{N}\} = \{0, g, 2g, \ldots, p - g\}$, so the sequence is also periodic with $g < \ell$, a contradiction. $\square$

Theorem 4.2. For periodic subtraction games, the period length of the outcome sequence divides the period length of the nim-sequence.

Proof. Clearly, when nim-sequence has period p, the outcome sequence is also periodic with period p. Since all nim values in the period cannot be same, the period of outcome sequence must divide p. □

Corollary 4.3. Let $\mathcal{S}$ be periodic subtraction game with prime period p. Then the the outcome sequence has also period p.

The following subtraction game is designed to provide the answers to Questions 2, 8, and 10:

Theorem 4.4. There exists an infinite subtraction game with any of the following properties:

(a) The outcome sequence is periodic and the nim-sequence is aperiodic *(Q. 8)*.

(b) Both the outcome sequence and the nim-sequence periodic with different period; provide an infinite class of such games *(Q. 10)*.

(c) For any fixed integer $c > 1$ there exists a subtraction game such that if its outcome sequence has period p, than its nim-sequence has (minimal) period $c \cdot p$ *(Q. 2)*.

Proof. We will consider subtraction games with the subtraction set

$$\mathcal{S} = \{1, 4, 10\} \cup \{1 + 11k : k \in K\}, \text{ with some } K \subset \mathbb{N}.$$

The following table shows the nim-sequence of the game $\mathcal{S}(1, 4, 10)$ which is periodic with pre-period $n_0 = 16$ and period $p = 11$ (the last exceptional value $\mathcal{G}(15) = 3$ is displayed in a box). Notice that the outcome-period and the nim-period are equal because the period is prime (Corollary 4.3).

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0+	0	1	0	1	2	0	1	0	1	2	3
11+	**2**	3	0	1	$\boxed{\mathbf{3}}$	0	1	0	1	2	0
22+	1	2	0	1	2	0	1	0	1	2	0
33+	1	2	0	1	2	0	1	0	1	2	0
44+	1	2	0	1	2			...			

Table 4.1: The nim-sequence of the game $\mathcal{S}(1, 4, 10)$.

Now, we can observe that the only possible way to $\mathcal{G}(n) = \mathcal{G}(n + 11k)$ for any $k \geqslant 1$ is when $n = 11k$ (it is enough to look for the same $\mathcal{G}$-values in neighboring columns (modulo 11) of the table 4.1). Notice how the sequence changes after adjoining some $1 + 11k$ with $k \geqslant 1$. The value $\mathcal{G}(10 + 11k)$ will change from 2 to 3, since there will be additional option to go to $\mathcal{G}(11) = 2$

(in Table highlighted in bold). But the observation above ensures us that no other $\mathcal{G}$-value will change. Also note that this does not have any effect on the consequence of the effect of adjoining some other $1+11k'$ with $k' \neq k$, because the value $\mathcal{G}(1+11k)$ for $k \geqslant 1$ does not actually affect any other $\mathcal{G}$-value (this can be verified by observing that $\mathcal{G}(1+11k) > \mathcal{G}(1+11k+s)$ for any $s \in \mathcal{S}$) and therefore we can safely increment it without consequences.

Now, let us use these properties of this game in proving the statements above. For any subset of integers K we will denote a *K-game* the subtraction game with a set $\{1, 4, 10\} \cup \{1 + 11k : k \in K\}$.

(a) To construct an aperiodic nim-sequence, it suffices to show that the sequence $\{\mathcal{G}(11n + 1) : n \in \mathbb{N}\}$ is aperiodic. But to achieve such aperiodicity, it is enough to choose $K = \{\lfloor k\sqrt{2} \rfloor\}$. Since $\sqrt{2}$ is irrational, this forms an aperiodic sequence and so the K-game is aperiodic as well. Also note that based on the observations above, the only values that we have changed are $\mathcal{G}(1 + 11k)$, so the kernel remains unchanged and so the outcome sequence still remains periodic with period 11.

(b) It is even easier to introduce different periodicity to the nim-sequence. Let us put $K = \{2n : n \in \mathbb{N}\}$. This changes the above sequence by setting $\mathcal{G}(n) = 3$ for $n = 34 + 22k$, so the resulting sequence is periodic with pre-period $n_0 = 16$ and period $p = 22$.

(c) It suffices to set $K = \{c \cdot n : n \in \mathbb{N}\}$. The resulting K-game will have the *o*-period p and $\mathcal{G}$-period $c \cdot p$ by the same argument as above.

□

4.1.1 Canonical Subtraction Games

In this section, we attempt to extend the theory of equivalences among subtraction games presented in Section 3.3.3. Let us start with a motivational example.

Example 4.5. The subtraction games $\mathcal{S}(2, 3, 5)$ and $\mathcal{S}(2, 4, 5)$ both have the same following nim-sequence periodic with period $p = 7$:

n	0	1	2	3	4	5	6	7	8	9	...
$\mathcal{G}(\mathbf{n})$	0	0	1	1	2	2	3	0	0	1	...

The natural question to ask is, when do these equivalences occur? Note that the question of equivalence of two subtraction games $\mathcal{S}$ and $\mathcal{T}$ can be reformulated as the question whether $\mathcal{S}(n) + \mathcal{T}(n) \in \mathcal{P}$ for all $n \in \mathbb{N}_0$. Nevertheless, the simplest way to find the strategy for the second player on this sum is probably still to calculate the nim-sequences.

However, as the survey of results on subtraction games in Chapter 3.3 suggests, the behavior of nim-sequences are still quite mysterious to us and thus do not represent a good tool to analyze these relationships among the nim-sequences.

We will argue that the expansion sets discussed in Chapter 3.3.3 might be more suitable to answer these kind of questions.

Definition 4.6. Let $\mathcal{S}, \mathcal{T}$ be subtraction games. We say that they **equal**, denoted $\mathcal{S} = \mathcal{T}$, if for all $n \in \mathbb{N}_0$ is $\mathcal{G}_{\mathcal{S}}(n) = \mathcal{G}_{\mathcal{T}}(n)$. Furthermore, if the subtraction sets $\mathcal{S}$ and $\mathcal{T}$ are the same, we say they are **identical**, denoted $\mathcal{S} \cong \mathcal{T}$.

Lemma 4.7. The relation $=$ on subtraction games is an equivalence.

Proof. Follows directly from the fact that $=$ on games is an equivalence. □

Theorem 4.8. Let $\mathcal{S}, \mathcal{T}$ be subtraction games. Then

$$\mathcal{S} = \mathcal{T} \text{ if and only if } \mathit{exp}(\mathcal{S}) = \mathit{exp}(\mathcal{T}).$$

Proof. First suppose that $\mathcal{S}' = \mathit{exp}(\mathcal{S}) = \mathit{exp}(\mathcal{T})$. Then the nim-sequence generated by the subtraction set $\mathcal{S}'$ is clearly unique, so $\mathcal{S} = \mathcal{T}$.

Conversely, suppose that $\mathcal{S} = \mathcal{T}$, so the nim-sequence are equal. But the sets $\mathit{exp}(\mathcal{S})$ and $\mathit{exp}(\mathcal{T})$ are defined solely by the nim-sequences of $\mathcal{S}$ and $\mathcal{T}$ (see Definition 3.38), so if they are equal, the expansion sets must be equal as well. □

Example 4.9. The two games from the Example 4.5 both have the same expansion set:

$$\mathit{exp}\{2,3,5\} = \mathit{exp}\{2,4,5\} = \{2,3,4,5\}^{*7}.$$

Unfortunately, this property does not help us much with the testing if the games are equal because in order to calculate the expansion set we do need to have calculated the nim-sequences in the first place. Nevertheless, the structure of the equivalence classes among subtraction sets could provide some insight about the equal subtraction games.

Lemma 4.10. Any equivalence class on subtraction games is a *directed* set ordered by inclusion.

Proof. Let $\mathcal{S}, \mathcal{T}$ be subtraction games such that $\mathcal{S} = \mathcal{T}$. Consider the subtraction game $\mathcal{S}' = \mathcal{S} \cup \mathcal{T}$. Since $\mathcal{S} = \mathcal{T}$, by definition of exp we have that $\mathcal{S} \subseteq \mathit{exp}(\mathcal{T})$ and $\mathcal{T} \subseteq \mathit{exp}(\mathcal{S})$ so clearly $\mathcal{S}' = \mathcal{S} = \mathcal{T}$. □

Definition 4.11. We will call the maximal element of the equivalence class $[\mathcal{S}]_=$ of some subtraction game $\mathcal{S}$ the **master**, denoted $\mathcal{M}(\mathcal{S}) := exp(\mathcal{S})$. The minimal elements will be called the **minimals**, denoted $m(\mathcal{S})$. Formally,

$$m(\mathcal{S}) := \{\mathcal{S}' : \mathcal{S}' = \mathcal{S} \text{ and there is no } \mathcal{T} \text{ such that } \mathcal{T} \subset \mathcal{S}'\}.$$

Example 4.12. The games $\mathcal{S}(2,3,5) = \mathcal{S}(2,4,5)$ are minimals of their equivalence class and the game $\mathcal{S}(2,3,4,5)$ is the master.

Theorem 4.13. Let $\mathcal{S}$ be a subtraction game. The minimals $m(\mathcal{S})$ along with their master $\mathcal{M}(\mathcal{S})$ provide sufficient information to determine if some subtraction game $\mathcal{T}$ equals $\mathcal{S}$ without calculating their nim-sequences.

Proof. We will show that $\mathcal{S} = \mathcal{T}$ if and only if $\mathcal{S}' \subseteq \mathcal{T} \subseteq \mathcal{M}(\mathcal{S})$ for some minimal $\mathcal{S}' \in m(\mathcal{S})$.

($\Leftarrow$) Since $\mathcal{S}' \in m(\mathcal{S})$ and $\mathcal{T} \subseteq exp(\mathcal{S})$, we have that $\mathcal{S}' = \mathcal{S}$ and $\mathcal{T}\backslash\mathcal{S}' \subseteq exp(\mathcal{S})$, so by definition of the expansion set it follows that $\mathcal{S} = \mathcal{T}$.

($\Rightarrow$) Since $\mathcal{S} = \mathcal{T}$, we have that $m(\mathcal{S}) = m(\mathcal{T})$ and it follows that there must exist a minimal $\mathcal{S}' \in m(\mathcal{S})$ such that $\mathcal{S}' \subseteq \mathcal{T}$. Furthermore, $exp(\mathcal{S}) = exp(\mathcal{T})$, so certainly $\mathcal{T} \subseteq exp(\mathcal{S})$.

$\square$

Corollary 4.14. Let $\mathcal{S}$ be a subtraction game and $[\mathcal{S}]_=$ be its equivalence class. Then

$$[\mathcal{S}]_= = \bigcup_{\mathcal{S}' \in m(\mathcal{S})} \{\mathcal{S}' \cup \mathcal{E} : \mathcal{E} \subset exp(\mathcal{S})\backslash\mathcal{S}'\}.$$

We will now focus on the problem of finding the minimals of a subtraction game.

Let $S(s_1, s_2, s_3, \ldots)$ be a subtraction game. For each position $n \in \mathbb{N}_n$ and for each $\mathcal{G}$-value $g \in \{0, 1, ..., \mathcal{G}(n) - 1\}$ we define a g-**restriction** for $\mathcal{S}$ as follows:

$$R(g, n) := \{s : \mathcal{G}(n - s) = g, s \in exp(\mathcal{S}), s \geqslant n\}.$$

We denote a collection of all such restrictions as $\mathcal{R}(\mathcal{S})$:

$$\mathcal{R}(\mathcal{S}) := \big\{R(g, n) : n \in \mathbb{N}_0, g \in \{0, 1, ..., \mathcal{G}(n) - 1\}\big\}.$$

Definition 4.15. Let U be a set and let $\mathcal{F}$ be a family of sets over U, $F \subset \mathcal{P}(U)$. We say that a set $S \subseteq U$ is a **hitting set** for F, if for each $A \in F$ is

$$S \cap A \neq \varnothing.$$

We say that an element $x \in S$ did **hit** the set A if $x \in A$.

Theorem 4.16. Let $\mathcal{S}$ be a subtraction game and $\mathcal{R}(\mathcal{S})$ a set of its restrictions. Then for each subtraction game $\mathcal{T}$ holds that

$$\mathcal{T} = \mathcal{S} \text{ if and only if } \mathcal{T} \text{ is a hitting set for } R(\mathcal{S}).$$

Proof. Let $\mathcal{T} = \mathcal{S}$ and for contradiction assume that $\mathcal{R}(g, n)$ is the restriction that was not hit by any element from $\mathcal{T}$. Assume that n and g are least such integers. Consider the position $\mathcal{G}_{\mathcal{T}}(n)$. Note that since $\mathcal{T} = \mathcal{S}$, we have that $\mathcal{G}_{\mathcal{T}}(m) = \mathcal{G}_{\mathcal{S}}(m)$ for each m. But since $R(g, n)$ was not hit, it follows by minimality of n and g that $\mathcal{G}_{\mathcal{T}}(n) = \mathit{mex}\{\mathcal{G}_{\mathcal{T}}(n - s) : s \in \mathcal{T}\} = g < \mathcal{G}_{\mathcal{S}}(n)$, a contradiction.

Conversely, assume that $\mathcal{T}$ is a hitting set for $R(S)$. By induction on n we will show that $\mathcal{G}_{\mathcal{T}}(n) = \mathcal{G}_{\mathcal{S}}(n)$. Let this be true for each $m < n$. We have that $\mathcal{G}_{\mathcal{T}}(n) = \mathit{mex}\{\mathcal{G}_{\mathcal{T}}(n - s) : s \in \mathcal{T}\}$. Since $\mathcal{T}$ is a hitting set, all $g \in \{0, 1, \ldots, \mathcal{G}(n) - 1\}$ are excluded. Furthermore, the value $\mathcal{G}(n)$ is not excluded, otherwise there would exist some $s \in \mathit{exp}(\mathcal{S})$ such that $\mathcal{G}_{\mathcal{S}}(n - s) = \mathcal{G}_{\mathcal{S}}(n)$, but since that is not allowed by the definition of the expansion set, the element s does not belong to the universe of the hitting set and thus cannot $s \in \mathcal{T}$. □

Corollary 4.17. A subtraction set $\mathcal{S}$ is a *minimal* if and only if it is a minimal (in inclusion) hitting set of $\mathcal{R}(\mathcal{S})$.

We have seen that some of the infinite subtraction games are periodic. The game $\mathcal{S}(\mathit{primes} \cup \{1\})$ is one of them. However, it has a finite minimal: $\mathcal{S}(1, 2, 3)$. In other words, the game is in fact not infinite, only its expansion set is (one can play the game using only a finite subset of its moves). Later in this section we will show that $\mathcal{S}(1, 2, 3)$ is actually the only minimal of its equivalence class, so the class forms a lattice (Theorem 4.32).

Now we could ask be asking the following question: is there an *infinite* periodic subtraction game with the property, that if we removed any element from the subtraction set, its nim-sequence would change. We answer this question immediately by the following observation.

Observation 4.18. There is a periodic subtraction game with an infinite minimal.

Proof. Consider again the game from Theorem 4.4 with the subtraction set $\mathcal{S} = \{1, 4, 10\} \cup \{1 + 11k : k \in K\}$, with some $K \subset \mathbb{N}$. If we put $K = \mathbb{N}$, the resulting K-game will be periodic with pre-period $n_0 = 16$ and period $p = 11$, because $\mathcal{G}(n) = 3$ for $n = 23 + 11k$. Also by removing any $k \in K$ the pre-period will change, because suddenly $\mathcal{G}(23 + 11k)$ would equal 2. □

This proof finishes our description of the equivalence class of subtraction games. We will follow up with several examples of application of this theory.

4.1.2 m-plicate Subtraction Games

In this section, we follow up on the theory described in Section 4.1.1 by describing a class of subtraction games which appear to have the largest size of their equivalence classes and also largest number of minimals.

The well known g.c.d. property (Theorem 3.35) tells us that in our analysis of subtraction games it suffices to consider subtraction sets whose members are relatively prime. This will show to be useful, for instance, when we try to find the period of given subtraction game. Theorem 3.29 tells us that it suffices to consider the nim-sequence of length $n_0 + p + s_k$. Now, for fixed n_0 and p and some subtraction game $S(s_1, s_2, \ldots, s_k)$ with $s_k \gg n_0 + p$ and $m = \mathit{gcd}(\mathcal{S})$ we can consider almost m times shorter nim sequence for the game $\mathcal{S}/g$, a considerable improvement.

However, the following example shows that such m-plicate games can not be always detected by simply taking the g.c.d of the subtraction set.

Example 4.19. The subtraction game $\mathcal{S} = \{2, 3, 4\}$ is the duplicate of the game $\mathcal{T} = \{1, 2\}$:

n	0	1	2	3	4	5	6	7	...
$\mathcal{G}_{\mathcal{T}}(\mathbf{n})$	0	1	2	0	1	2	0	1	...
$\mathcal{G}_{\mathcal{S}}(\mathbf{n})$	0	0	1	1	2	2	0	0	...

So let us provide an analysis of the expansion set of an m-plicate game.

Theorem 4.20. Let $\mathcal{S}$ be a subtraction game, m an integer and let us define the sets A and B as follows:

$$\begin{aligned} A =& \{m \cdot s : s \in \mathit{exp}(\mathcal{S})\}; \\ B =& \{ms + 1, ms + 2, \ldots, ms + (m - 1) : s, +1 \in \mathit{exp}(\mathcal{S})\}. \end{aligned}$$

Then the m-plicate game $m\mathcal{S}$ has the following expansion set:

$$\mathit{exp}(m\mathcal{S}) = A \cup B.$$

Proof. For each element $a \in A \cup B$ we will show that there is no integer n such that $\mathcal{G}(n) = \mathcal{G}(n + a)$. Consider the following cases:

(A) Consider $a, b \in \mathbb{N}_0$ such that $b - a = m \cdot s$ for some $s \in \mathit{exp}(\mathcal{S})$. For contradiction, suppose that $\mathcal{G}_{m\mathcal{S}}(a) = \mathcal{G}_{m\mathcal{S}}(b)$. Since the game is the m-plicate, we have that $\mathcal{G}_{\mathcal{S}}(\lfloor \frac{a}{m} \rfloor) = \mathcal{G}_{\mathcal{S}}(\lfloor \frac{b}{m} \rfloor)$. But since $b - a = m \cdot s$, it follows that $\lfloor \frac{b}{m} \rfloor - \lfloor \frac{a}{m} \rfloor = s \in \mathcal{S}$, so $\mathcal{G}(\lfloor \frac{a}{m} \rfloor) = \mathcal{G}(\lfloor \frac{a}{m} \rfloor + s)$, a contradiction.

(B) Consider $a, b \in \mathbb{N}_0$ such that $b - a = m \cdot s + i$ for some $i \in \{1, 2, \ldots, m-1\}$ and some $s \in \mathit{exp}(\mathcal{S})$. Since both $\mathcal{G}_{\mathcal{S}}(\lfloor \frac{b}{m} \rfloor), \mathcal{G}_{\mathcal{S}}(\lfloor \frac{b}{m} + 1 \rfloor) \neq \mathcal{G}_{\mathcal{S}}(\lfloor \frac{a}{m} \rfloor)$, by the m-plication we know that for all $j \in \{0, 1, 2, \ldots, 2m-1\}$ is

$$\mathcal{G}_{\mathcal{S}}\Big(\Big\lfloor \frac{a}{m} \Big\rfloor \cdot m + j\Big) \neq \mathcal{G}_{\mathcal{S}}\Big(\Big\lfloor \frac{a}{m} \Big\rfloor \cdot m + ms + j\Big).$$

By setting $j = a \bmod m + i$ ($j < 2m$) it follows that $\mathcal{G}_{m\mathcal{S}}(a) \neq \mathcal{G}_{m\mathcal{S}}(b)$.

It remains to show that the expansion set is complete. Consider any $s \notin A \cup B$. We will show that there is some $n \in \mathbb{N}_0$ such that $\mathcal{G}_{m\mathcal{S}}(n) = \mathcal{G}_{m\mathcal{S}}(n+s)$. For contradiction, let us assume that for all n is $\mathcal{G}_{m\mathcal{S}}(n) \neq \mathcal{G}_{m\mathcal{S}}(n+s)$. But by m-plication this implies that for all m is $\mathcal{G}_{\mathcal{S}}(m) \neq \mathcal{G}_{\mathcal{S}}(m + \lfloor \frac{s}{m} \rfloor)$, so $s \in \mathit{exp}(\mathcal{S})$, a contradiction. □

Definition 4.21. We will call the subtraction game $\mathcal{S}(1, 2, \ldots, k)$ the Bounded Nim(k) for obvious reasons. Also by applying Theorem 4.20, we will, call the game $\mathcal{S}(M, M+1, \ldots, M \cdot k)$ the M-plicate Bounded Nim(k).

Let us now consider the game $\mathcal{S}(2, 3, \ldots, k)$. We call it the Duplicate Bounded Nim(k) for some even $k \geqslant 2$. By Theorem 4.20 we know that this is the duplicate of the game $\mathcal{S}(1, 2, \ldots, \frac{k}{2})$, so it has the period $001122\ldots\frac{k}{2}\frac{k}{2}$. It is not difficult to see that for odd k the period becomes $001122\ldots\lfloor\frac{k}{2}\rfloor\lfloor\frac{k}{2}\rfloor\lceil\frac{k}{2}\rceil$. Furthermore, we cannot adjoin 1, so clearly

$$\mathit{exp}\{2, 3, \ldots, k\} = \{2, 3, \ldots, k\}^{*p},$$

where p is the length of the period. Hence $\mathcal{S}(2, 3, \ldots, k)$ is a master. Now let us consider the structure of its equivalence class.

Theorem 4.22. Let us denote the number of minimals of the equivalence class of the game $\mathcal{S}(2, 3, \ldots, k)$ with $k \geqslant 2$ as $\mathit{minCount}_2(k)$ and the size of the equivalence class as $\mathit{classSize}_2(k)$. Then these values are given by the recurrences

$$\begin{aligned} \mathit{minCount}_2(k) &= \mathit{minCount}_2(k-2) + \mathit{minCount}_2(k-3) \text{ and} \\ \mathit{classSize}_2(k) &= \mathit{classSize}_2(k-1) + \mathit{classSize}_2(k-2), \end{aligned}$$

with

$$\mathit{minCount}_2(2) = \mathit{minCount}_2(3) = \mathit{minCount}_2(4) = \mathit{classSize}_2(2) = \mathit{classSize}_2(3) = 1.$$

(Note that these sequences are shifted Padovan [75] and Fibonacci numbers [74]).

Proof. We will prove both recurrences by induction on k. For $k = 2$ the equivalence class has a single element so there is nothing to prove. Let the statements hold for the values less than k. We will consider the restrictions for the game $\mathcal{S}(2, \ldots, k)$. Clearly, $R(g, 0) = R(g, 1) = \varnothing$ for any g and

$R(g, 2) = \{2\}$, so any game $\mathcal{T}$ such that $\mathcal{S} = \mathcal{T}$ has $2 \in \mathcal{T}$. Now for any $n \in \{3, 4, \ldots, k+1\}$ and $g \in \{0, 1, \ldots, \mathcal{G}(n) - 1\}$, expect the case when $n = k+1, g = 0$, is

$$R(g, n) = \begin{cases} \{2(\mathcal{G}(n) - g) - 1,\ 2(\mathcal{G}(n) - g)\} & \text{if } n \text{ is even;} \\ \{2(\mathcal{G}(n) - g),\ 2(\mathcal{G}(n) - g) + 1\} & \text{if } n \text{ is odd.} \end{cases}$$

The case $R(0, k+1) = \{k\}$, because $k + 1 \notin \mathit{exp}(\mathcal{S})$, so for all $\mathcal{T}$ also $k \in \mathcal{T}$. To summarize,

$$\mathcal{R}(\mathcal{S}) = \big\{\{2\}, \{3,4\}, \{4,5\}, \ldots, \{k-2, k-1\}, \{k-1, k\}, \{k\}\big\}.$$

When calculating the number of possible minimals or elements of the equivalence class, we can ignore the elements $\{2\}$ and $\{k\}$, since they surely will be included. Without them, the restrictions are made of only two element set. Therefore, we can look at this set as on edges of undirected graph. Clearly, this graph is a path of length $k-4$ ($\mathcal{P}_{k-4}$). The hitting set on two-element set family is equivalent to a *vertex cover*.

Now, to calculate the number of minimal vertex covers of $\mathcal{P}_n$ (in inclusion), let us denote this number as $a(n)$. Then either we include the n-th vertex in the cover or not. If we do include it, to ensure minimality, the $(n-1)$-th vertex cannot be in the set. But then the $(n-2)$-th vertex has to be in the cover, so the number of different covers is $a(n-3)$. If we do not include it, we have to include the $(n-1)$-th vertex, so the number is $a(n-2)$. As a result $a(n) = a(n-2) + a(n-3)$. The cases if $n < 3$ are trivially $a(0) = 1$ and $a(1) = a(2) = 2$.

To calculate the number of different vertex covers of $\mathcal{P}_n$, we can proceed in a similar manner. Let us denote this number $b(n)$. If we include the n-th vertex, the number is $b(n-1)$. If we do not include n-th vertex, we have to include $(n-1)$-th, so the number is $b(n-2)$. As a result $b(n) = b(n-1) + b(n-2)$. The cases when $n < 2$ are trivially $b(0) = 1$ and $b(1) = 2$.

To finalize, we can observe that these sequences are just shifted by 4 and by filling in the trivial cases with the answer 1 we get the expected result. □

So the number of minimals and elements of the duplicate of $\mathcal{S}(1, 2, \ldots, k)$ rise with increasing k as fast as the Padovan and Fibonacci sequences, respectively. Let us now consider the number of minimals of a triplicate of the same game, denoted as $\mathit{minCount}_3(k)$.

Theorem 4.23. The number of minimals of the game $\mathcal{S}(3, 4, \ldots, k)$ is given by the following recurrence:

$$\begin{aligned} \mathit{minCount}_3(k) =& \mathit{minCount}_3(k-2) + \mathit{minCount}_3(k-3) \\ &+ \mathit{minCount}_3(k-4) - \mathit{minCount}_3(k-6), \end{aligned}$$

with $\mathit{minCount}_3(n) = 1$ for $n \in \{3, 4, 5, 6\}$ and $\mathit{minCount}_3(7) = \mathit{minCount}_3(8) = 3$.

Proof. By applying exactly the same method as used in Theorem 4.22, we conclude that

$$\mathcal{R}(\mathcal{S}) = \big\{\{3\}, \{3,4\}, \{3,4,5\}, \{4,5,6\}, \dots,\\ \{k-3, k-2, k-1\}, \{k-2, k-1, k\}, \{k-1, k\}, \{k\}\big\}.$$

Again, because for any $\mathcal{T}$ such that $\mathcal{S} = \mathcal{T}$ it must $\{3, k\} \subset \mathcal{T}$, we can ignore all but the 3-element restrictions. We can again look at this set as on the edges of a hypergraph having $k-4$ sorted vertices on a "path" with a hyperedges connecting all consecutive *triples* of vertices. Let us denote this "path" of length n as $\mathcal{P}'_n = (v_1, \dots, v_n)$, where v_i is the i-th vertex on this "path". We also denote the number of minimal hitting sets (in inclusion) as $a(n)$.

To be able to break down the recurrence more easily, we will also write $r(n,k)$ as the number of minimal hitting sets on $\mathcal{P}'_n$, provided that the first vertex included in the hitting set has to be one of the first k on the path.

Let $n \geqslant 3$ and let us divide the calculation of $a(n)$ into three cases based on the index of the first vertex included in the hitting set. Clearly, it has to be one of v_1, v_2 and v_3. Then we can write

$$\begin{aligned} r(n,1) &= r(n-3,1) + r(n-3,2) + r(n-3,3);\\ r(n,2) &= r(n-2,2) + r(n-2,1) \text{ and}\\ r(n,3) &= r(n-1,1). \end{aligned}$$

Since there is a recursion in $r(k,2)$, we can rewrite it as

$$r(n,2) = r(n-2,1) + r(n-4,1) + r(n-6,1) + \ldots + r(b,1) \quad \text{with } b \leqslant 3.$$

The base cases are clearly

$$r(3,1) = 3, r(3,2) = r(2,1) = r(2,2) = 2, \text{ and } r(3,1) = r(1,1) = 1.$$

Observe also that the difference $r(n,1) - r(n-2,1)$ eliminates this recursive term:

$$r(n,1) - r(n-2,1) = r(n-3,1) + r(n-4,1) - r(n-6,1).$$

Since $a(n) = r(n,1)$, the answer is

$$a(n) = a(n-2) + a(n-3) + r(n-4) - r(n-6).$$

By shifting this sequence by 4 and resolving the additional base cases, we get the expected result. $\square$

Note: The computational results (discussed in Section 4.4.5) suggest that the games in Theorem 4.23 are the games with maximal numbers of minimals. We showed that this holds at least for all games having all numbers in their masters up to 30.

Theorem 4.24. The size of the equivalence class of the game

$$\mathcal{S}(j,\ j+1,\ j+2,\ \ldots,\ k),$$

denoted $\mathit{classSize}_j(k)$, is given by the following recurrence:

$$\mathit{classSize}_j(k) = \sum_{i=1}^{j} \mathit{classSize}_j(k-i),$$

with $\mathit{classSize}_j(i) = 1$ for $i \in \{j, j+1, \ldots, 2j\}$. (This recurrence is sometimes called the j-*Bonacci* sequence [63].)

Proof. Similarly as in the analysis of the game $\mathcal{S}(3, 4, \ldots, k)$, we can observe that now is the question reduced to finding a hitting set on a hypergraph forming a "path" P_{k-4}, where the hyperedges connect all consecutive j-tuples of vertices. By analyzing the cases when the first vertex included in the hitting set is either nth, $(n-1)$th, $(n-2)$th, …, $(n-j+1)$th, or $(n-j)$-th and resolving the base cases in the same manner as in the proof of Theorem 4.22, we get the expected recurrence. □

This leads us to the following corollary that classifies all M-plicate Bounded Nim games.

Corollary 4.25. A subtraction game $\mathcal{S}$ is an M-plicate Bounded Nim(k) for some $M, k \geqslant 1$ if and only if $\{M, M \cdot k\} \subset \mathcal{S}$ and for each set

$$X = \{i, i+1, \ldots, i+M-1\} \text{ with } M \leqslant i \leqslant M \cdot (k-1)$$

we have $\mathcal{S} \cap X \neq \varnothing$.

Proof. By observing that the argument about the form of the restrictions made for $M = 2$ and $M = 3$ simply generalize to any M and that the equation in the theorem is the definition of a hitting set on this "path" hypergraph. □

4.1.2.1 Ultimately Bipartite Subtraction Games

In this section, we present a new ultimately bipartite subtraction game.

Theorem 4.26. The games $\mathcal{S}(3, 7, 10k-1)$ with $k \geqslant 1$ are ultimately bipartite.

Proof. Let us define the following sets:

$$A = \big\{10i + 1 : 0 \leqslant i < k\big\};$$
$$B = \big\{10i + b \text{ where } 0 \leqslant i < k, b \in \{4, 8\}\big\} \text{ and}$$
$$C = \big\{10j : k \leqslant j < 2k\big\}.$$

We will show that the set of lost positions is equal to

$$\mathcal{P} = \{i : i \text{ is even}\} \cup A \setminus B \setminus C.$$

This set of $\mathcal{P}$-values clearly imply a bipartite sequence. Consider now the following cases.

- Let $n \in \mathcal{P}$. We will show that all followers of n are in $\mathcal{N} = \mathbb{N} \setminus \mathcal{P}$.
  - Let $n \in A$. Then $n = 10i + 1$ for some $0 \leqslant i < k$, so $n - 3 = 10(i-1) + 8$ and $n - 7 = 10(i-1) + 4$ which is in $\mathcal{P}$ because of the set B and the third move cannot be used.
  - Otherwise, n is even so any move is to an odd number. If it is a move to $\mathcal{P}$, it would have to be because of A. But for any $m = 10i+1 \in A$ we will show that it can not be a move away from n. Indeed, if the move was 3, $n = m + 3 = 10i + 4$ and if the move was 7, $n = m + 7 = 10i + 8$, so in both cases we get that $n \notin \mathcal{P}$ (because of B). If the move was $10k - 1$, then $n = m + 10k - 1 = 10(i + k)$ which is not in $\mathcal{P}$ because of the set C. Both contradict the premise.
- Let $n \in \mathcal{P}$. We will show that there is always a move to $\mathcal{P}$.
  - Let n be odd. Then all moves are to an even number. So it remains to show that at least one of them is not in $B \cup C$. If $n - 3 \in \mathcal{P}$, we are done. Otherwise, there are two options why $n - a$ could be not lost: either $n - a \in B$ or $n - a \in C$.

    (B) If $n - 3 = 10i + 4$ for some $0 \leqslant i < k$, then $n = 10i + 7$ and we can move to $n - 7 = 10i$ which is in $\mathcal{P}$. If $n - 3 = 10i + 8$ for some $0 \leqslant i < k$, then $n = 10i + 11$. For $i < k - 1$ we get $n \in A$ which contradicts the premise. So let $i = k - 1$. Then $n = 10k + 1$ and we have a move to $n - 10k + 1 = 2$ which is a $\mathcal{P}$-position.

    (C) Let $n - 3 = 10j$ for some $k \leqslant j < 2k$. Then $n = 10j + 3$ and $n - 7 = 10j + 3 - 7 = 10(j - 1) + 6$ which is also a $\mathcal{P}$-position.
  - Let n be even. Then $n \in B \cup C$ If $n \in B$, then we have a move to some $10i + 1 \in A \in \mathcal{P}$. If $n \in C$, then $n = 10j$ and we can move to $n - 10k + 1 \in A \in \mathcal{P}$.

□

Now we remind a previously mentioned Conjecture 5 about the ultimate bipartite subtraction games:

Conjecture. *[43, p. 18] Ho conjectured that an ultimately bipartite subtraction game is non-expandable.*

Counterexample: This is not true and there seems to be many counter-examples, e.g. into ultimate bipartite game $\mathcal{S}(3, 11, 15)$ with a nim sequence

$$\mathcal{G}\text{-sequence}(\mathcal{S}) = (0^3 1^3)^2 2^2 0 3^2 1 2^2 0 3^2 1 0 2 \overline{0 1}.$$

For this sequence, it can be easily verified by hand or computer that the number 9 can be adjoined without changing the sequence.

We will actually show a whole class of subtraction games that violate this conjecture:

Lemma 4.27. For any $k \geqslant 1$, the game $\mathcal{S}(3, 5+6k, 9+6k)$ is ultimately bipartite with pre-period length $n_0 = 12k + 14$ and any of $\{6j + 3 : k \geqslant j \geqslant 1\}$ can be adjoined without changing its nim-sequence. Therefore, not only that there are expandable ultimately bipartite subtraction games, but there is also infinitely many of them and their expansion set can have any possible size.

Proof. We will show that the sequence takes the following form

$$\mathcal{G}\text{-sequence}(\mathcal{S}) = \underbrace{\boxed{0^3 1^3}^{1+k}}_{\text{(A)}} \; \underbrace{\boxed{2^2 0 3^2 1}^{1+k}}_{\text{(B)}} \; \underbrace{\boxed{0 2}}_{\text{(C)}} \; \overline{0 1}.$$

We have divided the pre-period into three disjunct blocks (A), (B) and (C), as shown in the description of the sequence using boxes. Let us start with proving the structure of the pre-period, block by block, while showing that adjoining $6j + 3$ will not affect it for any $j \geqslant 1$.

(A) Notice that first $6k + 5$ values are only affected by the move 3. From Theorem 3.54 and from Corollary 3.42 we know that $\mathcal{S}(3)$ has sequence $\overline{0^3 1^3}$ and expansion set $\mathit{exp}(\mathcal{S}) = \{3\}^{*6}$. Note that $6j + 3 \in \mathit{exp}(\mathcal{S})$ for any $j \geqslant 1$. The last value $\mathcal{G}(6k + 5)$ can use the move $6 + 6k$, but that will not have an effect on the sequence.

(B) Let $n \in [6(k + 1), 12(k + 1))$. We will prove the structure of the part (B) by induction on number of blocks of length 6. Note that since maximal position in (B) is $6(2k + 2) - 1$ and minimal move apart of 3 is $6k + 5$, the moves $5 + 6k$ and $9 + 6k$ are always to the block (A) (except the last position).

- $n \equiv 0, 1 \pmod 6$: The first move $n - 3 = 6(a - 1) + b$ for some $b \in \{3, 4\}$. In block (A) this will be 1, in (B) 3. The moves $5 + 6k$ and $9 + 6k$ are leaving piles of sizes $1, 2$ resp. $3, 4$ under modulo 6, so the corresponding nimbers are 0 and 1. We can see that we have moves to 0 and 1 and no move to 2, hence the value $\mathcal{G}(n) = 2$.
- $n \equiv 2 \pmod 6$: The move 3 goes to the end of a block of type (A) or (B), both resulting in nimber 1. Moves $5 + 6k$ and $9 + 6k$ again are in (A), leaving modules 3 and 5, so their values are 1. We have no move to 0, so $\mathcal{G}(n) = 0$.

- $n \equiv 3, 4 \pmod 6$: The move 3 always ends in (B), so its value is 2, the move $5 + 6k$ leaves modules $4, 5$, so its value is 1 and the move $9 + 6k$ leaves modules $0, 1$, so its value is 0. We have a move to $0, 1$ and 2 so the value $\mathcal{G}(n) = 3$.
- $n \equiv 5 \pmod 6$: All moves leaves modules $0, 2$ in (A) or 3 in (B) which are all 0 positions. The exception is the last position of (B) whose move $5 + 6k$ goes to start of (B), which is 2. This leaves us with the value $\mathcal{G}(n) = 1$.

(C) We can compute these two positions by definition:

$$\begin{aligned}
\mathcal{G}(6(2k+2)) &= \mathit{mex}(\mathcal{G}(6(2k+1)+3), \mathcal{G}(6(k+1)+1), \mathcal{G}(6k+3)) \\
&= \mathit{mex}(1, 2, 1) = 0 \text{ and} \\
\mathcal{G}(6(2k+2)+1) &= \mathit{mex}(\mathcal{G}(6(2k+1)+4), \mathcal{G}(6(k+1)+2), \mathcal{G}(6k+4)) \\
&= \mathit{mex}(3, 0, 1) = 2.
\end{aligned}$$

Now we can prove the bipartite period by stating that for all $n \geqslant 2(6k+6)+2$ we have $\mathcal{G}(n) = n \bmod 2$.

- $n \equiv 0 \pmod 2$: We will show that there is no move to the value 0. Since all moves subtract odd value the only possible 0 value positioned on odd index is in (A). But even the highest possible subtracted value $2(6k+6)+2-9-6k = 6k+5$ cannot reach it. Therefore $G(n) = 0$.
- $n \equiv 1 \pmod 2$: Clearly, $\mathcal{G}(n-3) = 0$ for any such n. Since all ones are on odd positions both in the period and in the block (B) and since the block (A) cannot be reached by a single move, we conclude that $\mathcal{G}(n) = 1$.

It remains to show that adjoining $6j+3$ to the subtraction set will not change this sequence with any $j \geqslant 1$. But from the structure of the sequence we can easily verify that two same values are never in the distance of 9 modulo 12 apart. □

Corollary 4.28. There is an infinite number of expandable ultimately bipartite subtraction games.

Infinite Subtraction Games

Motivated by Question 6, we provide here several simple proofs on the properties of some infinite subtraction games.

Theorem 4.29. Consider games $\mathcal{S}(1, b^a, b^{k_1 \cdot a}, b^{k_2 \cdot a}, \ldots)$ where $k_i \in \mathbb{N}$ for all $i \geqslant 1$ and b is even (for odd b we have a simple bipartite game). These games are purely periodic with period $b^a + 1$.

Proof. We show that for the expansion set of the game $\mathcal{S}(1, b^a)$ holds: $\{b^{ka} : k \in \mathbb{N}\} \subset exp(\mathcal{S})$ by induction on k. Let the claim hold for all $k \leqslant n$. Then the game $\mathcal{S}(1, b^a)$ and $\mathcal{S}(1, b^a, b^{2a}, \ldots, b^{na})$ has the same nim-sequence. Since $\mathcal{S}(1, b^a)$ is a purely periodic game with period $p = b^a + 1$, we have $S^{*p} \subset exp(\mathcal{S})$. Then any $s + k(1 + b^a) \in exp(\mathcal{S})$ such that $s \in \mathcal{S}, k \in \mathbb{N}$. Let $k = (b^{na} - b^{(n-1)a})$ and $s = b^{(n-1)a}$. We get: $(b^{na} - b^{(n-1)a})(1 + b^a) + b^{(n-1)a} = b^{na} - b^{(n-1)a} + b^{(n+1)a} - b^{na} + b^{(n-1)a} = b^{(n+1)a} \in exp(\mathcal{S})$. $\square$

Theorem 4.30. Consider the infinite game Take-A-Square: $\mathcal{S}(i^2 : i \in \mathbb{N})$. This game is aperiodic.

Proof. Suppose the game is periodic with period p. Then $\mathcal{G}(n) = \mathcal{G}(n + p) = \mathcal{G}(n + p^2)$, a contradiction. $\square$

Theorem 4.31. Consider a game Take-A-Prime: $\mathcal{S}(p : p \text{ is prime})$. This game has an infinite kernel.

Proof. By contradiction, let $K = \{k_0, k_1, \ldots, k_m\}$ be all $\mathcal{P}$ positions of $\mathcal{S}$ and let t be any integer such that $t > max\, K$. Consider now a position $n = t! + t$. Since $n > max\, K$, it is an N position and therefore it has to move to some $k_i \in K$. This implies that $n = k_i + p$ for some prime and $k_i \in K$. In other words, $n - k_i$ has to be prime. But we can see that all the positions $\{t! + 1, t! + 2, \ldots t! + t\}$ are all composite, a contradiction. $\square$

Theorem 4.32. Consider the game $\mathcal{S}(\{p : p \text{ is prime }\} \cup \{1\})$. This game is purely periodic with period 4. Furthermore, the equivalence class of this game forms a lattice with the minimal $\mathcal{S}'(1, 2, 3)$.

Proof. First, consider subtraction game $\mathcal{S}'(1, 2, 3)$. This game is purely periodic with nim-sequence $\mathcal{G}(n) = \overline{0123}$. Also note that since no nimber occurs in a single period more then once, it has expansion set $exp(\mathcal{S}) = \mathbb{N} \backslash \{kp : k > 0\}$. The primes are of course a subset of $exp(\mathcal{S}')$. To prove that the game $\mathcal{S}'$ is the minimal of the equivalence class of this game, it suffices to show that by removing any of $\{1, 2, 3\}$ from $\mathcal{S}$, the nim-sequence will change. But that is clear since $\mathcal{G}(1), \mathcal{G}(2)$ and $\mathcal{G}(3)$ can use only these moves. So by removing either $1, 2$ or 3, these values will become 0, respectively. $\square$

Theorem 4.33. Consider the game Subtract-Fibonacci, which allows to subtract any Fibonacci number. This game is aperiodic.

Lemma 4.34. For any integer $m > 0$ there is an integer k such that $F_k = \ell m$ for some ℓ.

Proof. First observe that the sequence $\{F_n \bmod m\}_{n>0}$ is periodic. Indeed, there are only m^2 possible values for blocks of this sequence of length 2, so there must exist indices k, k' with $F_{k-1} \equiv_m F_{k'-1}$ and $F_k \equiv_m F_{k'}$ where $k \neq k'$. Because we also have $F_{n-2} \equiv_m F_n - F_{n-1}$, we can run the recurrence backwards while maintaining the periodicity. Thus we get $F_0 \equiv_m F_{k'-k}$ and therefore $F_{k'-k} = \ell m$ for some ℓ. □

Proof. (Of Theorem 4.33) Now suppose the nim-sequence of SUBTRACT-FIBONACCI is periodic with period p. Let ℓ be an integer such that ℓp is the Fibonacci number (based on the previous lemma such ℓ always exist). There should be $\mathcal{G}(n) = \mathcal{G}(n-p) = \mathcal{G}(n-\ell p)$ for some big enough nm but since we are allowed to subtract ℓp, this is a contradiction. □

We end this section with a generalization of the structural property of the G-sequences of subtraction games showed in Proposition 3.25.

Proposition 4.35. Let $\mathcal{S}(s_1, s_2, \ldots)$ be a subtraction game and let k be the smallest positive multiple of s_1 not in $\mathcal{S}$. Then for any integer $n \geqslant s_1$ and $i \in \{1, 2, \ldots, k-1\}$

$$\mathcal{G}(n) = 0 \text{ implies that there exists } s \in \mathcal{S} \text{ such that } \mathcal{G}(n-s) = i.$$

Proof. For contradiction, let n be the smallest position violating this rule. So for all $s \in \mathcal{S}$ is $\mathcal{G}(n-s) \geqslant k$. Let $\mathcal{G}(n-s_1) = m$ for some $m \geqslant k$. Then there exists $t \in \mathcal{S}$ such that $\mathcal{G}(n-s_1-t) = \ell - 1$. But by the Generalized Ferguson property (Proposition 3.24) we have that $\mathcal{G}(n-t) = \ell$ which contradicts the assumption that there is no move to ℓ. □

4.2 Results on Code-Digit Games

In this section, we provide a few results on various code-digit games.

4.2.1 Code-Digit Games Equivalences

We start by proving the properties of equivalences among code-digit games. Guy noticed these equivalences and described them in [39, ch. 4] without proving them. For completeness of the theory, we provide proofs of these features here.

Proposition 4.36 (Code-digit equivalence, [39, ch. 4])**.** Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ be a code digit game with d_1 even. Now suppose that d_k contains 2^ℓ for some $k \geqslant 1$ and $\ell > 0$. Then it makes no difference to the game if code-digit d_{k+1} contains 2^{j-1} or not.

Proof. When d_1 is even, $\mathcal{G}_{\mathsf{D}}(1) = \mathcal{G}_{\mathsf{D}}(0) = 0$. From a heap of $n > k$ tokens, it is valid to take k tokens and split the remainder into ℓ non-empty heaps: $H_{a_1} + H_{a_2} + \ldots + H_{a_\ell}$ for any $a_i > 0$ such that $a_1 + \ldots + a_\ell = n - k$. Now set $a_\ell = 1$. Then $\mathcal{G}(n) = \mathcal{G}(H_{a_1}) \oplus \ldots \oplus \mathcal{G}(H_{a_{\ell-1}}) \oplus \mathcal{G}(H_{a_\ell}) = \mathcal{G}(H_{a_1}) + \ldots + \mathcal{G}(H_{a_{\ell-1}})$. In other words, this move is equivalent to a move that takes $k+1$ tokens and splits the remainder into $\ell - 1$ heaps. $\square$

Proposition 4.37 (Redundant ones)**.** Let $2^\ell \in \mathsf{d}_k$ for even $\ell > 0$. Then for any $u > 0$ it makes no difference if $\mathsf{d}_{k+\ell u}$ contains 1 or not.

Proof. The added move can be used only for such H_n where $n = k + \ell u$. But the same move can be done by taking k tokens and splitting the remainder into ℓ heaps of size u because $\mathcal{G}(H_u) \oplus \mathcal{G}(H_u) = 0$. $\square$

Theorem 4.38 (About standard form)**.** Let $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3\ldots$ with d_1 even. Consider a game $\mathsf{E} = {\cdot}\mathsf{e}_1\mathsf{e}_2\mathsf{e}_3\ldots$ constructed using the following rule:

$$\mathsf{e}_r \text{ contains } 2^{h+1}{-}1 \quad \text{whenever} \quad \mathsf{d}_{r-h+1} \text{ contains } 2^h.$$

Then $\mathsf{E} \equiv_1 \mathsf{D}$ (D is first cousin of E).

Proof. By induction on n. First note that $\mathcal{G}_{\mathsf{D}}(0) = \mathcal{G}_{\mathsf{D}}(1) = 0$. Now suppose that $\mathcal{G}_{\mathsf{E}}(i) = \mathcal{G}_{\mathsf{E}}(i+1)$ for all $i < n$. Consider any move in D using some $2^h \in \mathsf{d}_{r-h+1}$ from H_{n+1} into $H_{a_1} + H_{a_2} + \ldots + H_{a_h}$ with $a_1 + a_2 + \ldots + a_h = n + 1 - (r - h + 1) = n - r + h$. Thus $\mathcal{G}(n+1)$ is calculated with an excludent $e_{\mathsf{D}} = \mathcal{G}_{\mathsf{D}}(a_1) \oplus \mathcal{G}_{\mathsf{D}}(a_2) \oplus \ldots \oplus \mathcal{G}_{\mathsf{D}}(a_h)$.

Now by applying the rule above, there is a move $2^{h-k} \in \mathsf{e}_r$ into $H_{a_1-1} + H_{a_2-1} + \ldots + H_{a_h-1}$ such that $a_1 - 1 + a_2 - 1 + \ldots + a_h - 1 = n - r$ where $k = |\{i : a_i = 1\}|$, (moves to a heap of size one). Note that $k \in \{0, 1, \ldots, h\}$ because all heaps created by a move 2^{h-k} must be non-empty and therefore all moves $2^0, 2^1, \ldots, 2^h \in \mathsf{e}_r$. Thus the value $\mathcal{G}_{\mathsf{E}}(n)$ is calculated with an excludent $e_{\mathsf{E}} = \mathcal{G}_{\mathsf{E}}(a_1 - 1) \oplus \mathcal{G}_{\mathsf{E}}(a_2 - 1) \oplus \ldots \oplus \mathcal{G}_{\mathsf{E}}(a_h - 1)$ and by induction hypothesis, $e_{\mathsf{E}} = e_{\mathsf{D}}$. We can see that all moves in E correspond to the moves added to D' which can be proved symmetrically using the same argument. $\square$

4.2.2 A New Taking and Breaking Game

Here we describe a new taking and breaking game which is for most cases surprisingly well behaved.

Definition 4.39. Given a set of integers $\mathcal{S}$, the Take-K-Break-K game on heaps of tokens is played as follows. Players in their move choose some $k \in \mathcal{S}$ and some pile, say H_n, and removes k tokens from it while splitting the remaining $n - k$ tokens into k non-empty piles.

Note that this class of games includes for instance the game ·04, which has already been mentioned in Winning Ways [42, p. 104] and is still not solved even though Flammenkamp computed up to 2^{28} values [20].

Theorem 4.40. Let $\mathcal{S}$ be a set for the Take-K-Break-K game. Then

(a) If $1 \in \mathcal{S}$, it is a simple parity game with the nim-sequence

$$\mathcal{G}\text{-sequence}(\mathcal{S}) = 0\,\overline{01}.$$

(b) If $\min \mathcal{S} > 2$, then the nim-sequence is arithmetic periodic with $p = 2m - 1$ and saltus $s = 1$:

$$\mathcal{G}\text{-sequence}(\mathcal{S}) = 0\,\overline{0\ldots 0}\ (+1).$$

So the game is a first cousin of $(2m-1)$-plicate Nim.

Proof. We will prove both parts using induction on a size of the heap n.

(a) Note that the move 1 represents the subtraction of a single token from heap of size greater than 1 so the sequence for the game $\mathcal{S}(1)$ is obvious. It remains to show that no move $k > 1$ will not affect the sequence.

- Let $n > 0$ be odd. We will show that $\mathcal{G}(n) = 1$. First note that the move 1 leads to $\mathcal{G}(n-1) = 0$. It remains to show that there is no move to G-value 1. Consider a move with some $s \in \mathcal{S}$. For contradiction, suppose that it leads to a position with $\mathcal{G}$-value 1.

  If s is even, the move should consist of $2i + 1$ of G-values 0 and $2j + 1$ of G-values 1 for some $i, j \geqslant 0$. But this contradicts with $n - s$ being odd. If s is odd, the move should consist of $2i$ of G-values 0 and $2j + 1$ of G-values 1 for some $i \geqslant 1, j \geqslant 0$. But $n - s$ should equal $2i + 2j + 1$. But $n - s$ is even, a contradiction.

- Let $n > 0$ be even. We need to show that $\mathcal{G}(n) = 0$ so there is no move to zero. But this can be proved using a symmetric parity argument as for odd n.

(b) Let $m = \min \mathcal{S}$ such that $(m > 2)$ and let $n > 0, n = kp + d + 1$ for some $k \geqslant 0, 0 \leqslant d < p$ (n is in the k-th block).

- There is a move to each $i \in \{0, 1, \ldots, k-2\}$: After subtracting m we need to divide $n - m$ tokens into m non-empty heaps. Let us start with $m - 1$ heaps of size 1 and a single heap of size $ip + 1$. This move has clearly $\mathcal{G}$-value $\mathcal{G}(1) + \ldots + \mathcal{G}(1) + \mathcal{G}((i+1)p+1) = 0 \oplus \ldots \oplus 0 \oplus i = i$. We need to fix the move so it contains $n - m$ tokens in total. But since $m > 2$, there are at least 2 heaps of size 1 in this move. We can simultaneously increase each of them by 1 until the total size equals $n-m$ or $n-m+1$. Note that since $\mathcal{G}(a) \oplus \mathcal{G}(a) = 0$ for any a, this operation does not affect the $\mathcal{G}$-value of this move. If we end up with total size $n-m+1$, since $p = 2m-1 > 1$, we can increase the heap of size $(i+1)p+1$ by one without changing the $\mathcal{G}$-value, thus the $\mathcal{G}$-value of the move remains i.

- There is no move to $k-1$: Note that $n \leqslant (k+1)p$ and after subtracting minimal amount m we need to split the remaining $n - m \leqslant kp + m - 1$ tokens into m non-empty heaps. Observe the following property. Consider a heap of size ℓ with $\mathcal{G}(\ell) = 2^i + 2^j$ such that $i \neq j$. Notice that $\ell \leqslant (2^i + 2^j + 2)p$. We will show that if we were to replace this heap with two heaps of sizes ℓ_1 and ℓ_2 having $\mathcal{G}(\ell_1) \oplus \mathcal{G}(\ell_2) = 2^i + 2^j$, it would take more than ℓ tokens. If we look for the smallest possible heaps whose $\mathcal{G}$-values have i-th and j-th bit on, we get $\ell_1 \geqslant (2^i+1)p+1$ and $\ell_2 \geqslant (2^j+1)p+1$ so $\ell_1 + \ell_2 \geqslant (2^i + 2^j + 2)p + 2 > \ell$.

  The property above clearly implies that the smallest move (in terms of total size of heaps) with the desired $\mathcal{G}$-value is $\mathcal{G}(1) + \ldots + \mathcal{G}(1) + \mathcal{G}(kp+1)$. But the total size of this move equals $kp+m$ and we have available only $kp+m-1$ so there is no such move at all.

$\square$

We end this section by answering a simple question about the sparse spaces of octal games discussed in Section 3.4.2.

***Question* 11.** *Given disjunct sets $R, C \subset \mathbb{N}$, decide whether there exists a sparse space $\mathcal{R}$ which satisfies this decomposition.*

Answer: The answer can be determined in polynomial time by setting $A := R \cap 2^{\mathbb{N}}$ and verifying that all numbers in R have even population count and all numbers in C have odd population count when applying the ignore mask defined by A.

4.3 Results on Partizan Games

In this section, we will analyze a new family of partizan taking and breaking games and provide several results for some particular classes of games in this family.

4.3.1 Partizan Subtraction Games

Definition 4.41. Let $\mathcal{S}_L$ and $\mathcal{S}_R$ be impartial subtraction games. We will define the **partizan subtraction game** $\mathcal{S}(\mathcal{S}_L \mid \mathcal{S}_R)$ as a duel heap game $\mathcal{S}_L$ **versus** $\mathcal{S}_R$.

Theorem 4.42. The partizan subtraction game $\mathcal{S}(1 \mid a)$ for $a \geqslant 1$ has a periodic canonical sequence as follows.

$$\mathcal{C}\text{-sequence}(\mathcal{S}) = \overline{0\,1\,2\,\ldots\,(a-1)\,\{a-1|0\}}.$$

Proof. By induction on n. The first period is clear because these are the only moves available. Let that the statement holds for all positions of heap size less than n. Set $p = a+1$ and denote $n = kp + a$ for some $k \geqslant 0$ and $0 \leqslant a < p$. Now consider the following cases:

(a) $n \equiv 0 \pmod p$. Then $\mathcal{C}(n) = \{\mathcal{C}(kp+a) \mid \mathcal{C}(kp+1)\} \overset{\text{IH}}{=} \{a-1 \mid 0 \parallel 1\}$. Clearly, this is a lost game, so $\mathcal{C}(n) = 0$.

(b) $n \equiv \ell \pmod p$ where $\ell \in \{1, 2, \ldots, a-2\}$. Then

$$\mathcal{C}(n) = \{\mathcal{C}(kp+\ell-1) \mid \mathcal{C}(kp+(\ell-1)+k-a)\} \overset{\text{IH}}{=} \left\{k-1 \,\middle\|\, k+1\right\}.$$

By Simplicity rule (Theorem 2.52) we get $\mathcal{C}(n) = k$.

(c) $n \equiv a-1 \pmod p$. We apply the induction hypothesis the same way which results in $\mathcal{C}(n) = \{a-2 \parallel a-1 \mid 0\}$. Because $\{a-1 \mid 0\}$ reverses through $a-1$, $\mathcal{C}(n) = \{a-2 \mid\} = a-1$.

(d) $n \equiv a \pmod p$. Applying induction hypothesis results in $\mathcal{C}(n) = \{a-1 \mid 0\}$.

$\square$

Theorem 4.43. The partizan subsection game $\mathcal{S}(1 \mid 1, 2, a_3, a_4, \ldots)$ where $a_i \geqslant 2$ for all i has arithmetic-periodic canonical sequence with pre-period length 1, period length 2 and saltus $\Downarrow$ given by

$$\mathcal{C}(n) = 0\,\overline{*\downarrow} \quad (+\Downarrow).$$

Proof. (By induction on n). Let us first handle the cases with $n < 3$. Clearly, $\mathcal{C}(0) = 0$, $\mathcal{C}(1) = \{0 \mid 0\} = *$ and $\mathcal{C}(2) = \{* \mid 0, *\}$ where Right's $*$ reverses through 0 so $\mathcal{C}(2) = \{* \mid 0\} = \downarrow$.

Now let the statement holds for all positions smaller than n. Then consider the following cases:

(a) Let n is even. Then $\mathcal{C}(n) = \left\{\downarrow(n-2) \,\middle|\, G^R\right\}$ where G^R ranges over any possible subset of smaller positions. We will see that they are all dominated by $\mathcal{C}(n-1) = \downarrow(n-2)*$ or by $\mathcal{C}(n-2) = \downarrow(n-3)$. But that follows directly from the fact that $\Downarrow < \downarrow$ and $\Downarrow* < \downarrow*$ which we have seen in Observation 2.65. So we get that

$$\mathcal{C}(n) = \{\downarrow(n-2)* \mid \downarrow(n-2)*, \downarrow(n-3)\}.$$

But since $\downarrow n = \{\downarrow(n-1)* \mid 0\}$ and $\downarrow n* = \{\downarrow(n-1) \mid 0\}$, both Right's moves reverse out to 0 resulting in $\mathcal{C}(n) = \{\downarrow(n-2)* \mid 0\} = \downarrow(n-1)$.

(b) Let n is odd. Then by exactly the same argument we get

$$\mathcal{C}(n) = \{\downarrow(n-2) \mid \downarrow(n-2), \downarrow(n-3)*\},$$

which after bypassing the reversible moves results in $\mathcal{C}(n) = \{\downarrow(n-2) \mid 0\}$ $= \downarrow(n-1)*$.

□

We will now continue with our results by proving Conjecture 7 about canonical forms of the game $\mathcal{S}(1,2 \mid 1,3)$, stated by Plambeck in [61, ch. 3].

The Game $\mathcal{S}(1,2 \mid 1,3)$

The analysis of the canonical forms of the game $\mathcal{S}(1,2 \mid 1,3)$ will be a slightly more complicated. Let us define a new basic game value first.

Definition 4.44. We define the games $\uparrow^{[2]}$ and $\downarrow^{[2]}$ as

$$\uparrow^{[2]} = \{\uparrow \mid *\} \qquad \downarrow^{[2]} = \{* \mid \uparrow\}.$$

This game is a part of a greater family of games called the **uptimals**, see [1, LIP, p. 188].

The multiples of the uptimal $\uparrow^{[2]}$ has an interesting relationship to the multiples of up:

Lemma 4.45. Let k be integer with $k \geqslant 0$. Then

$$(a) \quad k \cdot \uparrow^{[2]} \leqslant \uparrow(k+1), \qquad\qquad (b) \quad k \cdot \uparrow^{[2]} \leqslant \uparrow(k+2) + *.$$

Also for $k \geqslant 1$ we have

$$(c) \quad \uparrow k \leqslant k \cdot \uparrow^{[2]}, \qquad\qquad (d) \quad \uparrow k \leqslant (k+1)\uparrow^{[2]} + *.$$

Proof. We will prove this lemma by induction on k. The only interesting cases for $k = 0$ are in (b), (c) and (d). We will show them by presenting a winning strategy for Left playing as a second on the game made by the difference of right and left hand side of the equations.

For (b), we want to show $0 \leqslant \Uparrow + *$. If Right moves to the up, Left can move to the star he created resulting in $\uparrow > 0$, so Left wins. If Right moves to the star, Left can eliminate a single up resulting also in $\uparrow > 0$.

For the case (c) we need to show that in $\uparrow^{[2]} + \downarrow$ Left wins as a second. If Right starts to $\uparrow^{[2]}$, Left will move to $\downarrow$ resulting in $* + * = 0$. If Right starts to $\downarrow$, left turns $\uparrow^{[2]}$ to $\uparrow > 0$.

In the case of (d), we want to show that $\uparrow \leqslant 2 \cdot \uparrow^{[2]} + *$. Notice that if Right starts with a move into $\uparrow^{[2]}$, Left can reply into $\uparrow^{[2]} + \downarrow$ which is by (c) Left player's win. If Right starts into $\downarrow$, Left can move to $2 \cdot \uparrow^{[2]} > 0$ and if Right starts into $*$, Left can move to $2 \cdot \uparrow^{[2]} + *$. From this position Right can move $\uparrow^{[2]}$ or $2 \cdot \uparrow^{[2]}$ which are both greater than zero, so Left wins.

Now suppose that the statements hold for all $k' < k$. For all the statements (a)-(d) we will show Left's strategy playing second on the difference of the right and left hand side.

(a) $\overbrace{\uparrow(k+1)}^{(1)} + \overbrace{k \cdot \downarrow^{[2]}}^{(2)}$. If Right starts to (1), Left replies to (2) resulting in $\uparrow k + (k-1) \cdot \downarrow^{[2]} + \overbrace{* + *}^{0}$ so by induction hypothesis, on (a), Left wins.

If Right starts to (2), Left continues into the same component resulting in $\uparrow k + (k-2) \cdot \downarrow^{[2]} + *$ so by induction hypothesis, on (B), Left wins.

(b) $\overbrace{\uparrow(k+2)}^{(1)} + \overbrace{k \cdot \downarrow^{[2]}}^{(2)} + \overbrace{\quad * \quad}^{(3)}$. Again, if Right starts to (1), Left replies to (2) resulting in $\uparrow(k+1) + (k-1) \cdot \downarrow^{[2]} + *$ so by IH on (b) we are done. If Right starts in (2), Left plays into the same component resulting in $\uparrow(k+1) + (k-2) \cdot \downarrow^{[2]}$ so by IH on (a) we are done. If Right starts in (3), Left plays to (1) resulting in $\uparrow(k+1) + k \cdot \downarrow^{[2]} + *$ so again by IH on (a) we are done.

(c) $\overbrace{k \cdot \uparrow^{[2]}}^{(1)} + \overbrace{\quad \downarrow k \quad}^{(2)}$. If Right starts in (1), Left replies in (2) resulting in $(k-1) \cdot \uparrow^{[2]} + \downarrow(k-1)$ which by IH on (c) is Left player's win. If Right starts in (2), Left can play into (2) as well because $k > 1$. This results in $k\uparrow^{[2]} + \downarrow(k-2) + *$. For $k > 2$ we can apply IH by (d) and we are done. For $k = 2$ we have $2 \cdot \uparrow^{[2]} + * > 0$ because $\uparrow^{[2]} > 0$.

(d) $\overbrace{(k+1)\cdot\uparrow^{[2]}}^{(1)} + \overbrace{\downarrow k}^{(2)} + \overbrace{*}^{(3)}$. If Right starts to (1), we get $k\cdot\uparrow^{[2]} + \downarrow k$ which Left can win as a second by same argument as above. If Right starts to (2), Left can reply to (3) resulting in $(k+1)\cdot\uparrow^{[2]} + \downarrow(k-1)$ which by IH on (c) Left wins. If Right starts to (3), Left continues to (2) resulting in $(k+1)\cdot\uparrow^{[2]} + \downarrow(k-1) + *$ which by IH on (d) Left wins.

□

Corollary 4.46. Let a, b, c be integers with $a + b \leqslant c$ and b > 0. Then from Lemma 4.45 follows

$$a\cdot\uparrow^{[2]} + \uparrow b \;\leqslant\; c\cdot\uparrow^{[2]}$$

and for a, b, c with $a + b < c$ also the starred version

$$a\cdot\uparrow^{[2]} + \uparrow b + * \;\leqslant\; c\cdot\uparrow^{[2]}$$

Now let us consider the canonical form of scalar multiples of this uptimal. We did not find a proof of this canonical forms in any literature, only Siegel states it as Exercise II.4.5 [71, CG, p. 97], so we provide the proof here.

Lemma 4.47. Let $k \geqslant 2$. The canonical form of $k\cdot\uparrow^{[2]}$ is

$$k\cdot\uparrow^{[2]} = \begin{cases} \{\ \underbrace{0 \,|||\, \cdots \,||\, 0}_{k-1} \mid \uparrow^{[2]}\ \} & \text{if } k \text{ is odd;} \\ \{\ \underbrace{0 \,|||\, \cdots \,||\, 0}_{k-1} \mid \uparrow^{[2]}*\ \} & \text{if } k \text{ is even.} \end{cases}$$

This represents the game where Right needs to move k times before Left will be able to move to $\uparrow^{[2]}$. If Left moves in this game earlier, he eliminates the game completely. So we have $2\cdot\uparrow^{[2]} = \{0 \mid \uparrow^{[2]}*\}$, $3\cdot\uparrow^{[2]} = \{0 \,\|\, 0 \mid \uparrow^{[2]}\}$, $4\cdot\uparrow^{[2]} = \{0 \,|||\, 0 \,\|\, 0 \mid \uparrow^{[2]}*\}$ and so on.

Proof. We will prove the statement by induction on k. Let $k = 2$. Then $2\cdot\uparrow^{[2]} = \{\uparrow^{[2]} + \uparrow \mid \uparrow^{[2]} + *\}$. By Corollary 4.46 we can see that Left's move reverses out completely to 0 because moves only replaces $\uparrow^{[2]}$ with $\uparrow$ and $*$ and $\uparrow, \downarrow$ with $*, 0$. Thus we get $2\cdot\uparrow^{[2]} = \{0 \mid \uparrow^{[2]} + *\}$.

Notice the following property of the expected canonical form: for any $k \geqslant 1$ we have

$$* + \{\ \underbrace{0 \,|||\, \cdots \,||\, 0}_{k} \mid \uparrow^{[2]}\ \} = \{\ \underbrace{0 \,|||\, \cdots \,||\, 0}_{k} \mid * + \uparrow^{[2]}\ \}. \qquad (4.1)$$

This is due to the fact that on each position of the game three, Left's moves to $*$ and $'\cdot\uparrow^{[2]}$ for some k' reverse out to zero through 0 and $k'\cdot\uparrow^{[2]}$, respectively.

Now let $k > 2$ and suppose that the statement holds for all $k' < k$. Then

$$
\begin{aligned}
k \cdot \uparrow^{[2]} &= \{\uparrow \mid *\} + (k-1) \cdot \uparrow^{[2]} \\
&= \left\{\uparrow + (k-1) \cdot \uparrow^{[2]}, \uparrow \mid * + (k-1) \cdot \uparrow^{[2]},\ \uparrow^{[2]} + (k-2) \cdot \uparrow^{[2]}\right\}. \\
&= \left\{\uparrow + (k-1) \cdot \uparrow^{[2]}, \uparrow \mid * + (k-1) \cdot \uparrow^{[2]},\ (k-1) \cdot \uparrow^{[2]}\right\}.
\end{aligned}
$$

Again by Corollary 4.46 Left's moves reverse out to zero resulting in $k \cdot \uparrow^{[2]} = \left\{0 \mid * + (k-1) \cdot \uparrow^{[2]},\ (k-1) \cdot \uparrow^{[2]}\right\}$. Also Right's move $(k-1) \cdot \uparrow^{[2]}$ reverses through $(k-2) \cdot \uparrow^{[2]} + \uparrow$, thus we get $k \cdot \uparrow^{[2]} = \left\{0 \mid * + (k-1) \cdot \uparrow^{[2]}\right\}$. Now the observation in Equation 4.1 about bypassing the star proves the expected canonical form. □

Finally, we have everything we need to prove the canonical form of the game $\mathcal{S}(1, 2 \mid 1, 3)$.

Theorem 4.48. The partizan subtraction game $\mathcal{S}(1, 2 \mid 1, 3)$ has its canonical form sequence defined as follows. Let $n = 3k + a$ where $k \geqslant 0$ and $0 \leqslant a < 3$.

If k is *even*:

$$
\mathcal{C}(n) = \begin{cases} \uparrow k & \text{if } a = 0; \\ k \cdot \uparrow^{[2]} + * & \text{if } a = 1; \\ \uparrow(k+1) & \text{if } a = 2. \end{cases}
$$

If k is *odd*:

$$
\mathcal{C}(n) = \begin{cases} \uparrow k + * & \text{if } a = 0; \\ k \cdot \uparrow^{[2]} & \text{if } a = 1; \\ \uparrow(k+1) + * & \text{if } a = 2. \end{cases}
$$

See the first few values of this sequence in Figure 4.1.

n	1	2	3	4	5	6	7	8	9	...
$\mathcal{C}(n)$	0	$*$	$\uparrow$	$\uparrow*$	$\uparrow^{[2]}$	$2\uparrow*$	$\uparrow 2$	$2 \cdot \uparrow^{[2]}*$	$\uparrow 3$	...

Figure 4.1: Canonical form sequence of $\mathcal{S}(1, 2 \mid 1, 3)$.

Proof. We will prove this by induction on n with a thorough case analysis where we will be applying Corollary 4.46 and Lemma 4.47 and the basic properties of ups and downs observed in 2.65. The base cases when $0 \leqslant< n < 3$ are already in canonical form. Suppose that $n = 3(k+1) + a$ where $k \geqslant 0$ and for all smaller values the statements holds.

(a) $a = 0$.

(i) k is even. Then

$$
\mathcal{C}(n) = \left\{\uparrow(k+1), k \cdot \uparrow^{[2]} + * \,\middle|\, \uparrow(k+1), \uparrow k\right\}.
$$

Observe that both Left's moves reverse out up to zero. The first one through $k\uparrow*$ by basic properties of ups and the second one through $(k-1)\uparrow^{[2]} + *$ by Lemma 4.47.a. Also the first Right's option is dominated by the second, so $\mathcal{C}(n) = (k+1) \cdot \uparrow + *$.

(ii) k is odd. Then

$$\mathcal{C}(n) = \left\{\uparrow(k+1) + *, k \cdot \uparrow^{[2]} \;\middle|\; \uparrow(k+1) + *, \uparrow k + *\right\}.$$

By properties of ups, the first Left's move reverse out through $\uparrow(k+1)$ and the first Right's move is dominated by the second. Also the second Left's move reverses through $(k-1)\cdot\uparrow^{[2]} + *$ by Lemma 4.47.b. Thus we get $\mathcal{C}(n) = \{0 \mid \uparrow k + *\} = (k+1)\cdot\uparrow$.

(b) $a = 1$.

(i) k is even. Then

$$\mathcal{C}(n) = \left\{\uparrow(k+1) + *, \uparrow(k+1) \;\middle|\; \uparrow(k+1) + *, k \cdot \uparrow^{[2]} + *\right\}.$$

The first Left's move reverses through $\uparrow k + * + *$ by Lemma 4.47.a and the second through $\uparrow + *$ by Lemma 4.47.d. The first Right's move is also dominated (by Lemma 4.47.a). By Corollary 4.46 we get $\mathcal{C}(n) = \left\{0 \;\middle|\; k \cdot \uparrow^{[2]} + *\right\} = k \cdot \uparrow^{[2]}$

(ii) k is odd. Then

$$\mathcal{C}(n) = \left\{\uparrow(k+1), \uparrow(k+1) + * \;\middle|\; \uparrow(k+1), k \cdot \uparrow^{[2]}\right\}.$$

The first Left's move reverses through $\uparrow k + *$ by Lemma 4.47.b, the second through $\uparrow k$ by properties of ups. The first Right's move is dominated because of Lemma 4.47.a. So by Corollary 4.46 we have $\mathcal{C}(n) = \left\{0 \;\middle|\; k \cdot \uparrow^{[2]}\right\} = k \cdot \uparrow^{[2]} + *$

(c) $a = 2$.

(i) k is even. Then

$$\mathcal{C}(n) = \left\{(k+1)\cdot\uparrow^{[2]}, \uparrow(k+1) + * \;\middle|\; (k+1)\cdot\uparrow^{[2]}, \uparrow(k+1) + *\right\}.$$

The first Left's move reverses through $k \cdot \uparrow^{[2]} + *$ by the Lemma 4.47.a and the second simply through $\uparrow k$. The first Right's move is dominated because of Lemma 4.47.d. We get $\mathcal{C}(n) = \{0 \mid \uparrow(k+1)\} = \uparrow(k+2) + *$.

(ii) k is odd. Then

$$\mathcal{C}(n) = \left\{\uparrow(k+1), \uparrow(k+1) + * \;\middle|\; \uparrow(k+1), k \cdot \uparrow^{[2]}\right\}.$$

The first Left's move reverses through $k \cdot \uparrow^{[2]}$ by Lemma 4.47.a, the second through $\uparrow k + *$ by property of ups. The first Right's move is dominated because of Lemma 4.47.c. We get $\mathcal{C}(n) = \{0 \mid \uparrow(k+1) + *\} = \uparrow(k+2)$.

□

4.3.2 Partizan Pure Breaking Games

The pure breaking games were presented under an impartial setting in Section 3.6 Here we will attempt to solve some subclasses of partial versions of these games under the duel generalization.

Definition 4.49. Let A, B be sets of positive integers called the *cut numbers*. We define a partizan Pure Breaking Game $\mathcal{PB}(A, B)$ as the heap game with the following options

$$H_n := \{H_{a_1} + \ldots + H_{a_k} \mid H_{b_1} + \ldots + H_{b_\ell}\}$$

such that $k \in A, a_0 + a_1 + \ldots + a_k = n, a_i > 0\ \forall i$ and $\ell \in B, b_0 + b_1 + \ldots + b_\ell = n, b_i > 0\ \forall i$. In other words, in $\mathcal{PB}(A, B)$ a player's move consists of choosing a heap and splitting it into $\ell + 1$ non-empty heaps with ℓ being his cut number. We will call this move an *ℓ-cut*.

In code-digit game notation, this corresponds to a game $(\mathsf{D}^L, \mathsf{D}^R)$ with

$$\mathsf{d}_0^L = \sum_{k \in A} 2^{(k+1)} \qquad \text{and} \qquad \mathsf{d}_0^R = \sum_{\ell \in B} 2^{(\ell+1)}$$

This game can be also played with a row of pins where each player can lay restricted number of separators between pins inside a region that has not yet been separated.

In this section, many proofs will be by induction on the size of the heap. Note that the base case $\mathcal{C}(0) = \mathcal{C}(1) = 0$ holds trivially for any such game.

Theorem 4.50. Let $A, B \subset \mathbb{N}$ be finite sets of integers such that $\min A < \min B$. Then $\mathcal{PB}(A) > \mathcal{PB}(B)$.

Proof. We will show that for any (big enough) integer n the $\mathcal{PB}(A \mid B)$-heap H_n is a Left player's win ($H_n > 0$). We will prove this by showing that $R(H_n) > 0$. Suppose that $A = \{a_1, \ldots, a_k\}, B = \{b_1, \ldots, b_\ell\}$ with $a_1 < b_1$ and let $n \geqslant (2a_1 + 1)(b_\ell + 1)$. After Right's first move, there are $m \leqslant (b_\ell + 1)$ heaps $H_{x_1} + \ldots + H_{x_m}$. Because $n \geqslant (2a_1 + 1)(b_\ell + 1)$, without loss of generality $x_1 \geqslant 2a_1 + 1$. Let Left performs a move $H_{x_1} \to H_{a_1+1} \cdot H_1 + H_{x_1 - 2a_1}$. Then let Left's response to any Right's move be any move in some heap H_x such that $x \geqslant b + 1$. If there is no such heap, the player R can not move and the value of such position equals $\sum_{x \in X} \left\lfloor \frac{x_i - 1}{a_1} \right\rfloor$ where X represents a set of heap sizes. By definition, this is also the value of the right stop of the game H_n. Since Left's first move created a heap of size $a_1 + 1$ and nobody has played in it yet, we get $R(H_n) \geqslant 1$. □

Theorem 4.51. The game $\mathcal{PB}(1, a_1, a_2, \ldots \mid 1, b_1, b_2, \ldots)$ for *odd* $\mathbf{a_i}, \mathbf{b_i} \geqslant 2$ is periodic with the following canonical form sequence:

$$\mathcal{C}\text{-sequence}(\mathcal{PB}) = (\overline{0, *}).$$

Proof. (By induction on n) Clearly, $\mathcal{C}(2) = \{0 \mid 0\} = *$. Suppose the claim holds for heaps smaller than n.

(a) Let $n = 2k + 1$. For contradiction let $\mathcal{C}(n) = *$. Note that by induction hypothesis, any move is just a sum of stars and zeros. Any move splits a heap into even number of non-empty heaps. So there has to be a move into $(2i + 1) \cdot * + (2j + 1) \cdot 0$ for some $i, j \geqslant 0$. But by IH, value 0 have only odd heaps and value $*$ have only even heaps, so the sum of heaps in this move would have to be odd, a contradiction.

(b) Let $n = 2k$. For contradiction let $\mathcal{C}(n) = 0$. Similarly as in odd case, there has to be a move into $(2i) \cdot * + (2j) \cdot 0$. But the sum of heap sizes in this move is odd, a contradiction.

□

Theorem 4.52. The game $\mathcal{PB}(1, 2, 3, a_1, a_2, \ldots \mid 1, 2, 3, b_1, b_2, \ldots)$ for $\mathbf{a_i}, \mathbf{b_i} > 3$ reduces to nim.

Proof. By induction on n, let us assume that for each $m, 0 \leqslant m < n : \mathcal{C}(m) = *(m-1)$. It suffices showing that from $\mathcal{C}(n)$ there is a move to $\mathcal{C}(m)$ for any m such that $0 \leqslant m < n$. Such move looks as follows:

$$\mathcal{C}(m) + \begin{cases} \mathcal{C}(1) & \text{for } n - m = 1, \\ \mathcal{C}(\frac{n-m}{2}) + \mathcal{C}(\frac{n-m}{2}) & \text{for even } n - m, \\ \mathcal{C}(1) + \mathcal{C}(\frac{n-m-1}{2}) + \mathcal{C}(\frac{n-m-1}{2}) & \text{for odd } n - m > 2. \end{cases}$$

Using IH, the value of this move clearly equals $*(m-1)$, therefore $\mathcal{C}(n) = *(n-1)$. □

Theorem 4.53. The game $\mathcal{PB}(a, b_1, b_2, \ldots \mid a, c_1, c_2, \ldots)$ for $\mathbf{a}, \mathbf{b_i}, \mathbf{c_i} > 1$ is periodic with the following canonical form sequence

$$\mathcal{C}\text{-sequence}(\mathcal{PB}) = \underbrace{\overline{00\ldots0}}_{a} \quad (+*).$$

Proof. (By induction on n.) Suppose that the statement holds for any heap smaller than n. Let $n = 1{+}k{\cdot}a{+}\ell$ for some $\ell \in [0, a), k \geqslant 0$. We will prove that $\mathcal{C}(1{+}k \cdot a + \ell) = *k$. Since these are *nimbers*, we will use the Sprague-Grundy notation $\mathcal{G}(1{+}k \cdot a + \ell) = k$. This holds if and only if:

(a) There is an m-cut with nim-value i for each $i \in [0, k)$. Let us fix a single heap of a-cut of size $1{+}i \cdot a + \ell$. Observe that $\mathcal{G}(1{+}i \cdot a + \ell) = i$. Now we only need to divide the remaining $1{+}k \cdot a + \ell - (1{+}i \cdot a + \ell) = (k - i)a$ stones into a heaps without changing this value.

(i) If a is even, we can use a heaps of size $k - i$ resulting in the move to $\mathcal{G}(1+i \cdot a + \ell) \oplus a \cdot \mathcal{G}(k-1) = i$.

(ii) If a is odd, first we use heap of size a – observe that $\mathcal{G}(a) = 0$. Then by using the same trick we get a move to $\mathcal{G}(1+i \cdot a + \ell) \oplus \mathcal{G}(a) \oplus a \cdot \mathcal{G}(k-1) = i$.

(b) There is no m-cut to nim-value k. For contradiction, suppose there is a m-cut ($m \geqslant a$) to the nimber k of the form:

$$\mathcal{G}(1+k_0 + \ell_0) \oplus \mathcal{G}(1+k_1 + \ell_1) \oplus \ldots \oplus \mathcal{G}(1+k_m + \ell_m) \;=\; k$$

where $\ell_i \in [0, a), k_i \geqslant 0$. Now observe that

$$k = \bigoplus_{i=0}^{m} \mathcal{G}(1+k_i + \ell_i) = \bigoplus_{i=0}^{m} k_i \leqslant \sum_{i=0}^{m} k_i \leqslant k \text{ which implies } \sum_{i=0}^{m} k_i = k.$$

The first inequality stems from the fact that the sizes of the heaps need to sum up to $n = 1+)k \cdot a + \ell$. The second inequality is just a basic property of xor operation. The third inequality results from the premise that $\ell_i < a$ for each i. This actually says that no heap in such move can have size of the form $j \cdot a$ or any $j \geqslant 0$.

Thus each of the k whole a-tuples of the $1+k \cdot a + \ell$ stones is in some heap of the move together. Let us consider the reminders – the heap sizes modulo a.

$$m \geqslant 1+\ell = \sum_{i=0}^{m} (1+\ell_i) > 1+m, \text{ a contradiction.}$$

□

Theorem 4.54. The game $\mathcal{PB}(1 \mid a)$ for $\mathbf{a} \geqslant 1$ is arithmetic periodic with the following canonical form sequence:

$$\mathcal{C}\text{-sequence}(\mathcal{PB}) = \overline{012\ldots(a-1)} \quad (+\{a-1 \mid 0\}).$$

Proof. Let us begin the proof with some observations. Let $\Delta = \{a-1 \mid 0\}$.

Observation 1: Number avoidance theorem (Theorem 2.56) implies that $2 \cdot \{a \mid 0\} = a$ so we can write the expected canonical form of a heap of size $n = 1 + ka + \ell$ for $k \geqslant 0$ and $\ell \in [0, a)$ as:

$$\mathcal{C}(1 + ka + \ell) = k \cdot \Delta + \ell = \begin{cases} \frac{k}{2}(a-1) + \ell & \text{for even } k, \\ \lfloor \frac{k}{2} \rfloor (a-1) + \ell + \Delta & \text{for odd } k. \end{cases}$$

Observation 2: For all $k \geqslant 1$ we have:

$$\begin{aligned}\mathcal{C}(ka) - \mathcal{C}(ka+1) =& (k-1)\cdot\Delta + a - 1 \;-\; k\cdot\Delta \\ =& \{0|-(a-1)\} + a - 1 \\ =& \Delta.\end{aligned}$$

For all other n we can write $\mathcal{C}(n) - \mathcal{C}(n+1) = -1$ (the increase of value happens only on the end of each block of size a). Following table shows this property on a part of the canonical sequence of the game $\mathcal{PB}(1 \mid 3)$.

$+1 \quad +1 \quad -\Delta \quad +1 \quad +1 \quad -\Delta \quad +1 \quad +1 \quad -\Delta$

n	1	2	3	4	5	6	7	8	9	...
$\mathcal{C}(n)$	0	1	2	$\{2 \mid 0\}$	$\{3 \mid 1\}$	$\{4 \mid 2\}$	2	3	4	...

$-1 \quad -1 \quad +\Delta \quad -1 \quad -1 \quad +\Delta \quad -1 \quad -1 \quad +\Delta$

Figure 4.2: Canonical form sequence of $\mathcal{PB}(1 \mid 3)$.

We will prove that the game has truly this form by induction on the size of the heap. For n such that $2 \leqslant n \leqslant a$ Right player has no moves and the first player can move to $\mathcal{C}(1) + \mathcal{C}(n-1) = n-2$ which dominates all his other moves. Suppose now that the statement holds for all heaps of size less than $n > a$ and consider a heap of size $n = 1 + ka + \ell$ for $\ell \in [0, a)$.

(a) We can write Left's moves as a sequence $(c_1, c_2, \ldots, c_{n-1}) = (\mathcal{C}(1)+\mathcal{C}(n-1), \mathcal{C}(2)+\mathcal{C}(n-2), \ldots, \mathcal{C}(n-1)+\mathcal{C}(1))$. We will consider the difference between consecutive values in this sequence is $(d_i) = (c_i - c_{i+1})_{i \geqslant 1}$ and apply Observation 2.

(i) For $\ell = 0$ is the difference $(d_i) = (0, 0, \ldots, 0, \Delta - \Delta, 0, \ldots, 0) = (0, 0, \ldots, 0)$ so the value is same no matter which move the player chooses. In the canonical form Left player has available a single move into $\mathcal{C}(1) + \mathcal{C}(n-1) = \mathcal{C}(n-1) = \mathcal{C}(1 + (k-1)a + a - 1) \overset{\text{IH}}{=} (k-1)\cdot\Delta + a - 1$.

(ii) For $\ell > 0$, the difference is $(d_i) = (0, 0, \ldots, 0, 1 + \Delta, 0, \ldots, 0, -\Delta - 1, 0, \ldots)$ where the first non-zero value is $(\ell+1)$-th in the sequence. Then we conclude (by domination) that Left player has available only a single move, which is into

$$\begin{aligned}\mathcal{C}(2+\ell) + \mathcal{C}(n-\ell-1) =& \; \mathcal{C}(n-1) + 1 + \{a-1|0\} \\ =& \; \mathcal{C}(ka+\ell) + 1 + \{a-1|0\} \\ \overset{\text{IH}}{=}& \; k\cdot\Delta + \ell - 1\end{aligned}$$

(b) Right can move into $\mathcal{C}(x_0)+\mathcal{C}(x_2)+\ldots+\mathcal{C}(x_a)$ where $x_0+\ldots+x_a=n$. We will show that the move $M=\mathcal{C}(1)+\mathcal{C}(1)+\ldots+\mathcal{C}(1)+\mathcal{C}(n-a)$ dominates all the others. This leads directly from the observation 2: the move M can be transformed into any valid move by a sequence of increments of some smaller heap while decrementing the biggest heap only. The value of the games in this sequence can change only when either of these heaps crosses the boundary of a block. When the smaller heap crosses the boundary upwards (in size), the value decreases by $-(1+\Delta)$. When the biggest heap crosses the boundary downwards, the value increases by $1+\Delta$. Now observe that because the movement downwards happens only on the biggest heap, there needs to be a downward cross of the boundary for each upward cross of the boundary. And because $k\cdot X+\ell < k\cdot X+\ell+(1+X)$ for any $k,\ell \geqslant 0$, we can say that the move M dominates all other moves. Therefore Right player has available only a single move, which is into $M=\mathcal{C}(n-a)=\mathcal{C}(1+(k-1)a+\ell)\overset{\text{IH}}{=}(k-1)\cdot\Delta+\ell$.

Now there remains to check that the canonical forms matches the premise that $\mathcal{C}(n)=\mathcal{C}(1+ka+\ell)=k\cdot\Delta+\ell$:

(a) For $\ell=0$, we get $\mathcal{C}(n)=\{(k-1)\cdot\Delta+a-1 \mid (k-1)\cdot\Delta\}=k\cdot\Delta$ by definition of Δ.

(b) For $\ell>0$, we get $\mathcal{C}(n)=\{k\cdot\Delta+\ell-1 \mid (k-1)\cdot\Delta+\ell\}=k\cdot\Delta+\ell$.

□

Theorem 4.55. The game $\mathcal{PB}(1 \mid 1,2,a_1,a_2)$ with any $a_i \geqslant 2$ is arithmetic periodic with the following canonical form sequence

$$\mathcal{C}\text{-sequence}(\mathcal{PB})=\overline{0\,*}\quad(+{\downarrow}).$$

Proof. We need to prove that the canonical form of a heap of size n equals

$$\mathcal{C}(n)=\begin{cases}{\downarrow}k & \text{for } n=2k+1,\\ {\downarrow}(k-1){*} & \text{for } n=2k.\end{cases}$$

We prove this by induction on n. First observe that $C(0)=C(1)=0$ because there are no splitting moves available. Also notice that

$$\mathcal{C}(n)-\mathcal{C}(n-1)=\begin{cases}{\downarrow}{*} & \text{for } n=2k+1,\\ {*} & \text{for } n=2k.\end{cases}\tag{4.2}$$

This relationship of neighboring heaps is shown in the following table:

$$\overset{*}{\curvearrowright}\ \overset{\downarrow *}{\curvearrowright}\ \overset{*}{\curvearrowright}\ \overset{\downarrow *}{\curvearrowright}\ \overset{*}{\curvearrowright}\ \overset{\downarrow *}{\curvearrowright}\ \overset{*}{\curvearrowright}\ \overset{\downarrow *}{\curvearrowright}\ \overset{*}{\curvearrowright}$$

n	1	2	3	4	5	6	7	8	9	...
$\mathcal{C}(n)$	0	$*$	$\downarrow$	$\downarrow *$	$\downarrow 2$	$\downarrow 2*$	$\downarrow 3$	$\downarrow 3*$	$\downarrow 4$	...

$$\underset{*}{\curvearrowleft}\ \underset{\uparrow *}{\curvearrowleft}\ \underset{*}{\curvearrowleft}\ \underset{\uparrow *}{\curvearrowleft}\ \underset{*}{\curvearrowleft}\ \underset{\uparrow *}{\curvearrowleft}\ \underset{*}{\curvearrowleft}\ \underset{\uparrow *}{\curvearrowleft}\ \underset{*}{\curvearrowleft}$$

Figure 4.3: Canonical form sequence of $\mathcal{PB}(1 \mid 1, 2)$.

Now suppose that the statement holds for all heaps of size smaller than n. Now consider the following cases:

(a) Let $n = 2k + 1$ for some $k > 0$.
Left can move to $\mathcal{C}(1) + \mathcal{C}(2k), \mathcal{C}(2) + \mathcal{C}(2k-1)$ and so on. By equation 4.2, the difference between these moves $\mathcal{C}(1+\ell) + \mathcal{C}(2k-\ell) - \mathcal{C}(1+\ell+1) + \mathcal{C}(2k-\ell-1) = (\mathcal{C}(1+\ell) - \mathcal{C}(1+\ell+1)) + (\mathcal{C}(2k-\ell) - \mathcal{C}(2k-\ell-1))$ equals $* - * = 0$ for any even $\ell \in [0, k)$ and $\uparrow * - \uparrow * = 0$ for any odd $\ell \in [0, k)$. In other words, all these moves have the same canonical form which by IH equals $\mathcal{C}(1) + \mathcal{C}(n-1) = \downarrow(k-1)*$.

Right can of course move to $\downarrow(k-1)*$ as well using the 1-cut.

Let us now consider the 2-cuts. One of them is $\mathcal{C}(1) + \mathcal{C}(1) + \mathcal{C}(2k-1) = \downarrow(k-1)$. By IH, any other such move is a sum of downs and stars (stars zero out themselves pairwise). let us consider any move to non-zero heaps $\mathcal{C}(a) + \mathcal{C}(b) + \mathcal{C}(c)$ such that $a + b + c = 2k + 1$. The number of downs equals to $\lfloor \frac{a-1}{2} \rfloor + \lfloor \frac{b-1}{2} \rfloor + \lfloor \frac{c-1}{2} \rfloor \leqslant \lfloor \frac{2k+1-3}{2} \rfloor = \lfloor \frac{2(k-1)}{2} \rfloor = k - 1$. Because $\Downarrow < \downarrow$ and $\Downarrow * < \downarrow *$, the moves to $\downarrow(k-1)$ and $\downarrow(k-1)*$ dominate all the others. Furthermore the move to $\downarrow(k-1)*$ reverses through $\downarrow(k-1)$.

We can see that all a-cuts for $a \geqslant 2$ are also dominated by this move.

This implies that $\mathcal{C}(n) = \{\downarrow(k-1)* \mid \downarrow(k-1)\} = \downarrow k$.

(b) Let $n = 2k$ for some $k > 0$.
Left can move to $\mathcal{C}(1) + \mathcal{C}(2k-1), \mathcal{C}(2) + \mathcal{C}(2k-2)$ and so on. The difference between consecutive games in this sequence by equation alternates 4.2 $(-\downarrow, +\downarrow, -\downarrow, \ldots)$ with $\mathcal{C}(1) + \mathcal{C}(2k-1) = \downarrow(k-1)$ and $\mathcal{C}(2) + \mathcal{C}(2k-2) = \downarrow(k-2)$. The latter dominates the former.

Right can also use Left's move $\downarrow(k-1)$ and also has a move to $C(1) + \mathcal{C}(1) + \mathcal{C}(2k-2) = \downarrow(k-2)*$. Now consider any move which breaks down the heap into three heaps $\mathcal{C}(a) + \mathcal{C}(b) + \mathcal{C}(c)$ such that $a + b + c = 2k + 1$. The number of possible downs equals $\lfloor \frac{a-1}{2} \rfloor + \lfloor \frac{b-1}{2} \rfloor + \lfloor \frac{c-1}{2} \rfloor \leqslant \lfloor \frac{2k-3}{2} \rfloor = k-2$ so all these moves are dominated by $\downarrow(k-1)$. Furthermore, $\downarrow(k-2)*$ reverses through $\downarrow(k-2)$.

By the same argument, all higher-order cuts are also dominated, therefore we can conclude that $\mathcal{C}(n) = \{\downarrow(k-2) \mid \downarrow(k-1)\} = \downarrow(k-1)*$.

□

***Conjecture* 12.** *We conjecture that the game $\mathcal{PB}(1 \mid 1, 2k, a_1, a_2, \ldots)$ with $k \geqslant 1, a_i \geqslant 2k$ has arithmetic periodic canonical form sequence:*

$$\mathcal{C}\text{-}sequence(\mathcal{PB}) = \overline{\underbrace{0 * 0 * \ldots 0 *}_{2k}} \quad (+\downarrow).$$

Reduced Canonical Forms

Although we were able to show the canonical form of several subclasses of Pure Breaking games, it seems that for the majority these games is the canonical form rather too complex to be able to give much information about the game. On the other hand, tools like temperature theory, atomic weights and relatively new reduced canonical forms reveal a great deal of information about some particular games.

Theorem 4.56. The game $\mathcal{PB}(1 \mid 2, 3)$ is arithmetic periodic with the following reduced canonical form sequence

$$\textit{rcf}\text{-sequence}(\mathcal{PB}) = \begin{array}{ccccccccccc} \scriptstyle 0 & \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 & \scriptstyle 5 & \scriptstyle 6 & \scriptstyle 7 & \scriptstyle 8 & \scriptstyle 9 \\ 0 & 0 & 1 & \alpha & \beta & \overline{1} & \overline{1} & \overline{1{+}\alpha} & \overline{1{+}\alpha} & \overline{1{+}\gamma} \end{array} \quad (+1)$$

where $\alpha := \{1 \mid 0\}$, $\beta := \{2 \mid 0\}$ and $\gamma := \{2 \mid 1 \parallel 0\}$.

Proof. We will prove the statement by induction on n. The first four values of the pre-period are trivial and $\mathcal{C}(4) = \{2, \alpha \mid 0, 1\} = \{2 \mid 0\}$ by dominance. For $n \geqslant 5$, we will show that Right's move $H_2 + H_{n-2}$ and Left's move $H_1 + H_1 + H_1 + H_{n-3}$ inf-dominate all their other moves.

(a) Left player's move $\Lambda := H_2 + H_{n-2} = 1 + H_{n-2}$. To show that this move dominates any other, let us analyse rcf value of any other possible move $\Gamma := H_{a_1} + H_{a_2}, a_1 + a_2 = n, a_i \neq 2$. let us write $a_i = 5k_i + \ell_i$ for $0 \leqslant \ell_i < 5$. By the induction hypothesis we can assume that these values have numeric component $\lfloor \frac{a_i}{5} \rfloor = k_i$ and a possible non-numeric component α, β, γ which we will refer to as *special.* Let us analyze any possible combination of these special components of Γ alongside any possible values $a_1 + a_2 \pmod 5$ in Table 4.2. We will write over-lined modulus if $a_1 \bmod 5 + a_2 \bmod 5 \geqslant 5$ to be able to determine the sum $k_1 + k_2$ in the following case-analysis. Let us also calculate the left and right stops for all these moves which will help with determining the inf-dominance.

Table 4.2: Combinations of game *rcf*-values for $\mathcal{PB}(1 \mid 2,3)$

Game	Modulus	LS	RS
0	$0,1,2$	0	0
$\alpha := \{1 \mid 0\}$	$2,3,4$	1	0
$\beta := \{2 \mid 0\}$	$\overline{0},\overline{1}$	2	0
$\gamma := \{2 \mid 1 \parallel 0\}$	$\overline{0},4$	1	0
$\alpha{+}\alpha = 1$	$\overline{0},\overline{1},\ 4$	1	1
$\alpha{+}\beta = \{3 \mid 2 \parallel 2 \mid 1\}$	$\overline{1},\overline{2}$	2	2
$\alpha{+}\gamma \equiv_{inf} \{2 \parallel 1 \mid 0\}$	$\overline{1},\overline{2}$	2	1
$\beta{+}\beta = 2$	$\overline{3}$	2	2
$\beta{+}\gamma \equiv_{inf} \{4 \mid 3 \parallel 2 \parallel\!\mid 2 \mid 1 \parallel 0\}$	$\overline{3}$	2	2
$\gamma{+}\gamma \equiv_{inf} \{2 \mid 1\}$	$\overline{3}$	2	1

We will show the inf-domination by proving that $R(\Lambda - \Gamma) \geqslant 0$ for any such Γ. Consider the following cases (the symbol $\equiv_5$ denotes equivalence modulo 5) in Figure 4.4:

$$(n \equiv_5 0) \to k + \alpha - \begin{cases} k + \alpha \\ (k-1) + \begin{cases} 1 \\ \beta \\ \gamma \end{cases} \end{cases} \qquad (n \equiv_5 1) \to k + \begin{Bmatrix} \beta \\ \gamma \end{Bmatrix} - \begin{cases} k \\ (k-1) + \begin{cases} 1 \\ \beta \\ \alpha + \beta \\ \alpha + \gamma \end{cases} \end{cases}$$

$$(n \equiv_5 2,3) \to k + 1 - \begin{cases} k + \begin{cases} 0 \\ \alpha \end{cases} \\ (k-1) + \begin{cases} 2 \\ \alpha + \beta \\ \alpha + \gamma \\ \beta + \gamma \\ \gamma + \gamma \end{cases} \end{cases} \qquad (n \equiv_5 4) \to 1 + \alpha - \begin{cases} 1 \\ \alpha \\ \gamma \end{cases}$$

Figure 4.4: Case analysis for the game $\mathcal{PB}(1 \mid 2,3)$.

Let us now analyze the non-obvious cases of $R(\Lambda - \Gamma)$:

- $R(1+\alpha-\beta) \geqslant R(1)+R(\alpha-\beta) = R(1)+R(\{1 \mid 0 \parallel -1 \mid -2\}) = 0$,
- $R(1+\alpha-\gamma) \geqslant R(1)+R(\alpha)-L(\gamma) = 1+0-1 = 0$,
- $R(1-\alpha) \geqslant R(1)-L(\alpha) = 1-1 = 0$.
- $R(1{+}\beta{-}\alpha{-}\gamma) \geqslant R(1){+}R(\beta{-}\alpha){+}R(-\gamma) = 1{+}R(\{2 \mid 1 \parallel 0 \mid -1\}){-}1 = 0$,
- $R(1{+}\gamma{-}\alpha{-}\beta) = R(\{2 \mid 0,2 \mid 1,1 \mid 0 \mid 1 \mid 0 \parallel -1 \mid -2,0 \mid -1,-1 \mid -2\}) = 0$,
- $R(2-x-y) \geqslant 2-L(x+y) = 2-2 = 0$, for any $x,y \in \{\alpha,\beta,\gamma\}$.

To conclude, we have shown that for any move Γ we have $R(\Lambda - \Gamma) \geqslant 0$. Therefore, every move is inf-dominated by Λ.

(b) Right's move $H_1 + H_1 + H_1 + H_{n-3} = H_{n-3}$ domination can be showed by using exactly the same technique. However, analyzing all triples and quadruples of possible special values is quite tedious. therefore we did this part of the proof using computer evaluation in *cgsuite*. This also motivated a new technique of proving the numeric-arithmetic periodicity, which is described in the following section and which allows us to prove this theorem with much more ease.

□

4.3.3 Periodicity of Partizan Taking and Breaking Games

Theorem 4.57. Let $\Gamma = \mathcal{PB}(\mathcal{S}_L, \mathcal{S}_R)$ be a pure breaking game. The (reduced) canonical form sequence of the game is **integer arithmetic periodic** with pre-period $n_0 \geqslant 0$, period $p \geqslant 1$ and saltus $s \geqslant 0$ if and only if

$$\forall n : n_0 + p \leqslant n < k(n_0 + p) \text{ we have } \mathcal{C}(n) = \mathcal{C}(n-p) + s,$$

where $k := \mathit{max}\, s + 1$ for all $s \in \mathcal{S}_L, \mathcal{S}_R$.

Proof. Let $n \geqslant k(n_0 + p)$. Then

$$\mathcal{C}(n) = \{H_{a_1} + H_{a_2} + \ldots + H_{a_\ell} \mid H_{b_1} + H_{b_2} + \ldots + H_{b_r}\},$$

where a_i, b_i, ℓ and r range over

$$\ell \in \mathcal{S}_L, r \in \mathcal{S}_R, a_i, b_i \geqslant 0 \text{ and } \sum_{i=1}^{\ell} a_i = \sum_{i=1}^{r} b_i = n.$$

Since $n \geqslant k(n_0 + p)$, in each considered move there is $a_i \geqslant n_0, b_j \geqslant n_0$ for some i, j. Without loss of generality, there is always $a_1, b_1 \geqslant n_0$. Now, let us consider a position $n - p$. We can therefore write

$$\mathcal{C}(n-p) = \{H_{a_1-p} + H_{a_2} + \ldots + H_{a_\ell} \mid H_{b_1-p} + H_{b_2} + \ldots + H_{b_r}\}$$

where a_i, b_i, ℓ and r range over the same values as in the description of $\mathcal{C}(n)$ above. Now, by the induction hypothesis, we have $\mathcal{C}(a_1) = \mathcal{C}(a_1 - p) + s$ and $\mathcal{C}(b_1) = \mathcal{C}(b_1 - p) + s$. So by using the Number avoidance theorem we can write

$$\begin{aligned} \mathcal{C}(n) &= \{s + H_{a_1-p} + H_{a_2} + \ldots + H_{a_\ell} | s + H_{b_1-p} + H_{b_2} + \ldots + H_{b_r}\} \\ &\overset{\text{NA}}{=} s + \{H_{a_1-p} + H_{a_2} + \ldots + H_{a_\ell} | H_{b_1-p} + H_{b_2} + \ldots + H_{b_r}\} \\ &= s + \mathcal{C}(n). \end{aligned}$$

□

Corollary 4.58. This provides us with an algorithm that solves a integer arithmetic periodic partial pure breaking game in finite time. Note that the computation time depends on the complexity of the (reduced) canonical forms of the first $n_0 + p$ games.

The following table summarizes the results of using this theorem on several simpler taking and breaking games.

Table 4.3: Computational results on integer arithmetic periodicity of pure breaking games.

Game	$\mathbf{n_0}$	p	s
$\mathcal{PB}(1 \mid 2,3)$	5	5	1
$\mathcal{PB}(1 \mid 2,3,4)$	5	6	1
$\mathcal{PB}(1 \mid 2,4)$	1	16	3
$\mathcal{PB}(1 \mid 2,3,[4],5)$	7	7	1
$\mathcal{PB}(1 \mid 2,3,4,5,6)$	9	8	1
$\mathcal{PB}(1 \mid 2,4,5)$	9	18	3
$\mathcal{PB}(1 \mid 2,3,4,6)$	5	8	1
$\mathcal{PB}(1 \mid 2,3,6)$	1	8	1
$\mathcal{PB}(1 \mid 2,5)$	1	18	3
$\mathcal{PB}(1 \mid 2,6)$	1	12	2
$\mathcal{PB}(1 \mid 3,4)$	13	7	2
$\mathcal{PB}(1 \mid 3,4,5)$	7	8	2
$\mathcal{PB}(1 \mid 4,5)$	25	9	3
$\mathcal{PB}(1 \mid 4,6)$	9	10	3
$\mathcal{PB}(1,2 \mid 2)$	1	36	11

4.3.4 Black and White Taking and Breaking Games

As described in Section 3.1.4, there is a straightforward way of generalizing an impartial game to a partizan game by assigning one of two colors to each position and of the game as well as to both players and allowing player to move only to a position of his color [16]. In this section, we will discuss some of possible black-white generalizations of taking-breaking games.

Definition 4.59. Let $\mathcal{S}$ be a set of integers. We call the **chessboard subtraction game** a subtraction game where Left (resp. Right) player can take $s \in \mathcal{S}$ token from a heap of size n only if $n - s$ is odd (even resp.).

n	1	2	3	4	5	6	7	8	9	...
$\mathcal{C}(n)$	0	1	1/2	3/4	5/8	11/16	21/32	43/64	85/128	...

Figure 4.5: Canonical sequence of chessboard subtraction game $\mathcal{S}(1,2)$.

Definition 4.60. The n-th *Jacobsthal number* (OEIS A001045) [76] is defined as

$$J_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ J_{n-1} + 2J_{n-2} & \text{if } n > 1. \end{cases}$$

Lemma 4.61. For any $n > 1$ the following holds

$$J_n - 2J_{n-1} = \begin{cases} 1, & \text{if } n \text{ is even;} \\ -1, & \text{if } n \text{ is odd.} \end{cases}$$

Proof. By induction on n. $J_2 - 2J_1 = 1$. Let $n > 2$.

$$J_n - 2J_{n-1} = J_{n-1} + 2J_{n-2} - 2J_{n-1} = -(J_{n-1} - 2J_{n-2}),$$

so by induction hypothesis, the differences alternate 1 and -1. □

Lemma 4.62. For any $n > 1$ the following holds

$$J_n - 4J_{n-2} = \begin{cases} 1, & \text{if } n \text{ is odd;} \\ -1, & \text{if } n \text{ is even.} \end{cases}$$

Proof. Let $n > 1$. Then $J_n - 4J_{n-2} = J_{n-1} + 2J_{n-2} - 4J_{n-2} = J_{n-1} - 2J_{n-2}$ so by applying lemma 4.61 we get the stated result. □

Theorem 4.63. Let $\mathcal{S}(1, 2, a_1, a_2, \ldots)$ be a chessboard subtraction game with $a_i > 2$ for all $i \geqslant 1$. Then

$$\mathcal{C}(n) = \frac{J_n}{2^{n-1}}.$$

.

For first few values of this sequence, see Figure 4.5.

Proof. We will prove this by induction on n. Let us firs show that this sequence holds for the game $\mathcal{S}(1,2)$. Clearly, $\mathcal{C}(0) = 0$ and $\mathcal{C}(1) = 1$. Let $n > 1$.

- Let n be odd. Then

$$\mathcal{C}(n) = \{\mathcal{C}(n-1) \mid \mathcal{C}(n-2)\} \overset{\text{IH}}{=} \left\{ \frac{J_{n-1}}{2^{n-2}} \,\middle|\, \frac{J_{n-2}}{2^{n-3}} \right\} = \left\{ \frac{J_{n-1}}{2^{n-2}} \,\middle|\, \frac{2J_{n-2}}{2^{n-2}} \right\}$$

and by applying Lemma 4.61 and the Simplicity Rule 2.52 follows

$$\mathcal{C}(n) = \left\{ \frac{J_{n-1}}{2^{n-2}} \;\middle|\; \frac{J_{n-1}+1}{2^{n-2}} \right\} = \frac{2J_{n-1}+1}{2^{n-1}} = \frac{J_n}{2^{n-1}}.$$

- Let n be even. Then by application of a symmetrical argument we get

$$\begin{aligned}\mathcal{C}(n) &= \{\mathcal{C}(n-2) \mid \mathcal{C}(n-1)\} \overset{\text{IH}}{=} \left\{ \frac{2J_{n-2}}{2^{n-2}} \;\middle|\; \frac{J_{n-1}}{2^{n-2}} \right\} \\ &= \frac{J_n}{2^{n-1}}.\end{aligned}$$

It remains to show that the moves $A-\{a_1, a_2, \ldots\}$ will not affect this sequence. But all these moves are dominated for both players which can be observed by applying Lemma 4.62 and by the fact that when players have a move to $n-1$ or to $n-2$, then they can only use the moves from A to go to $n-1-2k$ or to $n-2-2k$ for some $k \geqslant 1$, respectively. □

Theorem 4.64. Let A be a set of integers with $1 \in A$ and $2k$ be the minimal even element of A (if exists). Then the Chessboard subtraction game $\mathcal{S}(A)$ has the following canonical sequence

$$\mathcal{C}(n) = \begin{cases} \mathcal{C}'\Big(2\Big\lfloor \dfrac{n}{2k} \Big\rfloor + 1\Big) & \text{if } n \text{ is odd;} \\ \mathcal{C}'\Big(2\Big\lfloor \dfrac{n}{2k} \Big\rfloor\Big) & \text{if } n \text{ is even,} \end{cases}$$

where $\mathcal{C}'$ is a canonical sequence of Chessboard subtraction game $\mathcal{S}(1,2)$.

Proof. We can easily observe that first $2k$ canonical values have the form $(01)^k$. The remaining of the proof is exactly the same as in Theorem 4.63. □

4.4 Algorithmic and Complexity Results

We start this section by describing *the game solving algorithms* which we used for analyzing the games presented in this thesis. Most of these algorithms were helping us in finding and verifying the results presented in the previous sections.

We focus here on the algorithms for impartial games, as the great Aaron Siegel's toolkit for analyzing combinatorial games called the *cgsuite* [70] was sufficient for our needs in the area of partizan games.

As we have seen in Chapter 3 the most of the solutions of heap games rely on periodicity of some of their sequences. The two most common approaches in resolving periodicity of a game are (a) resolving it for some subclass through combinatorial analysis, or (b) verifying the periodicity using some *periodicity theorem* (Theorem 3.29 for subtraction games, Theorem 3.73 for code-digit games in general, Theorem 3.85 for hexadecimal games, Theorem 3.89 for pure breaking games and Theorem 4.57 for some partizan heap games).

4.4.1 Finding the Period and Pre-period of a Nim-sequence

We will start by presenting an algorithm which for given code-digit game Γ and its nim-sequence of some length N, efficiently verifies if the game satisfies some periodicity test. Here we will describe the testing of simple periodicity of code-digit games (Theorem 3.73), since the testing of arithmetic periodicity can be easily reduced to it by observing that if a sequence is ultimately arithmetic-periodic, its discrete derivative is ultimately periodic.

Finding the period

We define a **longest common prefix** of positions k, l, in a sequence $(a)_{i=0}^{n}$ as the maximal length m such that $a_{k+i} = a_{l+i}$ for all $i \in [0, m)$. This number is usually denoted by $\mathrm{LCP}(k, l) = m$. We also define a **border** of a substring (or continuous subsequence) as its *own* both prefix and suffix. By **own** prefix (suffix) we mean that it cannot equal to the substring itself.

Suppose that we have generated all values of the nim-sequence in interval $[0, N)$ for some $N > 0$. Let k, t be the parameters of the code-digit game Γ from Theorem 3.73. Now we need to verify that there exist n_0 and p such that $\mathcal{G}(n + p) = \mathcal{G}(n)$ holds for all n with $n_0 \leqslant n \leqslant tn_0 + (t - 1)p + k$ where $tn_0 + (t - 1)p + k < N$.

This can be done by the calculating longest common prefix for all pairs $\{(i, N - i) : i \in [0, N - k)\}$. If there exists such i that the

$$t \cdot \mathrm{LCP}(i, N - i) \geqslant (t - 1) \cdot N + k,$$

it follows that the nim-sequence in question is periodic with period $p = N - \mathrm{LCP}(i, N-i)-i$. If there is no such i, we can safely say that $tn_0+(t-1)p+k > N$ and we need more values. Otherwise, we look for the highest value i because we need a period p to be minimal.

To efficiently find such maximal i that the inequality for $\mathrm{LCP}(i, N - i)$ holds, we notice that we are looking for the rightmost position in sequence such that the prefix from this position matches the suffix of length i and i is small enough. This is true when the *border* of substring from given position

Algorithm 1: Calculate the period.

Input: N, $\mathcal{G}[N]$ *Nim-sequence of length N*

Output: p *Period or -1 if the sequence is too short*

```
1  j ← −1
2  B[0] ← j                                          ◁ Initialize border array
3  for i from N − 1 downto 0 do
4      while j ≠ −1 and G[N − j − 1] ≠ G[i] do
5          j = B[j]                                  ◁ Jump to longest border
6      j ← j + 1
7      B[N − i] = j
8      if t · B[N − i] ⩾ (t − 1) · N + k, then       ◁ The border is large enough
9          return N − B[N − i] − i
10 return −1                                         ◁ The period cannot be determined yet
```

to the end of the sequence equals k. To find such position, we calculate the border array from right to left the same way as the well known *KMP* or *Z* algorithm does and stop when the border is large enough. See pseudocode 4.1 for details on this calculation.

Theorem 4.65. Algorithm 4.1 returns minimal period in $\mathcal{O}(N)$ time and space or decides that the sequence is too short to determine the period.

Proof. First we will show that the following invariant holds throughout the running time of the algorithm: in any time array B contains the length of longest border of substring $\mathcal{G}[N-k,\ldots,N-1]$ at positions $k \in [1,\ldots,N-i]$. For $N-i=1$ clearly the value of j gets to -1 on line 4 and 0 gets written in B[1]. By induction on $N-i$ we now assume that the values $\mathrm{B}[0],\ldots,\mathrm{B}[k-1]$ are correct. The value j contains the longest prefix matching a suffix of the sequence from previous position. If we can extend this suffix when current value $\mathcal{G}[i]$ matches the position before this suffix, we skip line 6 and write the extended value on line 9. Otherwise, we look for expandable suffix by always shortening the suffix length to the length of the border of such suffix until we find it or reach the end of the string at line 7. If there is an own suffix that matches some prefix from position i, it will not be missed this way, because we always jump to the longest matching suffix of shorter length.

Now observe the following invariant: the value i iterates from $N-1$ to 0 by step one. For each iteration the value of j can first strictly decrease on line 7 because all values in array B contain values less then length of current border. But since all values in border array are positive or -1 and the iteration stops at -1, variable j will have values from interval $[-1,N)$. Because on line

Algorithm 3: Calculate the pre-period.

Input: N, $\mathcal{G}[N]$, p — *Nim-sequence of length N and its period*

Output: n_0 — *Pre-period length*

```
1 n_0 ← N − p − k
2 while n_0 ⩾ 0 and G[n_0 − 1] = G[n_0 + p − 1] do
3     n_0 ← n_0 − 1                          ◁ We rotate the period
4 return n_0
```

8 it only increases by one, overall the value of j increases by N so it can also decrease by N. Therefore the variable j is changed N times at most. The time complexity $\mathcal{O}(N)$ simply follows from this. Since only memory used by this algorithm are arrays $\mathcal{G}$ and B which are both linear, the space requirement is also $\mathcal{O}(N)$. Since all values in array B are correct and we iterate from right to left, from discussion at the beginning of this section it follows that the returned value is also correct. □

Finding the pre-period Suppose that we have generated first N values of the sequence and we know that it is periodic with period p and pre-period n_0 such that $n_0 + p + k \leqslant N$. Given the first N values of the nim-sequence and period p, our task is to find such minimal pre-period n_0 so that for all $n_0 \leqslant n < N$ holds $\mathcal{G}[n] = \mathcal{G}[n+p]$. Pseudocode 4.2 shows a simple algorithm to find such n_0.

Theorem 4.66. Algorithm 4.2 returns minimal pre-period in $\mathcal{O}(N)$ time.

Proof. At the start of the algorithm we know that the nim-sequence is periodic with period p and that it could be decided by the nim-sequence of length N. Because $tn_0 + (t-1)p + k < N$ the pre-period is less than $(N - (t-1)p - k)/t$. Suppose it is shorter. Then from periodicity follows that $\mathcal{G}[n_0 - 1] = \mathcal{G}[n_0 + p - 1]$. This condition is checked by the algorithm at line 4 and n_0 is decremented while it holds. This can be understood by rotating the period until such rotation that allows to find the shortest pre-period. The cycle runs clearly in linear time so the time complexity is $\mathcal{O}(N)$. □

When we are given a code-digit game, we can first convert it to some simpler equivalent form based on equivalences mentioned in Section 4.2.1. If the code-digit game is a subtraction game, we can also check first some trivial cases following from theorems 3.54 and 3.52. If the subtraction set matches their requirements, we return the answer in constant time. We also

Algorithm 5: Determine period and pre-period for code-digit game.

Input: $\mathsf{D} = \mathsf{d}_0 \cdot \mathsf{d}_1\mathsf{d}_2\mathsf{d}_3 \ldots \mathsf{d}_\mathsf{k}$ *Code-digit game*
Output: n_0, p *Period and pre-period*

1 $N \leftarrow 2 \cdot \mathit{max}(k, t+1)$
2 **while** (*true*)
3     $\mathcal{G}[N] \leftarrow$ Calculate nim sequence up to length N by definition
4     $p \leftarrow$ *Find the period using algorithm* 4.1
5   **if** $p \neq -1$
6         $n_0 \leftarrow$ *Find the pre-period using algorithm* 4.2
7   **return** (n_0, p)
8     $N \leftarrow 2 \cdot N$

can calculate the greatest common divisor of the set and based on theorem 3.35 we divide all elements from the set to calculate only the interesting part of the sequence (we multiply n_0 and p by this common divisor at the end of this algorithm).

Unfortunately we do not know how long sequence we need to generate so that it is long enough to find out its period. To know this fact, we would first need to know the period and pre-period which is a "chicken-egg" problem. To solve this problem efficiently, we start with a sequence of length $2 \cdot \mathit{max}(k, t+1)$ and we double its size each time we get negative answer from algorithm 4.1. The final algorithm can be seen in pseudocode 4.3.

Theorem 4.67. Let $N = tn_0 + (t-1)p + k$. Algorithm 4.3 returns minimal period and pre-period in $\mathcal{O}(N)$ time and space for given code-digit game.

Proof. The correctness follows directly from algorithms 4.1 and 4.2 and the termination from theorem 3.73. Let $m = \mathit{max}(k, t+1)$ and let ℓ be minimal integer such that $2^\ell \cdot m \geqslant N$. When we omit the time to generate the nim-sequence, i-th iteration of the while cycle starting on line 2 takes $\mathcal{O}(2^{i+1})$ time and is executed exactly ℓ times. Because $\sum_{i=1}^{\ell} 2^{i+1} \cdot m = \mathcal{O}(2^{\ell+2} \cdot m)$, the running time of the period and pre-period finding part of algorithm is $\mathcal{O}(N)$. However, it takes $\mathcal{O}(N^t \cdot k)$ time to generate the nim-sequence which is for most cases the bottleneck of the algorithm. $\square$

Algorithm 7: Calculate $\mathcal{G}(n)$ for finite octal game with **sparse space**.

Input: n, $\mathcal{G}[0, \ldots, n-1]$ *Nim-sequence of length n*
Output: $\mathcal{G}[n]$

```
1 𝒜 ← {}
2 for H_a + H_b ∈ Options(H_n), a is rare
3     𝒜 ← 𝒜 ∪ {𝒢[a] ⊕ 𝒢[b]}            ◁ Calculates all common and some rare
4 c = min(𝒞\𝒜)                          ◁ The smallest common not in 𝒜
5 ℬ ← ℛ ∩ {0, 1, ..., c − 1}\𝒜          ◁ Rares that were not excluded and are < c
6 for H_a + H_b ∈ Options(H_n), a, b common
7     ℬ ← ℬ\{𝒢[a] ⊕ 𝒢[b]}
8     if ℬ = ∅ then break               ◁ Return value is common
9 return min ℬ ∪ {c}
```

4.4.2 Algorithm for Solving Octal Games

As we have seen in the previous section, the bottleneck of verifying the periodicity of some code-digit game Γ lies in the requirements on the length of computed nim-sequence. For octal games, we need to perform $\mathcal{O}(n^2 \cdot k)$ operations to calculate a nim-sequence of length N by the naïve algorithm, where k is the number of non-zero digits in the octal code.

in Section 3.4.2 we have described the sparse space phenomenon, which (based on Flammenkamp's results) is the state-of art approach to solving octal games [20].

Based on this theory, we describe an efficient algorithm that will calculate a new $\mathcal{G}$-value provided that all the previous $\mathcal{G}$-values in the sequence are computed.

Theorem 4.68. Algorithm 4.4 terminates and returns $\mathcal{G}(n), n > 0$ correctly for given finite octal game.

Proof. The termination comes directly from finiteness of the game (it has finite set of options). To prove correctness, we will consider the following cases:

a) $\mathcal{G}(n)$ is common. By definition of sparse space, the common values can originate only from the nim-sum of rare and common value, so the value c on line 6 is already the value we seek. Because the set $\mathcal{B}$ contains only rare values smaller than this value, they must all get excluded by lines 5 and 9, so line 11 will return the value c.

b) $\mathcal{G}(n)$ is rare. Because c is not excluded and is common, $\mathcal{G}(n) < c$ so $\mathcal{G}(n) \in \mathcal{B}$. Any rare excludent originates from either both rare values, which are handled on line 5 or both common which are handled on line 9. The minimal value of $\mathcal{B}$ must necessarily be $\mathcal{G}(n)$.

$\square$

4.4.3 Algorithm for Computing Nim-sequences of Code-digit Games

Let $\mathsf{D} = \mathsf{d_0}{\cdot}\mathsf{d_1 d_2 d_3} \ldots \mathsf{d_k}$ be a finite code-digit game. Let $t \geqslant 0$ be the maximal number of heaps that are created in a single move ($\mathsf{d_i} < 2^{t+1}$ for all i). The calculation of a nim-sequence of length N by definition takes $\mathcal{O}(N^t \cdot k)$, since we need to consider all moves.

The main issue with the naïve approach is that the running time is exponential in the number of heaps we create. In this section, we lower this upper bound by using a *dynamic programming* technique and we also propose several low-level optimizations for this new algorithm.

Let us first describe a data structure MexDS that stores a set of non-negative integers while supporting the following operations:

- MexDS.insert(integer x)
  Inserts non-negative integer x into the data structure.
- MexDS.insertAll(MexDs D)
  Inserts all non-negative integer present into the the structure D.
- $y \leftarrow$ MexDS.mex()
  Calculates the *mex* function of inserted values.
- MexDS.addXor(MexDS D, integer z)
  Inserts into the data structure all values $\{y \oplus z : y \in D\}$

Suppose g is the maximal $\mathcal{G}$-value that occurs in the sequence. Then it is easy to see that we can implement the data structure MexDS by using a bit vector that uses $\mathcal{O}(g)$ bits and the operations insertAll, mex and addXor run all in $O(g)$ time while the operation insert runs in $\mathcal{O}(1)$ time.

Let us now use this data-structure in a dynamic programming approach

Theorem 4.69. The algorithm 4.5 computes the nim-sequence of code-digit game Γ of length N in $\mathcal{O}(mgN + gN^2)$ time and $\mathcal{O}(tgN)$ space, where m is number of digit-bits in D and g is maximal $\mathcal{G}$-value of the computed nom-sequence.

Algorithm 9: Given a code-digit game, calculate first N $\mathcal{G}$-values.

Input: N, $\mathsf{D} = \mathsf{d_0{\cdot}d_1d_2d_3 \dots d_k}$ *Required length, code-digit game*
Output: $\mathcal{G}[0], \mathcal{G}[1], \dots, \mathcal{G}[N-1]$ *Computed nim-sequence*

```
1  for d = 0, 1, ..., t                           ◁For each number of heaps created d > 1
2      for n = 0, 1, ..., N − 1                   ◁For each position in nim-sequence n < N
3          D[d][n] ← MexDS.init()                 ◁ Initialize data structure in 2D array D.
4  for n = 0, 1, ..., N − 1
5      cur ← MexDS.init()
6      for each 2^d ∈ d_s ∈ D          ◁ For each type of move in the code-digit game D
7          cur.insertAll(D[d][n-s])
8      G[n] ← cur.mex()
9      for d = 1, 2, ..., t
10         for i = 1, 2, ... ⌊n/d⌋
11             D[d][n].addXor(D[d-1][n-i],G[i])
```

Proof. Termination is clear, since all cycles range over finite sets. Since each operation on the data structure MexDS runs in $\mathcal{O}(g)$ time, the cycle starting on line 1 runs in $\mathcal{O}(tgN)$ time and the cycle starting on line 4 runs in $\mathcal{O}(mgN + gN^2)$ time because in each iteration, the line 11 will be executed $\mathcal{O}(n)$ times at most. Since $t \leqslant m$, we get the total time $= \mathcal{O}(tgN + mgN + gN^2) = \mathcal{O}(mgN + gn^2)$. Furthermore, the array $\mathcal{G}$ takes $\mathcal{O}(N)$ space and since each instance of MexDS takes $\mathcal{O}(g)$ space, the 2D array of data structures D takes $\mathcal{O}(gtN)$ space, so the algorithm uses $\mathcal{O}(gtN)$ space in total.

For proving the correctness, we will first argue that at the end of the $(m+1)$-th iteration of the cycle starting at line 4 the data structure $D[d][n]$ contains the correct values for all $0 \leqslant n \leqslant m$ and $0 \leqslant d < t$. By correct values we mean all the possible nim-values that can be achieved by splitting the heap of size n into d non-empty heaps, provided that the $\mathcal{G}$-values are computed correctly. We show this by induction on n. At the first iteration, the cycle on line 10 does not admit any value so the structure $D[d][0]$ will be empty for all d. Suppose now that the structure contains correct values for all $n < m$ and for contradiction, let there be some splitting move $a_1, a_2, \dots, a_k$ of the heap of size n into k non-empty heaps $(0 \leqslant k \leqslant t)$ that is not computed correctly. But since the sum of these values is n and they are positive, there must exist $a_j \leqslant \left\lfloor \frac{n}{k} \right\rfloor$. Since the values of k and a_j are considered on line 11, by induction hypothesis, this move has to be computed correctly, a contradiction.

Now it remains to show that we took into account each possible move while calculating some $\mathcal{G}[n]$. For contradiction, suppose that some lexicographically smaller move $H_{a_1} + H_{a_2} + \ldots + H_{a_d}$ based on some digit $2^d \in \mathsf{d}_s$ was not considered in the calculation of the mex for some $\mathcal{G}[n]$ on line 8. Then the structure $D[d][n-s]$ did not contain the mex of this move, a contradiction. $\square$

Optimizing the MexDS Structure

In the description of an optimization of the MexDS structure, we will use the RAM model with limited size of integers. We will denote the maximum number of such integer w and call it the **word size**. For simplicity, we expect w to be a power of two. We will also assume that the integers support constant-time operations AND, OR, NOT, LSB (least significant bit) and shift operations represented by $\ll$ and $\gg$.

Let g be again the maximum $\mathcal{G}$-value that appears in the nim-sequence. We will show that we can optimize the operations of the MexDS structure so that the operations insert and mex will take $\mathcal{O}(g/w)$ time and the addXor operation will take $\mathcal{O}(g\log(g)/w)$ time.

We can represent the bit vector of the original implementation of the MexDS data structure as an integer vector of length $m = \lceil \frac{n}{w} \rceil$, denoted $(a_0, \ldots, a_{m-1})$. For purposes that will become clear soon, we set initially all bits to one. The insert still runs in $\mathcal{O}(1)$ time by setting the corresponding bit to zero. The insertAll operations can be implemented as follows: $(a_0, \ldots, a_{m-1})$.insertAll$(b_0, \ldots, b_{m-1})$ assigns $a_i = a_i$ AND b_i for $i \in \{0, \ldots, m-1\}$, which runs in $\mathcal{O}(m)$ time. The operation *mex* will find the smallest i such that $a_i \neq 0$ and then returns $i \cdot w + \mathrm{LSB}(a_i)$. Finally, the operation addXor can be implemented as follows:

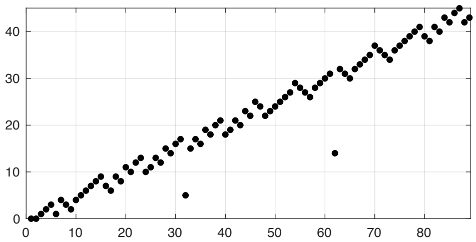
For $0 \leqslant k < \log_2(w)$, denote W_k such integer whose binary expansion alternates with blocks of ones and zeros of length 2^k, starting with ones. For instance, if $w = 8$ then $W_0 = (10101010)_2$, $W_1 = (11001100)_2$ and $W_2 = (11110000)_2$. Note that these values can be precomputed so we can access them in constant time. Now consider operation

$$(a_0, \ldots, a_{m-1}).\mathsf{addXor}(b_0, \ldots, b_{m-1}, 2^k) \quad \text{with } 0 \leqslant k < \log_2(w).$$

Then for each $i \in \{0, \ldots, m-1\}$ we assign

$$\begin{aligned} a_i = a_i \text{ AND } \Big(&\big((b_i \ll k) \text{ AND } W_k\big) \text{ OR} \\ &\big((b_i \gg k) \text{ AND NOT } W_k\big)\Big). \end{aligned}$$

We can observe that this operation swaps each two consecutive blocks of bits of length 2^k in b_i which is exactly the transformation that corresponds to applying a xor operation with the xor value 2^k to each position present in the data structure. To apply the same operation with $k \geqslant \log_2(w)$, we need to



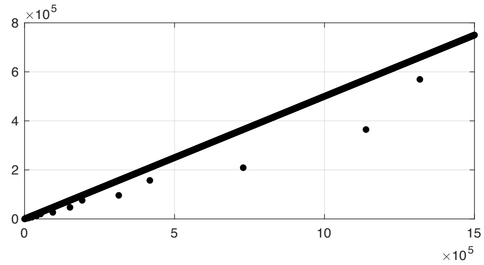


Figure 4.6: The nim-sequence of $\mathcal{PB}(1,2)$.

swap only consecutive blocks of integers of length $2^k/w$ in the representation by a vector of integers. Now observe that if we are applying the addXor operation with a xor value that is not a power of two, say $x = 2^{k_1} + 2^{k_2} + \ldots + 2^{k_\ell}$, we can successively apply the operations for each 2^{k_i} as described above. Each such transformation is independent so we can choose an arbitrary order of their execution. Since each such transformation with a xor value of a power of two takes $\mathcal{O}(g/w)$ time, the execution of the operation addXor will take $\mathcal{O}(g\log(g)/w)$ time in total.

If we apply this optimized data structure in Algorithm 4.5, we get the time complexity $\mathcal{O}(mgN/w + g\log(g)N^2/w)$. Note that the addXor operation is usually the bottleneck of the algorithm (typically $m \ll N$) so this optimization will have considerable effect only if $\log(g) \ll w$.

4.4.4 Computing Particular Code-Digit Game

Dailly et al. in [11] computed many pure breaking games which are described in Section 3.6. All of their computation showed arithmetic complexity of these games, except for one: the game $\mathcal{PB}(1,2)$ (in code-digit notation it is the hexadecimal game C·). Their investigation of the structure of this game suggests that all $\mathcal{G}$-values appear in the sequence exactly twice. Also, the sequence has interesting property that can be observed on the graphs of this sequence on figure 4.6: apart of small number of exceptions, the second appearance of the same nim-value is in the distance less than 6 from the previous. More importantly, this distance is always odd. The consequence is that the absolute difference between consecutive values is apart from the few exceptional values always 3 at most.

Table 4.4: Exceptional values in the nim-sequence of $\mathcal{PB}(1, 2)$.

$\mathcal{G}(n)$	First	Second	Distance
5	10	31	21
14	28	61	33
33	66	199	133
98	196	341	145
173	346	511	165
255	508	1,021	513
513	1,026	3,079	2,053
1,538	3,076	5,141	2,065
2,573	5,146	7,231	2,085
3,615	7,228	13,501	6,273
6,752	13,506	24,119	10,613
12,059	24,116	39,589	15,473
19,797	39,594	53,231	13,637
26,614	53,228	94,157	40,929
47,081	94,162	151,399	57,237
75,699	151,396	192,053	40,657
96,029	192,058	314,015	121,957
157,006	314,012	417,757	103,745
208,880	417,762	728,983	311,221
364,490	728,980	1,138,373	409,393
569,188	1,138,378	1,317,967	179,589

However, the exceptional values do not seem to show any regularity at all. The table 4.4 shows the first and second position of these exceptional values in the sequence and the distance between them. Although we do not bring any insight on the behavior of the nim-sequence of this game, we bring the computational result in the form of this table. Note that Dailly et al. in [11] report only 4,000 computed values, while we have achieved the computation of 1,500,000 values.

We will now describe the algorithm that achieved the computation of this number of values on an average laptop computer in a relatively short time. Note that if we used the naïve algorithm, we would need to perform approximately $3 \cdot 10^{18}$ operations. If we used the algorithm 4.5, since the $\mathcal{G}$-values grows half as fast as the size of heap, we would need to perform approximately $1.8 \cdot 10^{18}$ operations and if we would use the optimized version on a computer with a word size $w = 64$, we would need to perform circa $3 \cdot 10^{17}$ operations.

Algorithm 11: Given a code-digit game, calculate the nim-sequence values up to N.

Input: N, $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3 \ldots \mathsf{d}_k$ *Required length of sequence, code-digit game*
Output: $\mathcal{G}[0], \mathcal{G}[1], \ldots, \mathcal{G}[N-1]$ *Computed nim-sequence*

```
nimbers ← {0}                                ◁ Set of reasonable candidates for a nim-value
maxNimber ← 0
G[0] ← 0
for n = 1, 2, ..., N − 1
    cur ← null
    for g ∈ nimbers      ◁ For all nimbers that are not clearly among excludents
        if notExcluded(g)         ◁ Run procedure which finds if g is an excludent
            cur ← g
            break
    if cur = null                              ◁ New (higher) nimber encountered
        cur ← maxNimber + 1
        while notExcluded(cur)               ◁ Still could be excluded (by nim-sum)
            cur ← cur + 1
        nimbers ← nimbers ∪{cur}                ◁ New value can be an excludent
        maxNimber ← cur
    else
        nimbers ← updateNimbers(nimbers, cur)
    G[n] ← cur
```

The algorithm exploits the properties of the nim-sequence mentioned above. However, it can be easily generalized to other games where the $\mathcal{G}$-values do not repeat too often. The main idea lies in the proof of the Proposition 3.84 where we were quickly able to create a nim-value of almost any position. The condition of the proposition is not completely met (otherwise the game probably would not be much interesting). However, the key idea can be still partly applied.

Algorithm 4.6 describes main skeleton of our approach. The $\mathcal{G}$-values are computed one by one by testing *only reasonable* candidates for current value whether they *are excluded*.

Testing if a value is excluded is done by maintaining the current nimbers and their positions in a data structure composed by a binary rooted tree $T = (V, E, r)$ where $r \in V$ is root. Each vertex $v \in V$ of the tree corresponds to a binary prefix of a nim-value. Thus we can label the children of a vertex based on the value it appends to the prefix (0 or 1), denoted $\text{child}_0(v)$ and $\text{child}_1(v)$. Each vertex also maintains information about the minimal and maximal position of a nimber with a corresponding prefix, denoted $\min(v)$ and $\max(v)$.

The Algorithm 4.7 describes the procedure *notExcluded* (called on lines 7 and 12 in Algorithm 4.6) having such tree and the on the input. For each move that splits the heap into d non-empty heaps, it performs a depth-first search on state-space described as a d-tuple of vertices, each on the same level. The state corresponds to the prefixes of $\mathcal{G}$-values which in the nim-sum result in the prefix of g of the same length. The transition to a next state is done by considering all possible combination of the next bit in the prefix that would result in the prefix of the g value. Since this depends on the parity of sum of these next bits only, the number of such combinations can be exponential. A pruning of this search space is done by evaluating the maximum and minimum total size of heaps in the considered move and verifying it with the size of the heap we are splitting.

Even with such pruning, this search is highly inefficient if we were to perform it for each possible excludent. However, for games with "nice" properties (as the game $\mathcal{PB}(1,2)$) we can for many $\mathcal{G}$-values decide that they are excludents without such lengthy search. We will maintain the set of nimbers that are still valid candidates for some next $\mathcal{G}$-value. The procedure *updateNimbers* in Algorithm 4.7 should carry out this decision. We do not describe details of this procedure because it can be highly dependent on the analyzed game. But we will state an observation that implies the logic of such procedure for the game $\mathcal{PB}(1,2)$ and which led to our biggest speedup in the computation of its nim-sequence.

Observation 4.70. Suppose that the nimber g appeared already twice in the nim-sequence of $\mathcal{PB}(1,2)$, say at positions a and b ($a < b$), and suppose that $b - a$ is odd. Let $n \geqslant b + 2$. Then g is an excludent for the value $\mathcal{G}(n)$.

Proof. Since $b - a$ is odd, one of them must be even, say b. Than $H_{(n-b)/2} + H_{(n-b)/2} + H_b$ is an option of H_n and $\mathcal{G}((n-b)/2) \oplus \mathcal{G}((n-b)/2) \oplus \mathcal{G}(b) = \mathcal{G}(b) = g$. □

4.4.5 Computational Results on Subtraction Games

Inspired by the compendium of all subtraction games with numbers up to 7 provided by Guy et al. in [42, WW, p. 84], we have computed several properties of all subtraction games with numbers up to 30. There is $2^3 0 - 1 = 1,073,741,823$ such subtraction games. However, many of them have the same nim-sequence. We have applied the theory described in Section 4.1.1 to

Algorithm 13: Using a tree data structure holding nim-values, find if given value is excluded.

Input: g,$n \in \mathbb{N}_0$ tree $T = (V, E, r)$, max: $V \to \mathbb{N}$, min: $V \to \mathbb{N}$, $\text{child}_0(v) : V \to V$, $\text{child}_1(v) : V \to V$, $\mathsf{D} = \mathsf{d}_0{\cdot}\mathsf{d}_1\mathsf{d}_2\mathsf{d}_3 \ldots \mathsf{d}_\mathsf{k}$.
Output: True if and only if $\mathcal{G}(n) \neq g$.

```
for each 2^d ∈ d_s ∈ D              ◁ For each type of move in the code-digit game D
    if rec((r, ..., r), n − s, g)    ◁ A state is represented by a d-tuple of vertices
        return true
return false
procedure rec(state, n, g)
    minSum ← 0
    maxSum ← 0
    if nodes in state are leaves
        return whether exists a combination of positions that sums up to n
    for v ∈ state
        if min(v) = null
            return false
        else
            minSum ← minSum + min(v)
            maxSum ← maxSum + max(v)
    if minSum < n or maxSum > n
        return false
    for each binary d−tuple v whose parity of ones equals current bit in g
        newState ← empty d−tuple
        for i = 1, ..., d
            newState[i] = child_{v[i]}(state[i])
        if rec(newState, n, g)
            return true
    return false
```

calculate the master subtraction game for each encountered subtraction set. To speed up the computation, we marked each subtraction game for which we knew that it had the same nim-sequence as the one currently analyzing (based on its expansion set) as solved.

Another optimization has been done in calculation of the expansion set. The naïve algorithm to compute the expansion set clearly runs in $\mathcal{O}(N^2)$ where $N = n_0 + p + s_k$ (consequence of Observation 3.41). Let g be the maximal nimber in the sequence. To check if the property $\mathcal{G}(n) \neq \mathcal{G}(n+s)$ with $0 \leqslant n < n_0 + p + s_k$ when the $\mathcal{G}(n)$ can take only two possible values,

Table 4.5: Aggregated statistics for subtraction games.

	Lattice	Symmetric	Bipartite	Expandable	Singleton
Masters	90,393,070	23,013	7,681	51,544,131	88,501,672
	99.87 %	0.03 %	0.01 %	56.95 %	97.78 %
Total	93,567,644	149,761,490	9,400	204,341,428	88,501,672
	38.39 %	61.45 %	0.004 %	83.85 %	36.31 %

we can this simply calculate using a solution to a standard problem known as *Boolean convolution* [19]. In order to achieve only two types of values of $\mathcal{G}(n)$ we will compute a single Boolean convolution for each value $i \in \{0, 1, \ldots, g\}$. Thus we can compute the expansion set in time $\mathcal{O}(g \cdot N \log(N))$.

Here we present some statistics that we acquired during these computations. While examining these circa 10^9 distinct subtraction sets, we encountered only $90,509,909$ (8.43 %) distinct nim-sequences whose masters belonged to the examined category. In total 68.35% games had masters with higher maximal number than 30, in average by the factor of 4.7.

The masters with numbers up to 30 corresponded to $155,204,392$ subtraction games up to 30 in total. We also distinguished the following four properties of subtraction games and counted how many distinct subtraction games with these properties in this class exists: *lattices* (denotes that the equivalence class has a single source), *symmetric games*, *bipartite games* (ultimately bipartite, we required only that $p = 2$), *expandable games* and *singletons* (their equivalence class consist of a single game). Table 4.5 lists distribution of these properties across master games only and across all counted games. Table 4.6 lists the counts for all combinations of these properties that occurred among these games.

We also examined the following properties: *period*, *pre-period* (only lengths), *sources* (number of sources) and the *source range* (difference between the size of largest and smallest source). We looked for maximal values of these properties for each class of subtraction games $\mathcal{S}$ that have $\mathfrak{max}(\mathcal{S}) \in \{1, 2, \ldots, 30\}$. These statistics are listed in Table 4.7 and visualized in Figure 4.7.

4.4.6 Results in Complexity of Taking and Breaking Games

In this section, we examine a variant of subtraction game which is PSPACE-hard in its impartial form and EXPTIME-hard under the duel partizan generalization. This game was first introduced by Eppstein in [15, ch. 2.1]

Table 4.6: Statistics on subclasses of subtraction games.

Lat.	Sym.	Bip.	Unexp.	Singl.	Master count	Total Count
no	no	no	no	no	9,473	38,609
yes	no	no	no	no	329,508	694,844
no	yes	no	no	no	3	960
yes	yes	no	no	no	13	2,616
no	no	no	yes	no	92,568	1,138,261
yes	no	no	yes	no	1,554,466	3,569,434
no	yes	no	yes	no	15,028	148,959,045
yes	yes	no	yes	no	6,918	797,818
no	no	yes	yes	no	18	1,132
yes	no	yes	yes	no	493	1,260
yes	no	no	no	yes	38,627,194	38,627,194
yes	no	no	yes	yes	49,866,420	49,866,420
yes	yes	no	yes	yes	1,050	1,050
yes	no	yes	yes	yes	7,007	7,007
yes	yes	yes	yes	yes	1	1
					90,510,322	155,204,392

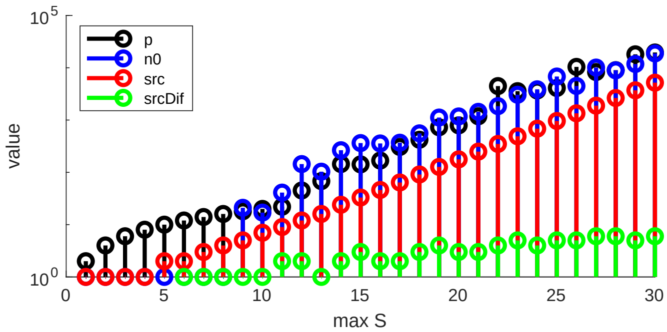


Figure 4.7: Graph of properties of masters of subtraction games based on their maximal number.

Definition 4.71. A **subtraction game with hot-spots** is a heap game defined by two sets $S(s_1, s_2, \ldots, s_k)$ and $\mathcal{H} = \{h_1, h_2, \ldots\}$, denoted $\mathcal{SH}(\mathcal{S}, \mathcal{H})$. In their moves, players can subtract any $s \in S$ or, if the size of heap equals to some $h \in \mathcal{H}$, remove the whole heap. Formally, the $\mathcal{SH}$-heap is defined by

$$H_n = \{H_{n-s} : s \in \mathcal{S}\} \cup \{H_0 \text{ if } n \in \mathcal{H}\}.$$

Such $\mathcal{SH}$-heaps H_n for which $n \in \mathcal{H}$ are called **hot-spots**.

Definition 4.72. A **subtraction game with succinct hot-spots** is a subtraction game with hot-spots where the set $\mathcal{H}$ is represented by a Boolean circuit a single input of $\log(n)$ bits, where n is maximal size of heap in the game, and a single output that tells if given heap-size is a hot-spot.

Table 4.7: Maximal properties of masters of subtraction games based on their maximal number.

$max(\mathcal{S})$	**Period**	**Pre-period**	**Sources**	**Sources range**
1	2	0	1	0
2	4	0	1	0
3	6	0	1	0
4	8	0	1	0
5	10	0	2	0
6	12	0	2	1
7	14	0	3	1
8	16	0	4	1
9	18	21	5	1
10	20	17	7	1
11	22	41	9	2
12	45	144	12	2
13	68	103	16	1
14	144	265	24	2
15	142	363	33	3
16	168	357	46	2
17	306	369	64	2
18	420	557	91	3
19	728	1,117	127	4
20	792	1,173	177	3
21	1,180	1,437	249	3
22	4,464	1,845	349	4
23	3,570	3,047	489	5
24	3,680	3,872	684	4
25	4,104	6,800	960	5
26	10,400	4,484	1,345	5
27	8,280	10,092	1,884	6
28	9,006	8,923	2,640	6
29	18,090	11,797	3,700	5
30	19,600	18,784	5,185	6

Such representation is commonly known as the **succinct** representation [59, ch. 20]. Fenner and Rogers [16, p. 31] analyzed the complexity of several Poset Games that have their positions described by this representation.

We will denote an instance of the Outcome problem for a subtraction-game with succinct hot-spots by a triple $(\mathcal{S}, \mathcal{H}, N)$, where $\mathcal{S}$ is its subtraction set, $\mathcal{H}$ are its hot-spots and N is the size of the $\mathcal{SH}$-heap played on.

Theorem 4.73. The problem Outcome to given single-pile position of a subtraction game with succinct hot-spots, described by a non-negative integer n, a finite subtraction set $S(s_1, s_2, \ldots, s_k)$ and a hot-spot set described using an AC^0 circuit is PSPACE-hard.

Proof. We show that the problem of determining Outcome of Node Kayles is polynomial reducible to this game. Suppose that $G = (V, E)$ is an instance of Node Kayles. Let $n = |V|, m = |E|$ and let us denote the set of vertices as $V = \{u_0, u_1, \ldots, u_{m-1}\}$ and the set of edges as $E = \{e_1, \ldots, e_{m-1}\}$.

We construct an instance of subtraction game with succinct hot-spots $(\mathcal{S}, \mathcal{H}, N)$ as follows. Let us define the heap size of the constructed instance in a 4-ary notation

$$N = (\underbrace{2 \mid 2 \mid \ldots \mid 2}_{n} \mid \underbrace{2 \mid 2 \ldots 2}_{m} \mid)_4.$$

The subtraction set is then defined as follows

$$\mathcal{S} = \left\{ 3^{i+m} + \sum_{e_j = \{u_i, u_k\} \in E} 3^j \ : \ u_j \in V \right\} \tag{4.3}$$

and the set of hot-spots $\mathcal{H}$ as any number h such that if we denote h in 4-ary as

$$h = (a_0 \mid a_1 \mid \ldots \mid a_{m+n-1})_3$$

then there exists some $i \in \{0, 1, \ldots, m+n-1\}$ such that $a_i = 0$. In other words, hot-spots are all positions that contain a digit 0 in the $m+n$ least significant digits of their 4-ary representation.

Notice that the position N and the subtraction set $\mathcal{S}$ can be computed in polynomial time and space in $m+n$. We will describe how in the same time and space also construct an AC^0 Boolean circuit of polynomial size. The circuit will have $L = 2(n+m)$ inputs and a single output. Let us denote the inputs $b_1, b_2, \ldots, b_L$. In order to correctly compute if a size of heap on the input is an element of $\mathcal{H}$, we need to compute the following function:

$$\begin{aligned} &\neg((b_1 \vee b_2) \wedge (b_3 \vee b_4) \wedge \ldots \wedge (b_{2(n+m)-1} \vee b_{2(n+m)})) \\ =&((\neg b_1 \wedge \neg b_2) \vee (\neg b_3 \wedge \neg b_4) \vee \ldots \vee (\neg b_{L-1} \wedge \neg b_L)). \end{aligned}$$

This can clearly be done with an circuit of constant depth, provided that we have allowed the fan-in of the single OR gate to be $n+m$. The not gates are located only at the inputs, so this circuit is AC^0.

We will now show that G is a won game of Node Kayles if and only if the subtraction game with succinct hot-spots $(\mathcal{S}, \mathcal{H}, N)$ is a won game. Notice that for each move in G there is a corresponding move in $\mathcal{SH}$ (described by Equation 4.3) for which an act of removing a vertex v "marks" the vertex and its neighboring edges as used by subtracting 1 from their 4-ary digits. However, there are some moves in $\mathcal{SH}$ that do not correspond to moves in G. These are only those that "mark" a certain vertex or edge for the second time. By definition of the hot-spot set, if some player moves this "forbidden" moves, the other one instantly wins.

Now, let the first player has a winning strategy for G. Then he can easily apply it in the game $\mathcal{SH}$. If the other player would play a "forbidden" move, he would lose instantly. Let us assume he did not play any such move. Since first player has a winning strategy, the second player will be the first who will be out of moves that are not forbidden. If there were no forbidden moves left, he would lose. If there is a forbidden move, he plays it and the first player wins.

The converse implication can be proven symmetrically. □

Definition 4.74. A **partizan subtraction game with succinct hot-spots** is a duel partizan generalization of subtraction game with hot-spots. Each player has assigned their own subtraction set, denoted $\mathcal{S}_L$ and $\mathcal{S}_R$, and their own hot-spots, denoted $\mathcal{H}_L$ and $\mathcal{H}_R$, respectively. We denote a single-heap instance of this game as $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$.

Theorem 4.75. The problem Outcome for an instance of partizan subtraction game with succinct hot-spots $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$, described by a non-negative integer N, finite subtraction sets $\mathcal{S}_L$ and $\mathcal{S}_R$, and hot-spot sets $\mathcal{H}_L$ and $\mathcal{H}_R$ described using an AC^0 circuit, is $\mathsf{EXPTIME}$-hard.

Proof. We show that the problem of determining Outcome of the game G_4 by Sockmeyer and Chandra [78] described in Section 2.4.1 is polynomial reducible to this game.

Suppose that $F(X, Y)$ is a formula (in 13-DNF), $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_m)$ and $X \cap Y = \emptyset$. Let $\alpha : X \cup Y \to \{0, 1\}$ represents the current assignment of the variables. We construct an instance of the partizan subtraction game with succinct hot-spots $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$ as follows.

First notice that even though the pass moves are allowed in this game, we can expect that if some player has a winning strategy, he has also a winning strategy without redundant pass moves. This strategy takes at most $M = 2 \cdot 2^{n+m}$ moves. This is due to the fact that the state of the game is determined by the player who is on the move and the configuration of the variables, so there are only M distinct states. If the game took longer, there would need to exist some state that appeared twice. Since some player has a winning strategy, the play can not loop infinitely so the initiative in this move has to be his. Thus he must at some point choose another option from this move in order to win. Assume he will choose this option the first time he arrives to this state.

Now let us define the heap size of the constructed instance in a B-ary notation, where $B = 2M$.

$$N = \big(M \mid M + \alpha(x_1) \mid M + \alpha(x_2) \mid \dots \mid M + \alpha(x_n) \\ \mid M + \alpha(y_1) \mid M + \alpha(y_2) \mid \dots \mid M + \alpha(y_m)\big)_B$$

Now let us define the subtraction sets of Left and Right players as follows.

$$\mathcal{S}_L = \big\{M \cdot B^{n+m} + M \cdot B^{i+m} : i \in \{0, 1, \dots, n\}\big\} \cup \{M \cdot B^{n+m}\} \text{ and}$$
$$\mathcal{S}_R = \big\{M \cdot B^{n+m} + M \cdot B^{j} : j \in \{0, 1, \dots, m\}\big\} \cup \{M \cdot B^{n+m}\}.$$

We define the hot-spots $\mathcal{H}_L$ (H_R) by a set of number h with the following properties: If we denote h in B-ary as

$$h = (a_0 \mid a_1 \mid \dots \mid a_{m+n})_B.$$

then either:

(1) There exists some $i \in \{0, 1, \dots, m + n - 1\}$ such that $a_i = 0$ (the game takes at least M moves).

(2) There exists some $s \in \mathcal{S}_L$ ($\mathcal{S}_R$) such that if we denote $h - s$ in B-ary as

$$h - s = (z \mid x'_0 \mid x'_1 \mid \dots \mid x'_{n-1} \mid y'_0 \mid y'_1 \mid \dots \mid y'_{m-1})_B,$$

such that if we define the assignment α as $\alpha(x_i) = 1$ if and only if $x'_i \bmod 2 = 1$ for all $i \in \{0, 1, \dots, n - 1\}$ and $\alpha(y_j) = 1$ if and only if $y'_j \bmod 2 = 1$ for all $j \in \{0, 1, \dots, m - 1\}$, then the formula F is under α satisfied.

Notice that the position N and the subtraction sets $\mathcal{S}_L$ and $\mathcal{S}_R$ can be easily computed in polynomial time and space in $m + n$. We will describe how we can also construct an AC^0 Boolean circuit of polynomial size in the same time and space. The circuit will have $L = (n + m + 2)^2$ inputs and a single output. To test if a given number in B-ary provided in these inputs is a hot-spot, we proceed as follows. The check the existence of zero B-ary digit from the condition (1) can be done similarly as we have seen in the proof of Theorem 4.73. The difference is only in the number of gates and the fan-out, which can be here up to $n + m + 2$. The condition (2) can be also checked by an AC^0 circuit: the formula $F(X, Y)$ is in 13-DNF and their inputs are the least-significant bits of the corresponding B-ary digits of $h - s$. For H_L (H_R) we will need a single such circuit for each $s \in \mathcal{S}_L$, $\mathcal{S}_R$ respectively and the result of will be OR of outputs of these circuits. Also the calculation of the difference $h - s$ can be done by an AC^0 circuit when using the *two's complement* representation of numbers, as described in [3, p. 117].

Now, by the same argument as in the proof of Theorem 4.73 we can observe that each move in $F(X, Y)$ corresponds to some move in $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$, apart from the moves that are performed after M first moves. Also, we can see that all the moves in $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$ correspond to a move in $F(X, Y)$. By the argument that if some player has a winning strategy then he has also a winning strategy which takes M or less moves and by the definition of the hot-spot sets which assures that for each winning move in $F(X, Y)$ there is a winning move in $(\mathcal{S}_L, \mathcal{H}_L, \mathcal{S}_R, \mathcal{H}_R, N)$, the reduction is complete. □

Notice that the classical algorithm for solving the impartial variant of this game by evaluating the $\mathcal{G}$-values of each position runs in exponential time, so clearly this problem belongs to EXPTIME. Its presence in the PSPACE seems unlikely and so does the completeness of this problem in this class. Similarly, the partizan variant is surely in NEXPTIME, but its presence in EXPTIME also seems to be hard to prove. The difficulties lie in the fact that the input is *succinct* and so the naïve algorithm runs in doubly exponential time.

Conclusion and Future Work

In this thesis we have provided a survey of the field of impartial and partizan taking and breaking games under a normal form.

Main Contributions

We have divided our main contributions into three categories:

Subtraction Games We have provided an example of a subtraction game for which the outcome sequence is periodic and the nim-sequence is aperiodic. Then we have outlined a new theory for the analysis of the structure of equivalence classes of subtraction games. This led to complete classification of a subset of these classes that appear to be the most prominent: the games M-PLICATE BOUNDED NIM(k). We have shown a new infinite class of subtraction games which are ultimately bipartite and expandable. We have also provided several results on infinite subtraction games.

Code-Digit Games We have provided proofs for several equivalences among these games and described a new class of taking and breaking games (TAKE-K-BREAK-K) that we have partially solved.

Partizan Generalizations of Taking and Breaking Games We have shown proofs of canonical sequences for several partizan subtraction games, including the game $\mathcal{S}(1,2 \mid 1,3)$ conjectured by Plumbeck [61, ch. 3]. Then we have given several results on the partizan pure breaking games, a class of games that has been (until now) studied only under impartial setting. Then we have generalized the technique used for these proofs into a new periodicity test for integer arithmetic periodicity of these games. Using these test, we have been able to prove several other such games with integer-arithmetic periodic reduced canonical form sequence. We have also presented an analysis for all CHESSBOARD SUBTRACTION GAMES having the number 1 and some even number in their subtraction sets.

Algorithmic Combinatorial Game Theory Among our results in game solving algorithms we have presented a new algorithm based on dynamic programming that computes a nim-sequence for any code-digit game. We have also provided a new heuristic approach that worked well in computing the nim-sequence of a pure breaking game. Using our algorithms, we have computed all subtraction games with numbers up to 30 and have made several observations about their structure.

In computational complexity of combinatorial games, we have described a new generalization of subtraction games called *subtraction games with succinct hot-spots*. We have shown PSPACE-hardness of the OUTCOME problem for this game under impartial setting and EXPTIME-hardness for the same problem under partizan generalization.

Future Work

In general, answers always lead to more questions. We provide here a list of questions that have arisen during our work.

(1) Analyze the MULTIPLY SUBTRACTION GAMES: for any s in subtraction game, players can subtract any $k \cdot s$ with $k \geqslant 1$. Consider also partizan setting of this game.

(2) Computational results discussed in Section 4.4.5) suggest that the games from Theorem 4.23 are the games with maximal numbers of minimals. We showed that this holds at least for all games having all numbers in their masters up to 30. Does it hold in general?

(3) Analyze the MULTIPLY TAKE-K-BREAK-K GAMES: for any k in provided set K, players can subtract any $k \cdot s$ with $k \geqslant 1$ and then split the remainder into k non-empty heaps. Consider also partizan setting of this game.

(4) Consider a restricted code-digit games where only a limited number of types of non-zero digits are allowed. Some of them were already successfully studied: SUBTRACTION GAMES are restricted with digit 3, SPLITTLES with digit 7. Games restricted to digits $\{1, 2, 3\}$ are known as QUATERNARY GAMES.

(5) Show that $\mathcal{PB}(1 \mid 1, 2k, a_1, a_2, \ldots)$ with $k \geqslant 1, a_i \geqslant 2k$ is arithmetic periodic with the following canonical form sequence:

$$\mathcal{C}\text{-sequence}(\mathcal{PB}) = \overline{\underbrace{0 * 0 * \ldots 0 *}_{2k}} \quad (+{\downarrow}).$$

(6) Show that $\mathcal{PB}(1 \mid 2, 4)$ is arithmetic periodic with the following canonical form sequence:

$$\mathcal{C}\text{-sequence}(\mathcal{PB}) = \overline{0\ 1\ \{1|0\}\ \{2|1\}} \quad \left(+\left\{2 \mid 1 \,\|\, 0\right\}\right).$$

(7) Generalize the Theorem 4.57 for integer arithmetic periodicity of pure breaking games to code-digit games and find a class of these games that satisfy this periodicity.

Appendix A

Contents of Enclosed Media

```
/
├── readme.txt ............................ Media contents description
├── src
│   ├── impl .................................. Game testing programs
│   └── thesis ............................. Source files of this thesis
└── text
    └── thesis.pdf ....................... Text of this thesis in PDF
```

Bibliography

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

[2] Ingo Althöfer and Jörg Bültermann. "Superlinear period lengths in some subtraction games". In: *Theoretical computer science* 148.1 (1995), pp. 111–119.

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

[4] R. Austin. "Impartial and partisan games". MA thesis. Univ. of Calgary, 1976.

[5] E.R. Berlekamp, J.H. Conway, and R.K. Guy. *Winning ways for your mathematical plays.* vol. 3. A.K. Peters, 2003. ISBN: 9781568811444.

[6] Bryce Emerson Blackham. "Subtraction Games: Range and Strict Periodicity". MA thesis. 2018.

[7] Grant Cairns and Nhan Bao Ho. "Ultimately bipartite subtraction games". In: *Australas. J. Combin* 48 (2010), pp. 213–220.

[8] Ashok K Chandra and Larry J Stockmeyer. "Alternation". In: *17th Annual Symposium on Foundations of Computer Science (sfcs 1976).* IEEE. 1976, pp. 98–108.

[9] C.J. Colbourn and E.S. Mahmoodian. *Combinatorics Advances.* Mathematics and Its Applications. Kluwer Academic Publishers, 1995. ISBN: 9781461335542.

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

[11] Antoine Dailly et al. "Partition games are pure breaking games". In: *arXiv preprint arXiv:1803.02621, <hal-01723190>* (2018).

[12] Erik D Demaine et al. “Appendix B: open problems at the 2002 Dagstuhl seminar on algorithmic combinatorial game theory”. In: *Theoretical Computer Science* 313.3 (2004), pp. 539–543.

[13] Eric Duchêne and Michel Rigo. “Invariant games”. In: *Theoretical Computer Science* 411.34-36 (2010), pp. 3169–3180.

[14] Eric Duchene et al. *Wythoff Wisdom*. 2016.

[15] David Eppstein. *Faster Evaluation of Subtraction Games*. 2018. arXiv: `1804.06515 [cs.DS]`.

[16] Stephen A Fenner and John Rogers. “Combinatorial game complexity: an introduction with poset games”. In: *arXiv preprint arXiv:1505.07416* (2015).

[17] Stephen A Fenner et al. “Game values and computational complexity: An analysis via black-white combinatorial games”. In: *International Symposium on Algorithms and Computation*. Springer. 2015, pp. 689–699.

[18] Thomas S Ferguson. “On sums of graph games with last player losing”. In: *International Journal of Game Theory* 3.3 (1974), pp. 159–167.

[19] Michael J Fischer and Michael S Paterson. *String-Matching and Other Products*. Tech. rep. MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, 1974.

[20] Achim Flammenkamp. *Sprague-Grundy Values of Octal-games*. 2012. URL: `http://www.uni-bielefeld.de/~achim/octal.html` (visited on 10/20/2018).

[21] Achim Flammenkamp, Arthur Holshouser, and Harold Reiter. “Dynamic one-pile blocking Nim”. In: *the electronic journal of combinatorics* 10.1 (2003), p. 4.

[22] Nathan Fox. “On Aperiodic Subtraction Games with Bounded Nim Sequence”. In: *arXiv preprint arXiv:1407.2823* (2014).

[23] A Fraenkel. “Recreation and depth in combinatorial games”. In: *The Lighter Side of Mathematics: Proceedings of the Eugène Strens Memorial Conference on Recreational Mathematics and Its History*. Cambridge University Press. 1994, p. 159.

[24] Aviezri S Fraenkel. “Aperiodic subtraction games”. In: *the electronic journal of combinatorics* 18.2 (2011), p. 19.

[25] Aviezri S Fraenkel. “Complexity, appeal and challenges of combinatorial games”. In: *Theoretical Computer Science* 313.3 (2004), pp. 393–415.

[26] Aviezri S Fraenkel. “From Nim to Go”. In: *Annals of Discrete Mathematics*. Vol. 6. Elsevier, 1980, pp. 137–156.

[27] Aviezri S Fraenkel. “Recent results and questions in combinatorial game complexities”. In: *Theoretical computer science* 249.2 (2000), pp. 265–288.

[28] Aviezri S Fraenkel and Anton Kotzig. “Partizan octal games: partizan subtraction games”. In: *International Journal of Game Theory* 16.2 (1987), pp. 145–154.

[29] Aviezri S Fraenkel and David Lichtenstein. “Computing a perfect strategy for n× n chess requires time exponential in n”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 1981, pp. 278–293.

[30] Aviezri S Fraenkel, Martin Loebl, and Jaroslav Nešetřil. “Epidemiography II. Games with a dozing yet winning player”. In: *Journal of Combinatorial Theory, Series A* 49.1 (1988), pp. 129–144.

[31] Aviezri S Fraenkel and Yaacov Yesha. “Complexity of problems in games, graphs and algebraic equations”. In: *Discrete Applied Mathematics* 1.1-2 (1979), pp. 15–30.

[32] Anil Gangolli and Thane Plambeck. “A note on periodicity in some octal games”. In: *International Journal of Game Theory* 18.3 (1989), pp. 311–320.

[33] Solomon W Golomb. “A mathematical investigation of games of “take-away””. In: *Journal of Combinatorial Theory* 1.4 (1966), pp. 443–458.

[34] J. P. Grossman. *Combinatorial game Theory Worshop Program at Banff International Research Station*. 2011. URL: `http://www.birs.ca/workshops/2011/11w5073/Programme11w5073.pdf` (visited on 10/20/2018).

[35] JP Grossman and J Nowakowski Richard. “A ruler regularity in hexadecimal games”. In: *Games of No Chance* 4.63 (2015), p. 115.

[36] JP Grossman and Aaron N Siegel. “Reductions of partizan games”. In: *Games of No Chance III, Proc. BIRS Workshop on Combinatorial Games*. Citeseer. 2005, pp. 427–455.

[37] Alan Guo. “Winning strategies for aperiodic subtraction games”. In: *Theoretical Computer Science* 421 (2012), pp. 70–73.

[38] Alan Guo and Ezra Miller. “Algorithms for lattice games”. In: *International Journal of Game Theory* 42.4 (2013), pp. 777–788.

[39] Richard K Guy and Cedric AB Smith. “The G-values of various games”. In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 52. 3. Cambridge University Press. 1956, pp. 514–526.

[40] R.K. Guy. *Combinatorial Games*. AMS Short Course Lecture Notes. American Mathematical Soc., 1991. ISBN: 9780821867488. URL: `https://books.google.cz/books?id=R1sAvfl9Lo8C`.

[41] R.K. Guy. *Unsolved problems in number theory*. Problem books in mathematics. Springer-Verlag, 1981. ISBN: 9783540905936. URL: `https://books.google.com.nf/books?id=PxsTMQAACAAJ`.

[42] RK Guy and John Horton Conway. *Winning ways for your mathematical plays*. 1982.

[43] Nhan Bao Ho. "On the Expansion of Three-element Subtraction Sets". In: *Theor. Comput. Sci.* 582.C (May 2015), pp. 35–47. ISSN: 0304-3975. DOI: `10.1016/j.tcs.2015.03.025`. URL: `http://dx.doi.org/10.1016/j.tcs.2015.03.025`.

[44] Nhan Bao Ho. "Subtraction games with three element subtraction sets". In: *Preprint. Available at http://arxiv. org/abs/1202.2986 v1* (2012).

[45] S Howse and Richard J Nowakowski. "Periodicity and arithmetic-periodicity in hexadecimal games". In: *Theoretical computer science* 313.3 (2004), pp. 463–472.

[46] James P Jones. "Some undecidable determined games". In: *International Journal of Game Theory* 11.2 (1982), pp. 63–70.

[47] John Charles Kenyon. "Nim-like games and the Sprague-Grundy theory". In: (1967).

[48] U. Larsson. *Games of No Chance 5*. Mathematical Sciences Research Institute Publications. Cambridge University Press, 2019. ISBN: 9781108485807. URL: `https://books.google.cz/books?id=i7WQDwAAQBAJ`.

[49] Urban Larsson. "The -operator and invariant subtraction games". In: *Theoretical Computer Science* 422 (2012), pp. 52–58.

[50] Urban Larsson and Nathan Fox. "An Aperiodic Subtraction Game of Nim-dimension Two". In: *Journal of Integer Sequences* 18.2 (Mar. 2015), p. 3.

[51] Urban Larsson and Johan Wästlund. "From heaps of matches to the limits of computability". In: *the electronic journal of combinatorics* 20.3 (2013), P41.

[52] Shuo-Yen R Li. "Sums of Zuchswang games". In: *Journal of Combinatorial Theory, Series A* 21.1 (1976), pp. 52–67.

[53] Neil A McKay. "Canonical forms of uptimals". In: *Theoretical Computer Science* 412.52 (2011), pp. 7122–7132.

[54] GA Mesdal III. "Partizan splittles". In: *Games of No Chance III, Proc. BIRS Workshop on Combinatorial Games*. Citeseer. 2005, pp. 447–461.

[55] Richard J Nowakowski. *Games of no chance*. Vol. 29. Cambridge University Press, 1996.

[56] Richard J Nowakowski and Michael H Albert. *Games of no chance 3*. Cambridge University Press, 2009.

[57] Richard J Nowakowski and Michael H Albert. *Games of no chance 4*. Cambridge University Press, 2015.

[58] Richard Nowakowski and Silvio Levy. *More games of no chance*. Vol. 42. Cambridge University Press, 2002.

[59] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[60] Pranay Pherwani. *Analysis Of Subtraction Games With Infinite Subtraction Sets Of Powers*. 2018.

[61] Thane Plambeck. "Partisan subtraction games". In: *Working notes, February* (1995).

[62] Michael O Rabin. "Effective computability of winning strategies". In: (1958).

[63] José L Ramírez and Víctor F Sirvent. "A Generalization of the k-Bonacci Sequence from Riordan Arrays". In: *The Electronic Journal of Combinatorics* 22.1 (2015), pp. 1–38.

[64] Ph.D RNDr. Tomáš Valla. personal communication. Dec. 14, 2018.

[65] Carlos Pereira dos Santos and Jorge Nuno Silva. "Konane has infinite Nim-dimension". In: *Integers: Electronic Journal of Combinatorial Number Theory* 8.G02 (2008), G02.

[66] Thomas J Schaefer. "On the complexity of some two-person perfect-information games". In: *Journal of Computer and System Sciences* 16.2 (1978), pp. 185–225.

[67] Jonathan Schaeffer et al. "Checkers is solved". In: *science* 317.5844 (2007), pp. 1518–1522.

[68] Leo Schneider. "Winning Ways for your Mathematical Plays". In: *The Mathematics Teacher*. vol. 2 95.2 (2002).

[69] Claude E Shannon. "Programming a computer for playing chess". In: *Computer chess compendium*. Springer, 1988, pp. 2–13.

[70] Aaron N Siegel. *Cgsuite (combinatorial game suite). Software toolkit for analyzing combinatorial games*. 2015. URL: `http://www.cgsuite.org` (visited on 05/06/2019).

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

[72] Aaron Nathan Siegel. "Loopy games and computation". PhD thesis. University of California, Berkeley, 2005.

[73] Angela Siegel. "Finite excluded subtraction sets and Infinite Geography". MA thesis. Dalhousie Univ., 2005.

[74] Neil JA Sloane. *The on-line encyclopedia of integer sequences: Sequence A000045–Fibonacci numbers*. 2007. URL: `https://oeis.org/A000045` (visited on 04/30/2019).

[75] Neil JA Sloane. *The on-line encyclopedia of integer sequences: Sequence A000931–Padovan numbers*. 2007. URL: `https://oeis.org/A000931` (visited on 04/30/2019).

[76] Neil JA Sloane. *The on-line encyclopedia of integer sequences: Sequence A001045–Jacobsthal numbers*. 2007. URL: `https://oeis.org/A001045` (visited on 03/20/2019).

[77] Eric Sopena. "i-Mark: A new subtraction division game". In: *Theoretical Computer Science* 627 (2016), pp. 90–101.

[78] Larry J Stockmeyer and Ashok K Chandra. "Provably difficult combinatorial games". In: *SIAM Journal on Computing* 8.2 (1979), pp. 151–174.

[79] S. Vajda. *Mathematical Games and How to Play Them*. Dover Books on Mathematics. Dover Publications, 2008. ISBN: 9780486462776.