Imperfect information variants of combinatorial games

Bergh, M.J.H. van den

Imperfect information variants of combinatorial games

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden
op gezag van rector maginificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op donderdag 1 juni 2023
klokke 15.00 uur

door

Mark Jan Hugo van den Bergh
geboren te ’s-Gravenhage
in 1994

Promotor:
Prof.dr. F.M. Spieksma

Co-promotor:
Dr. W.A. Kosters

Promotiecommissie:
Prof.dr. N. Jouandeau (Université Paris 8)
Dr. K. Burke (Florida Southern College)
Prof.dr. F. Thuijsman (Maastricht University)
Prof.dr. H.J. Hupkes
Prof.dr. M. Fiocco

Contents

Chapter 1

Introduction

In this thesis, we consider combinatorial games and all sorts of variants of these games. We start off with a preliminary exploration of these topics.

1.1 Combinatorial games

A combinatorial game is a game with perfect information and no random elements in which two players take turns to compete for the win. Every turn, the current player has a set of moves to choose from, arriving in the next game state. Usually, once a player no longer has any possible moves to do, that player loses.

Example 1.1.1. The game *Hackenbush*, or *Red-Blue-Hackenbush*, is played on a graph, one node being designated as the "ground", and each of the edges colored either blue or red. On their turn, the one player may cut any one of the blue edges, while the other player may cut a single red edge. If, after a move, a part of the graph is disconnected from the ground node, it disappears.

Traditionally, the ground is not drawn as a single node, but as a line, making no technical difference. An example position is shown in Figure 1.1. From here, the blue player could, for example, cut the right arm, removing the whole balloon in the process, as well. $\triangleleft$

Formally, a game is defined by the states to which both players can move in a single turn, also called the *options* of the players.

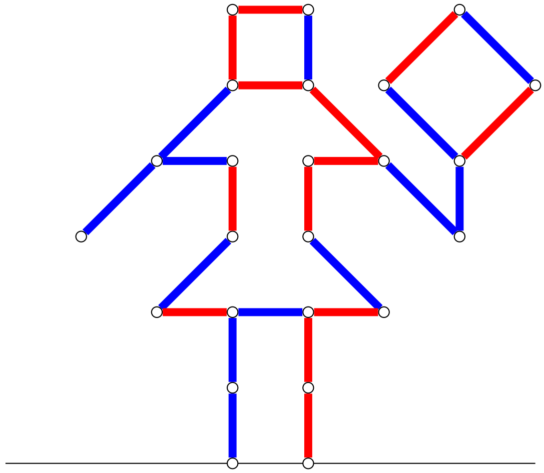


Figure 1.1: A position of Red-Blue Hackenbush.

Definition 1.1.2. We define a *game* G by its set of Left options, $\mathcal{G}^L$, and its set of Right options, $\mathcal{G}^R$, both consisting of games. Notation: $G = \{\mathcal{G}^L \mid \mathcal{G}^R\}$.

Hence, from a game G, Left may play to any $G^L \in \mathcal{G}^L$, and Right may play to any $G^R \in \mathcal{G}^R$. The "first" game in some sense, from which the recursive definition is started, is $\{\emptyset \mid \emptyset\}$, which is also denoted by $\{\ \mid\ \}$. We assume both $\mathcal{G}^L$ and $\mathcal{G}^R$ to be finite, and we denote by $\mathbb{G}$ the set of all such games.

Standard works on combinatorial game theory include Winning Ways for Your Mathematical Plays [1] and Lessons in Play [2], in which the theory is introduced in a lighthearted fashion, as well as the comprehensive Combinatorial Game Theory [3]. On Numbers and Games [4] takes a somewhat more formal approach, focusing more on the algebraic structure of the games.

The distinction in colors between the players, such as used in Hackenbush, is commonplace. The one player, using the blue or black pieces, is often called Left, being addressed as female, while the other, playing the red or white pieces, is called Right and uses male pronouns. By the deterministic nature of the games, under perfect play of both players, a game can have any of four outcomes: the Left player wins, the Right player wins, the starting player wins, or the second player to move wins. By this distinction, one may divide combinatorial games in four *outcome classes*, summarized in Table 1.1.

		Right moves first	
		Left wins	Right wins
Left moves first	Left wins	$\mathcal{L}$	$\mathcal{N}$
	Right wins	$\mathcal{P}$	$\mathcal{R}$

Table 1.1: The possible outcome classes of a game in $\mathbb{G}$.

A key insight in combinatorial game theory is that games may often be deconstructed into smaller, mostly independent parts, which are more easy to analyze. For positional games such as Go, Chess or Tic Tac Toe, which do not traditionally end upon a player no longer having a legal move, this can prove very useful, modelling the games as a combinatorial game and using the available theory. In doing so, steps have been made in the endgame analysis of these games [5,6].

More formally, we look at the deconstruction of games from a bottom-up point of view, facilitating the construction of larger games from smaller components. For two games $G, H \in \mathbb{G}$, we can define a new game by putting the games next to each other, a legal move in the new game now being a move in either component.

Definition 1.1.3. Let $G, H \in \mathbb{G}$. Then the (*disjunctive*) *sum* of G and H is

$$G + H = \{\mathcal{G}^L + H, G + \mathcal{H}^L \mid \mathcal{G}^R + H, G + \mathcal{H}^R\},$$

where we write $\mathcal{G}^L + H = \{G^L + H : G^L \in \mathcal{G}^L\}$.

Using this concept of sums of games and the outcome classes in Table 1.1, we may define a notion of equality of games. Two games are called equal if they "behave" the same in any context, that is, adding the games to any other context of games cannot produce a different outcome class. Note that this combinatorial definition of equality defines an equivalence relation on $\mathbb{G}$.

Definition 1.1.4. Let $G, H \in \mathbb{G}$. We define $G = H$ if $o(G + X) = o(H + X)$ for all $X \in \mathbb{G}$.

Even if games are unequal, we may often compare them, showing that one or the other is more beneficial to either of the players. These comparisons hinge on the order of the outcome classes depicted in Figure 1.2, associating greater games with being more favorable for Left.

Definition 1.1.5. Let $G, H \in \mathbb{G}$. We define $G \geq H$ if $o(G + X) \geq o(H + X)$ for all $X \in \mathbb{G}$.

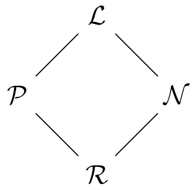


Figure 1.2: The partial order on the outcome classes.

The theory built from the definitions of the outcome classes, sums of games, and their equality and order provides a beautiful framework and myriad of tools for analyzing combinatorial games. We further explore this theory in Chapter 2, as well as introducing other concepts from, e.g., algorithmic game theory. Then, after looking at some actual combinatorial and positional games in Chapters 3 and 4, we ask the question of what remains of the theoretical framework if we drop one of the core assumptions for combinatorial games.

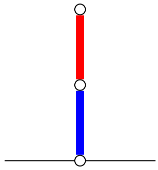
1.2 Variants

By definition, combinatorial games are games for two players taking turns, with perfect information and no chance. Naturally, each of these assumptions can be dropped, fundamentally altering the type of game encountered. Work has, for example, been done on deterministic perfect information games for more than two players [7,8], or on perfect information games for two players involving randomness [9].

In this thesis, we will focus on the removal of the perfect information component. On the one hand, this can be done by not revealing all details of a move to the opponent, such as in the game of Kriegspiel [10,11]. By doing so, a player at any time may be unsure in which state the game is precisely, and more information may be obtained through attempting to move. The game becomes a non-cooperative game in extensive form, detailed in [12]. We look more closely at this type of game in Chapter 5.

Another possibility of introducing imperfect information is to allow the players to move simultaneously, instead of on a turn-by-turn basis [13]. Under this regime, the outcome classes $\mathcal{N}$ and $\mathcal{P}$ no longer exist, as the concept of order between the players is lost. Instead, games may now end in a draw.

Example 1.2.1. Consider the Red-Blue Hackenbush game

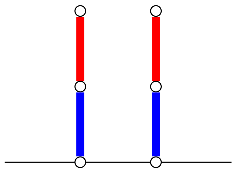
$G =$ 

Playing simultaneously, both players remove their only edge at once, resulting in the empty game. Subsequently, neither player having any options left, we declare the game a draw. ◁

One option of analyzing these games is to extend the definitions of equality and order for combinatorial games to this new class of synchronized games. This is explored for the synchronized game of Cherries in Chapter 6.

A second option is to view the synchronized games as nested zero-sum games, ultimately resulting in some payoff for either player. For Hackenbush, for example, one might argue that a game consisting of only n blue edges should be assigned value n, representing n "free" moves for Left. A game consisting of both blue and red edges is then valued equal to the value of its Nash equilibria.

Example 1.2.2. Consider the Red-Blue Hackenbush game

$G =$ 

The optimal (Nash) strategy for both players is to remove either of their edges with equal probability. Half of the times, this leads to a single blue edge remaining, having value 1; the other half, it leads to the game in Example 1.2.1, having value 0. Hence, the value of the Nash equilibrium of the game, and therewith of the game itself, is $\frac{1}{2}$. ◁

We see that, under this regime, the optimal strategy for both players need no longer be deterministic. We further discuss the analysis of synchronized games using their Nash values in Chapters 7 and 8.

1.3 Structure of the thesis

Though this thesis is designed as a single piece of work, each chapter can be read independently. To this end, some chapters contain some overlap, especially in the introductory paragraphs. We outline the contents of the chapters.

Chapter 2: Background. We start off by discussing the fundamental concepts that are developed further in the rest of the thesis. First, we cover existing material on combinatorial and economic game theory. Next, we introduce the concept of synchronized games. We state and prove some fundamental properties, and show how to construct synchronized versions of existing combinatorial games. Finally, we discuss two methods of evaluating these synchronized games.

Chapter 3: Hackenbush variants. In this chapter, we consider two combinatorial variants of Red-Blue Hackenbush. The first is Childish Hackenbush, introduced in [1]. In this variant of the game, players are not allowed to remove edges that would disconnect a part of the graph from the ground. The presented analysis is based on joint work with Nienke Burgers [14]. In the second variant, Uncolored Hackenbush, the game is started with a graph consisting of uncolored edges. In the first phase of the game, the players take turns coloring the edges blue and red. In the second phase, Hackenbush is played out as usual.

Chapter 4: Order and Chaos. Based on joint work with Sipke Castelein and Daan van Gent [15], in this chapter, we consider the positional game Order versus Chaos, introduced in [16]. In this variant of Tic Tac Toe, both players may place crosses or circles on a board. One player, called Order, attempts to construct a horizontal, vertical or diagonal line of identical symbols, while the other player, named Chaos, tries to prevent this while filling the board. We provide a theoretical analysis of the game on varying board sizes, showing that either player must win under perfect play, utilizing a SAT solver for part of the proofs. Moreover, we use Monte Carlo Tree Search to produce results for the games not covered by the theoretical analysis.

Chapter 5: Nim variants. In this chapter, based on [17], we analyse three turn-based imperfect information variants of the combinatorial game of Nim.

The game is played on heaps of coins. Every turn, the active player may take any number of coins from a single heap. The three variants differ in the amount of information that is provided to the opponent after making a move. One of the variants is inspired by Kriegspiel, a non-perfect information variant of chess. We model the variants as games in extensive form and compute Nash equilibria for some examples.

Chapter 6: Synchronized Cherries. In this chapter, based on joint work with Thomas de Mol [18], we consider the synchronized version of the game of Cherries. The game is played on strips of white and black tokens. Every turn, Left takes a black token from the end of a strip, and Right takes a white token from the end of a strip. We extend the definition of combinatorial equality to this synchronized game, and show that under this definition, every Cherries position is equal to a sum of positions of the game in which both players may only take cherries from one side of every strip. Moreover, we give an algorithm which computes this decomposition.

Chapter 7: Synchronized Hackenbush. We consider the synchronized version of the game of Red-Blue Hackenbush. We model these games as nested matrix games and compute their Nash equilibria. We show that, for some simple games, the Nash value of an increasing amount of copies of the game tends to the combinatorial value. Finally, we shortly consider the variant of the game with green edges, that may be cut by either player.

Chapter 8: Synchronized Push. Finally, we consider the synchronized version of Push, based on joint work with Ronald Takken [19]. Again, we model the games as nested matrix games and compute their Nash equlibria, concluding that the Nash value of copies of a position tends to their number value as a combinatorial game. We conclude with a short analysis of some games of synchronized Shove.

Chapter 2

Background

In this chapter, we will cover the background needed for the remainder of the thesis. We start out with a brief overview of selected topics in the area of combinatorial game theory, based on material from [1–4]. We repeat some material from the first chapter, and expand upon it — all results in this section are taken from the literature. Next, we touch upon some topics in algorithmic (non-cooperative) game theory, based on [12, 20]. Finally, we introduce the concept of synchronized games, inspired by [13, 21–23], and provide some new results.

2.1 Combinatorial game theory

Intuitively, a combinatorial game is a two-player game with perfect information and no chance, in which the players alternate taking turns. Well-known examples include Domineering, Nim and Hackenbush. An extensive theory has been developed to analyse these games. We will introduce some of the concepts from this theory.

2.1.1 Fundamental definitions

Two players, named Left (or bLue or bLack; female) and Right (or Red or white; male) compete, taking turns to make a move. Formally, such a game is defined recursively, as follows.

Definition 2.1.1. We define a *game* G by its set of Left options, $\mathcal{G}^L$, and its set of Right options, $\mathcal{G}^R$, both consisting of games. Notation: $G = \{\mathcal{G}^L \mid \mathcal{G}^R\}$.

Hence, from a game G, Left may play to any $G^L \in \mathcal{G}^L$, and Right may play to any $G^R \in \mathcal{G}^R$. The smallest game is $\{\emptyset \mid \emptyset\}$, which is also denoted by $\{\ \mid\ \}$, or 0, zero. Unless stated otherwise, we assume both $\mathcal{G}^L$ and $\mathcal{G}^R$ to be finite. We denote by $\mathbb{G}$ the set of all games.

One can also view a game as a *tree* rooted in G, each node H having as left children all elements in $\mathcal{H}^L$ and as right children all elements in $\mathcal{H}^R$. Any node in the tree, including G, is called a *position* of G. If, for two games $G, H \in \mathbb{G}$, their game trees are isomorphic, we call G and H *isomorphic* and write $G \cong H$. Two isomorphic games are the same for all intents and purposes.

The recursive definition of a game gives rise to the following definition.

Definition 2.1.2. Let G be a game. The *birthday* of G is defined recursively as

$$b(G) = \max_{H \in \mathcal{G}^L \cup \mathcal{G}^R} \{b(H)\} + 1,$$

with $b(0) = 0$.

Under the *normal play* convention, we say a player loses if they have no more moves available during that turn, that is, if P needs to move next while $\mathcal{G}^P = \emptyset$. Under *misère play*, a player wins if they cannot move. In this thesis, we will consider normal play, unless mentioned otherwise. Under this convention, the fundamental theorem of combinatorial game theory reads as follows.

Theorem 2.1.3. [2, *Theorem 2.1*] *In a game played between Left and Right, with Left moving first, either Left can force a win moving first, or Right can force a win moving second, but not both.*

According to this theorem, the games in $\mathbb{G}$ can be divided into four *outcome classes*, summarized in Table 2.1.

For a game $G \in \mathbb{G}$, we write $o(G) \in \{\mathcal{L}, \mathcal{R}, \mathcal{N}, \mathcal{P}\}$ for its outcome class. The classes are partially ordered as shown in Figure 2.1.

For two games $G, H \in \mathbb{G}$, we can define a new game by putting the games next to each other, a legal move being a move in either game. Formally, this is put as follows.

		Right moves first	
		Left wins	Right wins
Left moves first	Left wins	$\mathcal{L}$	$\mathcal{N}$
	Right wins	$\mathcal{P}$	$\mathcal{R}$

Table 2.1: The possible outcome classes of a game in $\mathbb{G}$.

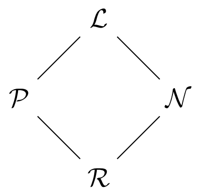


Figure 2.1: The partial order on the outcome classes.

Definition 2.1.4. Let $G, H \in \mathbb{G}$. Then the (*disjunctive*) *sum* of G and H is

$$G + H = \{\mathcal{G}^L + H, G + \mathcal{H}^L \mid \mathcal{G}^R + H, G + \mathcal{H}^R\},$$

where we write $\mathcal{G}^L + H = \{G^L + H : G^L \in \mathcal{G}^L\}$.

In general, like in the definition above, for a set X and property P, we will often abuse notation and write $P(X)$ to denote $\{P(x) : x \in X\}$.

Definition 2.1.5. Let $G \in \mathbb{G}$. The *negative* of G, denoted by $-G$, is defined recursively as

$$-G = \{-\mathcal{G}^R \mid -\mathcal{G}^L\}.$$

Hence, taking the negative of a game amounts to reversing the roles of the players.

We are now in position to define *equality* of games.

Definition 2.1.6. Let $G, H \in \mathbb{G}$. We define $G = H$ if $o(G + X) = o(H + X)$ for all $X \in \mathbb{G}$.

Two games are called equal if they "behave" the same in any context, that is, adding the games to any other context of games cannot produce a different outcome class. Note that, in fact, this combinatorial definition of equality actually defines an equivalence relation on $\mathbb{G}$. True "equality", that is, the same behavior in any circumstance, is only achieved if two games are isomorphic.

However, for most practical purposes, the concept of game equality is very useful. It makes it easy to see, for example, when a game is an element of $\mathcal{P}$.

Proposition 2.1.7. [2, *Theorem 4.12*] *Let* $G \in \mathbb{G}$*. Then* $G = 0$ *if and only if* $G \in \mathcal{P}$*.*

With the definitions given so far, we are able to prove the following.

Theorem 2.1.8. [2, *Theorem 4.26*] $(\mathbb{G}, +)$ *is an Abelian group.*

The partial order on the equivalence classes implies the following partial order on $\mathbb{G}$, similar to the definition of equality.

Definition 2.1.9. Let $G, H \in \mathbb{G}$. We define $G \geq H$ if $o(G + X) \geq o(H + X)$ for all $X \in \mathbb{G}$.

Theorem 2.1.10. [2, *Theorem 4.25*] *The relation* $\geq$ *is a partial order on* $\mathbb{G}$*.*

Every game has a unique "simplest form" in some sense, which we call the *canonical form* of a game.

Theorem 2.1.11. [2, *Theorem 4.33*] *Let*

$$G = \{G_1^L, G_2^L, G_3^L, \ldots \mid G_1^R, G_2^R, \ldots\}$$

and suppose $G_1^L \geq G_2^L$*. Then* $G = G'$ *with*

$$G' = \{G_1^L, G_3^L, \ldots \mid G_1^R, G_2^R, \ldots\}.$$

We say the option G_2^L *is* dominated *by the option* G_1^L*. Similarly, if* $G_1^R \leq G_2^R$*, the game* G *is equal to the game with the option* G_2^R *removed.*

Theorem 2.1.12. [2, *Theorem 4.34*] *Let*

$$G = \{G_1^L, G_2^L, G_3^L, \ldots \mid G_1^R, G_2^R, \ldots\}$$

and suppose that $(G_1^L)^R \leq G$ *for some Right option* $(G_1^L)^R$ *of* G_1^L*. Then* $G = G'$ *with*

$$G' = \{(\mathcal{G}_1^L)^{RL}, G_2^L, G_3^L, \ldots \mid G_1^R, G_2^R, \ldots\}.$$

We say the option G_1^L *is* reversible *through* $(G_1^L)^R$*, and call* $(\mathcal{G}_1^L)^{RL}$ *the* replacement set*. A similar result holds for Right.*

Definition 2.1.13. Let $G \in \mathbb{G}$. If G has no dominated nor reversible options, we say G is in *canonical form*.

Theorem 2.1.14. [2, *Theorem 4.36*] *Let* $G, H \in \mathbb{G}$ *be in canonical form. If* $G = H$*, then* $G \cong H$.

We can thus speak of *the* canonical form of a game G, which we will denote by $\mathrm{Can}(G)$.

Finally, note that, so far, we have only discussed formal games. In practice, we often like to discuss a set of games that are all played according to some specific rules, e.g., Hackenbush, Nim or Domineering. We call all games which belong to such a class of rules a *ruleset*.

2.1.2 Numbers

The *numbers* are a special class of games.

Definition 2.1.15. Let $n \in \mathbb{N}$. We define the *integers* recursively by $0 = \{\ |\ \}$,

$$n = \{n - 1 \mid\ \} \quad \text{and} \quad -n = \{\ \mid -(n-1)\}.$$

Moreover, more generally, we define the *number*

$$\frac{1}{2^n} = \left\{ \frac{1}{2^{n-1}} \,\middle|\, \right\}.$$

Naturally, we write $2^0 = 1$. The number games $\{2^{-n} \mid n \in \mathbb{N}\}$ generate a subgroup of games $\mathbb{D}$ isomorphic to the dyadic rationals. Moreover, their canonical form is straightforward.

Theorem 2.1.16. [3, *Theorem II.3.6*] *For any* $m, n \in \mathbb{N}$,

$$\frac{m}{2^n} = \left\{ \frac{m-1}{2^n} \,\middle|\, \frac{m+1}{2^n} \right\}$$

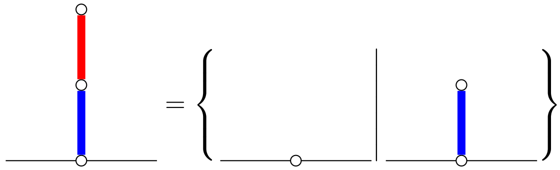
in canonical form.

Moreover, determining the value of games of which all the options are numbers is simple, as long as the left options are smaller than the right options.

Definition 2.1.17. Let $x < y$ be numbers. The *simplest number* between x and y is the unique number in the interval (x, y) with the smallest birthday.

Theorem 2.1.18 (Simplest number theorem)**.** *Let $G = \{\mathcal{G}^L \mid \mathcal{G}^R\}$ be such that all options are numbers, and $G^L < G^R$ for every $G^L \in \mathcal{G}^L$ and $G^R \in \mathcal{G}^R$. Then G equals the simplest number between $\max\{G^L\}$ and $\min\{G^R\}$.*

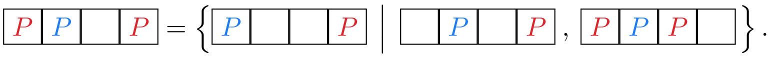
Example 2.1.19. Recall the game of *Red-Blue-Hackenbush* described in Example 1.1.1. By the simplest number theorem, we find, for example,



$$= \{0 \mid 1\} = \frac{1}{2}.$$

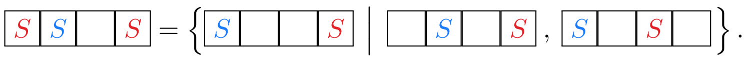
◁

Example 2.1.20. The game *Push* is played on a strip of squares. On her turn, Left may move a blue piece one square to the left, pushing any pieces that are in the way one space to the left as well, falling off the strip if they are moved off it. Right, on his turn, moves a red piece, also to the left. An example position is



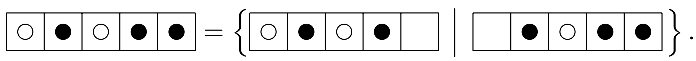
◁

Example 2.1.21. The game *Shove* is similar to Push, again being played on a strip of squares. The difference is that, in Shove, empty spaces are also pushed. Hence, the example position as shown previously would play out as follows:



◁

Example 2.1.22. The game of *Cherries* is also played on a strip of squares. On Left's turn, she may remove a black cherry that is adjacent to an empty square, or to the end of the strip. Right removes a white cherry under the same restrictions. An example game is



◁

For Red-Blue Hackenbush, Push, Shove, and Cherries, all positions are numbers [1,2,18,19]. We will consider variants of Hackenbush in Chapters 3 and 7, of Push and Shove in Chapter 8, and of Cherries in Chapter 6.

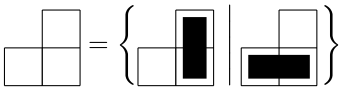
2.1.3 Infinitesimals

Not all games are numbers. An important class of games is that of infinitesimal games.

Definition 2.1.23. A game G is *infinitesimal* if $-x < G < x$ for all numbers $x > 0$.

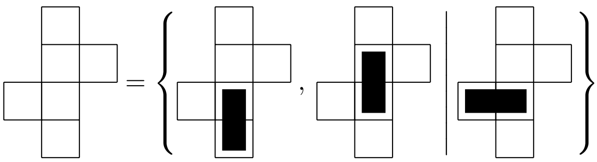
Example 2.1.24. Trivially, the game 0 itself is infinitesimal, being the only infinitesimal number. Somewhat less trivial is the game $* = \{0 \mid 0\}$, which is the smallest example of a next-player win. The game $\uparrow = \{0 \mid *\}$, pronounced *up*, is an example of an infinitesimal win for Left. ◁

Example 2.1.25. The game of *Domineering* is played on a board of squares. On her turn, Left places a domino covering two vertically adjacent squares; Right places a domino covering two horizontally adjacent squares. A player unable to place a domino on as-of-yet uncovered squares, loses. Example positions of Domineering are



$$= \{0 \mid 0\} = *$$

and, by symmetry and reversibility,



$$= \{*, 0 \mid *\} = \{0 \mid *\} = \uparrow.$$

◁

Even smaller than the games mentioned above are the tiny games.

Definition 2.1.26. Let G be any game. The game *tiny-G* is defined by $+_G = \{0 \parallel 0 \mid -G\}$. The game *miny-G* is its negative $-_G = \{G \mid 0 \parallel 0\}$.

Example 2.1.27. Taking $G = 0$ yields

$$+_0 = \{0 \parallel 0 \mid 0\} = \{0 \mid *\} = \uparrow .$$

◁

Definition 2.1.28. Let $G, H > 0$ be games. We say G is *infinitesimal with respect to* H, notation $G \ll H$, if $G < n \cdot H$ for any $n \in \mathbb{N}_{>0}$.

Theorem 2.1.29. [2, *Exercise 5.60*] *Let* $G > H \geq 0$ *be numbers. Then* $+_G \ll +_H$.

Hence, the tiny games provide us with an infinite sequence of ever-smaller games, each infinitely smaller than the previous one. Another such sequence is given by the uptimals.

Definition 2.1.30. We define $\uparrow^1 = \uparrow$, and, recursively, for $n \in \mathbb{N}_{>1}$,

$$\uparrow^n = \{0 \mid * - \uparrow^1 - \ldots - \uparrow^{n-1}\}.$$

Theorem 2.1.31. [2, *Theorem 9.12*] *For all* $n \in \mathbb{N}_{>1}$ *it holds that* $\uparrow^n \ll \uparrow^{n-1}$.

An infinitesimal game being denoted by

$$.n_1 n_2 n_3 \ldots = n_1 \cdot \uparrow + n_2 \cdot \uparrow^2 + n_3 \cdot \uparrow^3 + \ldots$$

is said to be in *uptimal notation*. Not every infinitesimal game can be written in uptimal notation.

2.1.4 Impartial games

In some games, there is no distinction between Left and Right.

Definition 2.1.32. A game G is *impartial* if, for every position H of G, we have $\mathcal{H}^L = \mathcal{H}^R$.

Example 2.1.33. The games 0 and $*$ are impartial. ◁

Example 2.1.34. The game of *Nim* is played on heaps of coins. On a player's turn, they may remove any amount of coins from any one single heap. A player unable to move loses.

Writing (i, j, k) for a position consisting of three heaps containing i, j and k coins, respectively, a game of Nim might unfold like this:

$$(3,7,2) \to (3,4,2) \to (1,4,2) \to (1,4,1) \to (1,0,1) \to (1,0,0) \to (0,0,0).$$

◁

Naturally, Nim is impartial by definition. We will consider variants of Nim in Chapter 5.

Theorem 2.1.35. [2, *Theorem 2.13*] *The outcome class of any impartial game is either $\mathcal{P}$ or $\mathcal{N}$.*

Theorem 2.1.36. [2, *Corollary 7.8*] *Every impartial game is infinitesimal.*

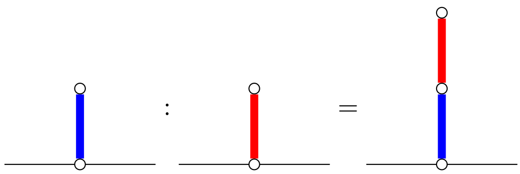
2.1.5 Ordinal sums

A different binary operation from the regular sum is the ordinal sum, which more or less amounts to 'putting two games on top of each other'.

Definition 2.1.37. For two games G and H, their *ordinal sum* is defined by

$$G : H = \{\mathcal{G}^L, G : \mathcal{H}^L \mid \mathcal{G}^R, G : \mathcal{H}^R\}.$$

Example 2.1.38. The concept or ordinal sum naturally appears in Hackenbush. We find, for example, that



In symbols, this reads $1 : -1 = \frac{1}{2}$. ◁

Example 2.1.39. In Red-Blue-Green Hackenbush, edges can also be colored green, which denotes an edge that may be cut by either player. If a position contains only green edges, the position is an impartial game. ◁

The following rules of arithmetic come in handy, taken from Chapter 10 in [2].

Theorem 2.1.40.

(i) $-(G : H) = (-G) : (-H)$
(ii) $G : 0 = 0 : G = G$
(iii) *If* $G, H \geq 0$ *are integers, then* $G : H = G + H$.

Theorem 2.1.41 (Colon principle)**.** *If* $H = K$*, then* $G : H = G : K$.

We will encounter the ordinal sum in detail in Section 3.2.

2.1.6 Switches

Not all next-player wins are infinitesimals.

Definition 2.1.42. Let $x > 0$ be a number. The game $\pm x = \{x \mid -x\}$ is called a *switch*.

For switches, the following theorem often comes in handy.

Theorem 2.1.43 (Number translation)**.** *Let G be not a number, and let x be a number. Then*

$$G + x = \{\mathcal{G}^L + x \mid \mathcal{G}^R + x\}.$$

Again, we will encounter switches in Section 3.2.

2.2 Algorithmic game theory

In combinatorial games, the players make moves sequentially, and both players always have perfect information. When the players move simultaneously, or imperfect information is introduced in the game, we move into the territory of (*economic*) *algorithmic game theory*. In games of this category, optimal strategies for the players may no longer be deterministic. In fact, we need to be more careful in specifying what "optimal" means.

We start by introducing the necessary concepts concerning zero-sum games and the corresponding optimal strategies, called Nash equilibria. We proceed by giving an algorithmic approach to finding these optimal strategies for any given game using linear programming, based on [12]. We conclude by looking at some imperfect information variants of existing combinatorial games.

2.2.1 Games in extensive form

To introduce the more general framework of zero-sum games in extensive form, we need some notation concerning trees.

Notation 2.2.1. Let $T = (V, A)$ be a directed tree rooted at $r \in V$. For a vertex $v \in V$, we denote its children by $N^+(v) \subseteq V$. The edges between v and $N^+(v)$ are denoted by $E^+(v) \subseteq A$. We let $V_0 \subseteq V$ be the set of leaves of T, that is, $V_0 = \{v \in V \mid N^+(v) = \emptyset\}$.

Now, again, two players Left and Right compete. We proceed with the definition of a game in extensive form and its corresponding Kuhn tree, as proposed in [20].

Definition 2.2.2. A finite *two-person zero-sum game in extensive form* is defined by the following:

(i) A finite directed tree $T = (V, A)$, called the Kuhn tree, rooted in the initial state of the game $r \in V$;

(ii) A payoff function $f\colon V_0 \to \mathbb{R}$ assigning some real value to every leaf of T;

(iii) A set $V_p \subseteq V \setminus V_0$ of chance vertices, with for each $v \in V_p$ a probability distribution p_v over $E^+(v)$;

(iv) A partition of $V \setminus (V_0 \cup V_p)$ into information sets $\mathcal{S}^L = \{S_1^L, \ldots, S_{K_L}^L\}$ and $\mathcal{S}^R = \{S_1^R, \ldots, S_{K_R}^R\}$, such that in all $v \in S_i^L$, it is Left's turn to move, and in all $v \in S_j^R$, it is Right's turn;

(v) For each $S_i^P \in \mathcal{S}^P$, a set of action(label)s $A_i^P = A(S_i^P)$, and for each $v \in S_i^P$, a bijection $\alpha_v\colon N^+(v) \to A_i^P$.

We call the vertices in $V \setminus (V_\ell \cup V_p)$ the *states* of the game. These states are grouped into *information sets*, or info sets for short. To a player, the states in an info set S_i^P are indistinguishable, that is, if P knows that the game is now in some state in S_i^P, it is unknown in which state the game is exactly. Therefore, the *moves* or *actions* in every state v in an info set S_i^P, represented by the edges $E^+(v)$ leading to the children of the vertex v, must be identical across all the vertices in the info set. This is guaranteed by the fifth point in the above definition.

When the game arrives in a chance node $v \in V_p$, the next vertex to which the game moves is determined by the probability distribution p_v. Unless stated otherwise, we will assume that $V_p = \emptyset$. Finally, when the game arrives in a leaf $v \in V_0$ of the tree, Left obtains a payoff of $f(v)$, if $f(v) > 0$. If $f(v) < 0$, Right receives a payoff of $|f(v)|$.

We continue by defining strategies.

Definition 2.2.3. Let G be a game in extensive form with Kuhn tree T. A *pure strategy* $\pi_P \in \prod_{i=1}^{K_P} A_i^P$ specifies for every information set of player P a move to make. A *mixed strategy* μ_P is a probability distribution over the set of pure strategies of P.

Definition 2.2.4. Let G be a game in extensive form with Kuhn tree T. A *behavior* *strategy* β_P for player P specifies for every A_i^P a probability distribution over its elements.

Note the subtle difference between mixed and behavior strategies. When playing using a mixed strategy, a player makes a single "dice roll" at the start of the game, which then specifies what to do in every possible information set for the whole of the game at once. When employing a behavior strategy, the player may make a "dice roll" every time a new vertex is encountered. As the following examples show, there may be mixed strategies which cannot be described as behavior strategies and vice versa. Here, "described as" means the following.

Definition 2.2.5. Two strategies of a player P are called *realization equivalent* if they reach any node $v \in V$ with the same probability, given some fixed strategy of the other player.

Example 2.2.6. Consider the game in Figure 2.2, called the *absent-minded driver problem* [24]. In this game, only Left has decisions to make. There is only one info set, S say, from which there is a choice between two moves labelled A and B.

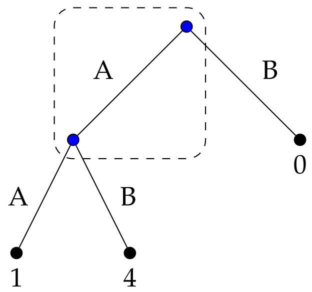


Figure 2.2: The absent-minded driver problem.

The two pure strategies available to Left are to choose either action A or action B in this one info set. A mixed strategy for Left thus consists of a probability distribution over the actions A and B. At the start of the game, it is decided whether Left will always play A or B according to this distribution. In practice, this means that Left will always end up with a payoff of 1 or 0.

A behavior strategy also consists of a probability distribution over the actions A and B, but now the player may draw from this distribution every time he enters a state in S. Hence, if we give picking A and B equal probabilities, for example, we end up with an average payoff of $\frac{1}{2} \cdot 0 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 4 = \frac{5}{4} > 1$. Note that this strategy is not realization equivalent to any mixed strategy, as this would not allow us to pick a different action the two times that Left finds herself in S. ◁

Example 2.2.7. Now, consider the game with Kuhn tree depicted in Figure 2.3. In this game, there are two info sets for Left, say S_1 with corresponding labels $L_1 = \{\text{A}, \text{B}\}$ and S_2 with labels $L_2 = \{\text{C}, \text{D}\}$. In this game, the pure strategies are the pairs (A,C), (A,D), (B,C) and (B,D). Mixed strategies are any probability distributions over these pairs, e.g., picking (A,D) or (B,C) both with probability $\frac{1}{2}$.

Note, however, that this mixed strategy in particular is not realization equivalent to a behavior strategy. Indeed, a behavior strategy can only specify a probability distribution over the elements of L_1 and a distribution over L_2; it cannot incorporate the dependence of the second action on the first.

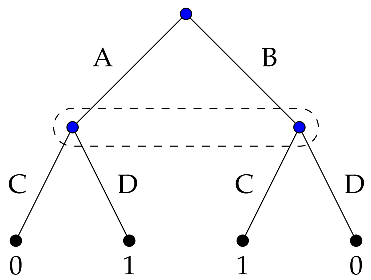


Figure 2.3: Dependency of moves.

◁

If the same information set cannot be entered twice during one iteration of the game, every behavior strategy is realization equivalent to a mixed strategy. Indeed, suppose $p(A_1^P), \ldots, p(A_{K_P}^P)$ are probability distributions over the elements of the action sets A_i^P, describing a behavior strategy β_P for player P. For an action $a_j^i \in A_i^P$, we write $\beta_P(a_j^i)$ to be the probability of playing action a_j^i in info set S_i^P under β_P. Let $\pi_P = (\pi_P^1, \ldots, \pi_P^{K_P}) \in \prod_{i=1}^{K_P} A_i^P$ be a pure strategy. Then

$$\beta_P(\pi_P) := \prod_{i=1}^{K_P} \beta_P(\pi_P^i)$$

is the probability of P playing by the pure strategy π_P under β_P. We call this the *realization probability* of π_P under β_P. Now, note that we can define a mixed strategy μ_P of P by setting $\mu_P(\pi_P) := \beta_P(\pi_P)$. In this way, μ_P is equivalent to β_P. In the sequel, we want this to hold, so we make the following assumption.

Assumption 2.2.8. Let G be a game in extensive form with Kuhn tree T. On any path from the root r of T to a leaf in V_0, every vertex of the path is contained in a distinct information set in $\mathcal{S}^L \cup \mathcal{S}^R$.

For any mixed strategy to be equivalent to some behavior strategy, we need another, stronger, property, which we develop in the next section.

2.2.2 Sequence form

In this section, we will develop the *sequence form* of a game, which is a somewhat efficient representation of a game in extensive form having size linear in the size of the game tree. We follow the treatment in [12].

Consider a game with Kuhn tree $T = (V, E)$ rooted at r and let $v \in V$ be a node of the tree. We will write $\sigma(v)$ for the *sequence* of all actions A_i^P encountered on the unique path from r to v, that is, $\sigma(v)$ is the sequence of moves made by the players in order to end up in the node v. If we write $\sigma_L(v)$, we consider only the sequence of actions A_i^L made by Left; the definition of $\sigma_R(v)$ is similar. We write Σ_L for all sequences of consecutive moves made by Left; analogous for Σ_R. Equipped with the notion of sequences, we can talk about perfect recall.

Definition 2.2.9. Player P is said to have *perfect recall* if for every information set $S_i^P \in \mathcal{S}^P$ and any two $v, w \in S_i^P$, we have $\sigma_P(v) = \sigma_P(w)$. In this case, we denote the unique sequence leading to a node in S_i^P by $\sigma_i^P = \sigma(S_i^P)$.

In words, a player with perfect recall will always remember the moves they made leading to the current game state. In practice, this can be enforced by storing the sequence of moves made so far in the description of the information set.

Note that, if a player has perfect recall, they cannot enter the same info set twice during the same playthrough. Therefore, both players having perfect recall implies Assumption 2.2.8. Furthermore, this assumption is enough to enable us to describe any mixed strategy by a behavior strategy. We make this precise, thus employing the following assumption.

Assumption 2.2.10. Both players have perfect recall.

Let $\pi_P \in \prod_{i=1}^{K_P} A_i^P$ be a pure strategy of player P, and let μ_P be a strategy that picks π_P with probability $\mu_P(\pi_P)$. Let $\sigma \in \Sigma_P$ be some sequence for the player P. We write $\pi_P[\sigma]$ for the realization probability of σ under π_P, being 1 precisely if π_P prescribes all moves in σ and 0 otherwise. Similarly, we define

$$\mu_P[\sigma] = \sum_{\pi_P} \mu_P(\pi_P)\pi_P[\sigma]$$

to be the realisation probability of σ under μ_P. We can in fact consider μ_P to be a map assigning to every sequence in Σ_P its realization probability.

Definition 2.2.11. Let μ_P be a mixed strategy for player P. The map $x\colon \Sigma_P \to [0,1]$ defined by $\sigma \mapsto \mu_P[\sigma]$ is called the *realization plan* of μ_P.

Note that any non-empty sequence $\sigma \in \Sigma_P$ can be seen as the unique sequence leading to the information set in which the last move was made, followed by this last move. Hence, for any $\sigma \in \Sigma_P$, we can write either $\sigma = \emptyset$ or $\sigma = \sigma_i^P a$, where $a \in A_i^P$ is the last move in the sequence.

Lemma 2.2.12. *Let x be a realization plan of player P. Then $x(\emptyset) = 1$ and*

$$\sum_{a \in A_i^P} x(\sigma_i^P a) = x(\sigma_i^P)$$

for all $S_i^P \in \mathcal{S}^P$. Conversely, any $x\colon \Sigma_P \to \mathbb{R}$ having these properties is a realization plan of a behavior strategy of player P.

Lemma 2.2.13. *Let μ_P and μ'_P be mixed strategies of player P. Then μ_P and μ'_P are realization equivalent if and only if they have the same realization plan, that is, $\mu_P[\sigma] = \mu'_P[\sigma]$ for all $\sigma \in \Sigma_P$.*

From Lemma 2.2.12 and Lemma 2.2.13 we can conclude the following.

Theorem 2.2.14 (Kuhn, [25])**.** *Under the assumption of both players having perfect recall, any mixed strategy is realization equivalent to a behavior strategy and vice versa.*

This allows us to drop the adjectives mixed and behavior and simply speak about strategies.

Definition 2.2.15. Let μ_L and μ_R be strategies for the players L and R, respectively. The *value* of the pair (μ_L, μ_R), denoted by $v(\mu_L, \mu_R)$, is the expected payoff to player L if the players use these strategies.

Definition 2.2.16. Let (μ_L, μ_R) be a pair of strategies with value v. If it holds that $v(\mu, \mu_R) \leq v$ for all strategies μ of player L, and $v(\mu_L, \mu) \geq v$ for all strategies μ of player R, we call the pair (μ_L, μ_R) a *Nash equilibrium* of the game.

Note that, by enumerating every combination of every pure strategy of both players, we may transform any game in extensive form to a non-cooperative game in strategic (matrix) form. As our definition of a Nash equilibrium for a game in extensive form then matches with the definition of such an equilibrium in a game in matrix form, the following theorem applies.

Theorem 2.2.17 (Nash [26])**.** *Every game in extensive form has at least one Nash equilibrium.*

Definition 2.2.18. Let G be a game in extensive form, and let (μ_L, μ_R) be a Nash equilibrium of G. We define the *value* of the game G by $v(G) = v(\mu_L, \mu_R)$.

By the discussion above, we could simply convert any game in extensive form to a game in strategic form and use standard methods to generate a Nash equilibrium in this converted game, such as the Lemke-Howson algorithm [27]. However, as one might expect, enumerating all possible combinations of strategies in all the different information sets leads to a game in strategic form of which the size is exponential in the size of the Kuhn tree. Hence, we need to do better.

2.2.3 Linear programming

The fact that we can characterize strategies by their realization plan is the key. By Lemma 2.2.13, a realization plan contains all the necessary information to completely determine a strategy. Therefore, all we need to find is an optimal realization plan for both players.

A realization plan for player P can be represented as a vector $x \in [0,1]^{|\Sigma_P|}$. Recall that any non-empty sequence can be represented by the unique sequence leading up to the last info set encountered, followed by the move chosen in this set. Hence, we may write

$$\Sigma_P = \{\emptyset\} \cup \{\sigma_i^P a \mid S_i^P \in \mathcal{S}^P, a \in A_i^P\},$$

from which it follows that

$$|\Sigma_P| = 1 + \sum_{S_i^P \in \mathcal{S}^P} |A(S_i^P)| = 1 + \sum_{i=1}^{K_P} |A_i^P|.$$

We can thus consider the problem of finding an optimal strategy for player P as an optimization problem on a number of variables linear in the size of the game tree. In fact, we may even formulate it as a *linear* optimization problem.

Let x be the strategy for Left we are searching for and y the strategy for Right. Lemma 2.2.12 gives us the appropriate constraints for our vectors x and y, being

$$Ex = e, \quad x \geq 0 \qquad \text{and} \qquad Fy = f, \quad y \geq 0,$$

where E has $1 + |\mathcal{S}^L|$ rows and $|\Sigma_L|$ columns and $e = (1, 0, \ldots, 0)^T \in \mathbb{R}^{|\Sigma_L|}$, so that the first row of $Ex = e$ represents the equation $x(\emptyset) = 1$ and the other rows represent the equations $\sum_{l \in L_i^P} x(\sigma_i^P l) - x(\sigma_i^P) = 0$. Similarly, F has $1 + |\mathcal{S}^R|$ rows and $|\Sigma_R|$ columns and $f = (1, 0, \ldots, 0) \in \mathbb{R}^{|\Sigma_R|}$, so that $Fy = f$ represents the equations for y.

For the optimization, define the $|\Sigma_L| \times |\Sigma_R|$-matrix A by $a_{\sigma\tau} = f(v)$ for $\sigma \in \Sigma_L$, $\tau \in \Sigma_R$, where $v \in V_0$ is the leaf node reached if Left follows the sequence σ and Right the sequence τ. If the combination of σ and τ does not lead to a leaf node, we define $a_{\sigma\tau} = 0$. Hence, if Left plays according to the realization plan x and Right plays according to y, the expected payoff for Left is $x^T Ay$. Thus, for a given realization plan y, Left tries to solve

$$\max \left\{ x^T Ay \;\middle|\; \begin{array}{r} Ex = e \\ x \geq 0 \end{array} \right\}.$$

The dual LP corresponding to this problem is given by

$$\min \left\{ e^T u \;\middle|\; \begin{array}{r} E^T u \geq Ay \\ u \lesseqgtr 0 \end{array} \right\},$$

where u is the dual variable. By strong duality, the optimal values of these two problems are equal. Therefore, if Right assumes that Left plays rationally, he wants to minimize the value of these problems by his choice of y. Now, note that in the second problem, making y a variable does not give problems for the linearity. Therefore, the LP that must be solved by Right to find an optimal strategy becomes

$$\min \left\{ e^T u \;\middle|\; \begin{array}{r} Fy = f \\ E^T u - Ay \geq 0 \\ u \lesseqgtr 0 \\ y \geq 0 \end{array} \right\}. \tag{2.1}$$

The dual to this problem which is solved by Left to find an optimal strategy is given by

$$\max \left\{ f^T v \;\middle|\; \begin{aligned} Ex &= e \\ F^T v - A^T x &\leq 0 \\ v &\lesseqgtr 0 \\ x &\geq 0 \end{aligned} \right\}. \tag{2.2}$$

By solving these problems, we obtain optimal realization plans x and y for both players. This is summarized in the following theorem, for which we give a more formal proof.

Theorem 2.2.19. *For any solutions (y^*, u^*) and (x^*, v^*) to (2.1) and (2.2), respectively, y^* and x^* form a Nash equilibrium.*

Proof. Let (y^*, u^*) and (x^*, v^*) be solutions to (2.1) and (2.2), respectively. First, note that

$$(v^*)^T f = (v^*)^T F y^* \leq x^* A y^* \leq x^* E^T u^* = e^T u^*,$$

so equality holds everywhere. Now, suppose x is some realization plan for Left. Then

$$x^T A y^* \leq x^T E^T u^* = (Ex)^T u^* = e^T u^* = x^* A y^*.$$

Moreover, for y any realization plan for Right,

$$(x^*)^T A y = (A^T x^*)^T y \geq (F^T v^*)^T y = (v^*)^T F y = (v^*)^T f = f^T v^* = x^* A y^*.$$

Hence y^* and x^* indeed form a Nash equilibrium. □

In practice, especially in Chapters 7 and 8, many of the linear programs concerned show ample symmetry. This can be exploited in efficiently solving the programs using the following result, adapted from [28].

Theorem 2.2.20. *If, in a linear programming problem, variables $x_1, \ldots, x_n$ may be permuted in any way without changing the objective function nor the solution set, we may define a new variable x and replace every occurrence of x_i by $\frac{x}{n}$ without changing the solution.*

2.3 Synchronized games

In combinatorial games, the players take turns making a move. A natural way of introducing imperfect information in these games is by requiring that both players move simultaneously. This concept was introduced in [21], and is further studied in [13].

We study the basics of synchronized games in Section 2.3.1. Though the concept is natural, in practice, it might be problematic. In some combinatorial games, for example, it might not be possible to always legally execute two sequential moves in a synchronized fashion. We discuss several ways of dealing with this in Section 2.3.4.

Moreover, even if synchronization is possible, it might not be straightforward to develop a well-defined and well-behaved notion of *value* for the resulting synchronized game. We develop two fundamentally different methods for doing so in Section 2.3.2 and Section 2.3.3. It turns out that, for different classes of combinatorial games, a different one of the two methods is better suited.

2.3.1 Definition and properties

Mirroring the definition of a combinatorial game, we give a recursive definition of a synchronized game.

Definition 2.3.1. A *synchronized game* G is a triple denoted by $\{\mathcal{G}^L \mid \mathcal{G}^S \mid \mathcal{G}^R\}$. Here, $\mathcal{G}^L = (G^L_1, \ldots, G^L_m)$ is a sequence of m synchronized games, called the *Left options* of G, $\mathcal{G}^R = (G^R_1, \ldots, G^R_n)$ is a sequence of the n *Right options* of G, and $\mathcal{G}^S = (G^S_{ij})_{ij}$ is an $m \times n$-matrix containing the *synchronized options* of G.

A synchronized game can also be denoted in matrix form, reading

$$G = \left(\begin{array}{c|c} & \mathcal{G}^R \\ \hline \mathcal{G}^L & \mathcal{G}^S \end{array} \right).$$

In practice, we will often denote, e.g., a Left option, by G^L, instead of G^L_i, mirroring the notation for combinatorial games. In doing so, we still presume that this option G^L is uniquely identifiable, even though $G^L_i = G^L_j$ might hold for $i \neq j$. Moreover, for a Left move $G^L = G^L_i$ and a Right move $G^R = G^R_j$, we use the notation G^{L+R} for the game G^S_{ij}. Finally, if one or both of the tuples and/or the matrix consists of only one element, we oftentimes omit the brackets.

For two synchronized games G and H, if H can be constructed from G by reordering rows and/or columns, we say the games are isomorphic, writing $G \cong H$. Isomorphic games are the same in all contexts, for all intents and purposes. Note that choosing $m = 0$ or $n = 0$ (or both) is allowed, resulting in the empty matrix and one or two empty tuples.

The smallest synchronized game is $G = \{ \mid \mid \}$, which we will call 0 (zero).

If either player has no more moves to make, the game ends. If this is the case, we say that the game has been decided.

Definition 2.3.2. A synchronized game $G = \{\mathcal{G}^L \mid \mathcal{G}^S \mid \mathcal{G}^R\}$ is called *decided* if $\mathcal{G}^S$ is the empty matrix.

In decided games, it is easy to appoint a winner. If $\mathcal{G}^L \neq \emptyset$ and $\mathcal{G}^R = \emptyset$, only Left has moves remaining, so it is natural to say that Left wins. Similarly, if $\mathcal{G}^R \neq \emptyset$ while $\mathcal{G}^L \neq 0$, Right wins the game. Now, if $\mathcal{G}^L = \mathcal{G}^R = \emptyset$, that is, neither player has any remaining moves, as there is no first or second player, we declare the game to be a draw. Note that 0 is the only decided game that is a draw.

Hence, decided games can be divided into three outcome classes: $\mathcal{L}$, in which Left wins; $\mathcal{R}$, in which Right wins; and $\mathcal{D}$, in which the game ends in a draw. However, for undecided games, these classes are not exhaustive. As the synchronization of the players' moves leads to imperfect information, it turns out that the optimal strategies for both players need not be deterministic. Hence, the outcome of an undecided game may, as the name suggests, as of yet be undecided.

Example 2.3.3. Define $1 = \{0 \mid \mid \}$ and $-1 = \{ \mid \mid 0\}$, mirroring the combinatorial definition. Note that both games are decided, and $1 \in \mathcal{L}$ and $-1 \in \mathcal{R}$ as expected. Now, consider $G = \{\mathcal{G}^L \mid \mathcal{G}^S \mid \mathcal{G}^R\}$ defined by $\mathcal{G}^L = (1, 1)$, $\mathcal{G}^R = (-1, -1)$ and

$$\mathcal{G}^S = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Playing on this synchronized game is like playing on a zero-sum matrix game with outcome matrix $\mathcal{G}^S$. Hence, the optimal strategy for both players is the Nash equilibrium in which both players pick either of their options with probability $\frac{1}{2}$, leading to a win for either player with probability $\frac{1}{2}$. The game G is therefore not an element of $\mathcal{L}$ nor $\mathcal{R}$ nor $\mathcal{D}$. ◁

As the previous example shows, it might be the case that the outcome of the game depends on chance, even if both players play optimally. Hence, more outcome classes than $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{D}$ are needed to characterize all games [21]. We define $\mathcal{LD}$ to be the class of games that either end in a draw or a Left-player win under optimal play. Similarly, we define $\mathcal{RD}$ to be the class of games that result in a draw or Right win. We let $\mathcal{LR}$ be the class of games ending in a win for either player [14]. Finally, we define $\mathcal{LRD}$ as the class of games that might have any outcome under optimal play. For a game G, we denote its outcome class by $o(G)$. For G in Example 2.3.3, we conclude $o(G) = \mathcal{LR}$.

We may also categorize the outcome classes discussed above as follows. Either player can have a *winning strategy* (ws), that is, a strategy with which the game is won regardless of the moves of the other player, a *drawing strategy* (ds), which is a strategy that enforces at least a draw whatever the opponent does. If neither exists, we say the player only has *losing strategies* (ls). It cannot be the case that both players have a winning strategy, nor that one has a winning strategy and the other a drawing strategy. The resulting outcome class for the other combinations are shown in Table 2.2.

Left \ Right	ls	ds	ws
ls	$\mathcal{LR} \cup \mathcal{LRD}$	$\mathcal{RD}$	$\mathcal{R}$
ds	$\mathcal{LD}$	$\mathcal{D}$	
ws	$\mathcal{L}$		

Table 2.2: Outcome classes in synchronized games.

Just like for combinatorial games, we can define the sum of two synchronized games, as well as the negative of one.

Definition 2.3.4. Let G and H be synchronized games, and set $|\mathcal{G}^L| = m$ and $|\mathcal{G}^R| = n$. We define the (*disjunctive*) *sum* $K = G + H$ as follows: $\mathcal{K}^L$ is the concatenation of $\mathcal{G}^L$ and $\mathcal{H}^L$; $\mathcal{K}^R$ is the concatenation of $\mathcal{G}^R$ and $\mathcal{H}^R$; and

$$\mathcal{K}^S_{ij} = \begin{cases} \mathcal{G}^S_{ij} + H & \text{if } i \leq m, j \leq n, \\ G + \mathcal{H}^S_{i-m,j-n} & \text{if } i > m, j > n, \\ \mathcal{G}^L_i + \mathcal{H}^R_{j-n} & \text{if } i \leq m, j > n, \\ \mathcal{G}^R_j + \mathcal{H}^L_{i-m} & \text{if } i > m, j \leq n. \end{cases}$$

In matrix notation:

$$G + H = \left(\begin{array}{c|cc} & \mathcal{G}^R + H & G + \mathcal{H}^R \\ \hline \mathcal{G}^L + H & \mathcal{G}^S + H & \mathcal{G}^L + \mathcal{H}^R \\ G + \mathcal{H}^L & \mathcal{G}^R + \mathcal{H}^L & G + \mathcal{H}^S \end{array}\right).$$

Definition 2.3.5. Let G be a synchronized game. We define its *negative* by

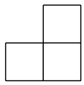
$$-G = \{-\mathcal{G}^R \mid -(\mathcal{G}^S)^\top \mid -\mathcal{G}^L\},$$

where $(\mathcal{G}^S)^\top$ denotes the transpose of $\mathcal{G}^S$.

Just like for combinatorial games, making a move on the sum of two synchronized games amounts to making a move in either one of the games. If the players make a move on different components of the sum, these moves are executed in parallel. If the players move on the same component, the corresponding synchronized move is executed.

The goal of the introduction of synchronized games is to study natural synchronized versions of combinatorial games. However, not all combinatorial games lend themselves as well to being synchronized.

Example 2.3.6. Consider the Domineering position given by

$G =$ 

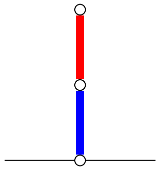
which as a combinatorial game would read $* = \{0 \mid 0\}$. For a synchronized version of this game, it would make sense to define $\mathcal{G}^L = \mathcal{G}^R = (0)$. However, it is unclear what choice would be suitable for $\mathcal{G}^S$, as the players cannot legally execute their moves simultaneously in this position, as the resulting dominoes would overlap. $\triangleleft$

Definition 2.3.7. Let G be a combinatorial game. If, for every position H of G, for every H^L and H^R, it holds that $H^L \in \mathcal{H}^{RL}$ or $H^R \in \mathcal{H}^{LR}$ or $\mathcal{H}^{LR} \cap \mathcal{H}^{RL} \neq \emptyset$, we say that G is *separable*. If, for all H^L and H^R, it holds that $\mathcal{H}^{LR} \cap \mathcal{H}^{RL} \neq \emptyset$, we say that G is *strongly separable*.

Intuitively, a separable game is a combinatorial game in which, from every position, every combination of a Left and a Right move can be executed legally in *some* order. A game is strongly separable if the moves can always be executed in *either* order. It is clear that any strongly separable game is also separable. Note that (strong) separability of a game depends on its form and is not preserved through combinatorial game equality. The definitions naturally extend to rulesets: we call a ruleset (strongly) separable if every game in the ruleset is.

Example 2.3.8. Note that Red-Blue Hackenbush, Push and Shove are all separable. Indeed, any combination of a Left and Right move can always be legally executed in some order. In Hackenbush, if both players play, for example, in the same part of a tree, the move furthest from the root is executed first. In Push and Shove, the piece closest to the edge of the playing field is moved first.

However, not all Red-Blue Hackenbush positions are strongly separable. Consider the game

$G =$ 

which may be written as $G = \{0 \mid 1\}$. We find $0^R = \emptyset$, so that $0^R \cap 1^L = \emptyset$ and G is not strongly separable.

Finally, Cherries is strongly separable. Indeed, the removal of a black and a white cherry can always happen simultaneously, so that for any Cherries game G and any G^L and G^R we have $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$. ◁

While separability of a game does depend on its form, separability is preserved if dominated or reversible options are removed. Moreover, this implies that any separable game must be a number.

Lemma 2.3.9. [19] *Let G be a separable game in canonical form. Then G is a number.*

Proof. Suppose that G is separable and in canonical form. If $\mathcal{G}^L = \emptyset$ or $\mathcal{G}^R = \emptyset$, then G is an integer (cf. [2, Problem 5.17]), and we are done. Hence, suppose $\mathcal{G}^L \neq \emptyset$ and $\mathcal{G}^R \neq \emptyset$. By the definitions of separability and canonical form, all options in $\mathcal{G}^L \cup \mathcal{G}^R$ are separable and in canonical form, and hence, by induction, numbers. As G is in canonical form, we conclude that $\mathcal{G}^L = \{G^L\}$ and $\mathcal{G}^R = \{G^R\}$ for a single Left option G^L and a single Right option G^R by domination.

If $G^R \in \mathcal{G}^{LR}$ or $G^L \in \mathcal{G}^{RL}$, then $G^L < G^R$, by both options being numbers. If $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$, then also $G^L < G^{LR} = G^{RL} < G^R$ for some options G^{LR} and G^{RL}. Hence, by the simplest number theorem, G itself is a number. □

Lemma 2.3.10. [19] *If G is separable, then so is its canonical form.*

Proof. First, note that removing a dominated option does not impair the separability, as there are now fewer pairs of options to check the definition for. By induction, we may assume all options of G are in canonical form, and therefore numbers by Lemma 2.3.9. Removing the dominated options, we end up with only one Left option G^L and one Right option G^R. Remains to show that, if either of the options is reversible, reversing out the option does not affect the separability of G.

Suppose that G^L is reversible, i.e., $G^{LR} \leq G$ and $G = \{G^{LRL} \mid G^R\}$, if G^{LRL} exists. If not, G is an integer and we are done. If it does, we show that the separability of G implies the separability of $\{G^{LRL} \mid G^R\}$.

If $G^R \in \mathcal{G}^{LR}$, then $G^R = G^{LR}$ and $G^{LRL} = G^{RL}$, as all games concerned are numbers in canonical form. Hence, $G^{LRL} \in \mathcal{G}^{LRL} = \mathcal{G}^{RL}$.

If $G^L \in \mathcal{G}^{RL}$, then $G^L = G^{RL}$, so $G^L < G^R$, which implies that $G^L < G < G^R$ by the simplest number theorem. Hence, $G^{RLR} = G^{LR} \leq G < G^R$, so G^{RL} is a reversible option of G^R, which is in contradiction with G^R being in canonical form.

Finally, suppose $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$, i.e., $G^{LR} = G^{RL}$. We first show that $G^{LR} = G$. Consider $G^{LR} - G$. Left starting play to $G^{LRL} - G < G^{LR} - G \leq 0$ loses. If Left starts playing to $G^{LR} - G^R$, Right responds to $G^{LR} - G^{RL}$ and wins. Right starting to $G^{LR} - G^R$ loses in a similar fashion. Finally, Right can start playing to $G^{LRR} - G = G^{RLR} - G$, to which Left responds to $G^{RLR} - G^R$, which is a win for Left as G^R has no reversible options. Hence, indeed $G^{LR} = G^{RL} = G$, so that also G^R is a reversible option for G, leading to $G = \{G^{LRL} \mid G^{RLR}\} \cong \{G^L \mid G^R\}$, as all positions of G except possibly G are in canonical form. Now, $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$, and the claim follows. □

Corollary 2.3.11. [19] *If G is separable, then it is a number.*

Proof. Follows immediately from Lemma 2.3.9 and Lemma 2.3.10. □

Corollary 2.3.12. *Any game or ruleset containing the game* $*$ (*as a position*)*, and therewith any impartial game, is not separable.*

The converse of Corollary 2.3.11, unfortunately, is not true.

Example 2.3.13. Consider $G = \{-2 \mid 2\}$, which equals 0 in canonical form. Clearly, G is a number. However, $\mathcal{G}^{LR} = \{-1\}$ and $\mathcal{G}^{RL} = \{1\}$, so that $G^L = -2 \notin \mathcal{G}^{RL}$, $G^R = 2 \notin \mathcal{G}^{LR}$ and $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} = \emptyset$. Hence, G is not separable. ◁

The separable games turn out to be a subgroup of the numbers. Strongly separable games are in turn a subgroup of the separable games.

Proposition 2.3.14. [18]

(i) *The set of separable games is a subgroup of* $\mathbb{D}$.
(ii) *The set of strongly separable games is a subgroup of the group of separable games.*

Proof.

(i) It is clear that 0 is separable.

Let $G_1, G_2 \in \mathbb{G}$ be separable, and consider a position $H_1 + H_2$ of $G_1 + G_2$, where H_1 is a position of G_1 and H_2 of G_2. For any Left option of the form $H_1^L + H_2$ and any Right option of the form $H_1 + H_2^R$, we find $H_1^L + H_2^R \in (\mathcal{H}_1^L + \mathcal{H}_2)^R \cap (\mathcal{H}_1 + \mathcal{H}_2^R)^L$. A similar statement holds for Left options of H_2 and Right options of H_1. For any Left option $H_1^L + H_2$ and Right option $H_1^R + H_2$, we find that, as H_1 is separable, it holds that $H_1^L + H_2 \in \mathcal{H}_1^{RL} + H_2 \subseteq (\mathcal{H}_1^R + \mathcal{H}_2)^L$ or $H_1^R + H_2 \in \mathcal{H}_1^{LR} + H_2 \subseteq (\mathcal{H}_1 + \mathcal{H}_2^L)^R$ or $(\mathcal{H}_1^L + \mathcal{H}_2)^R \cap (\mathcal{H}_1 + \mathcal{H}_2^L)^R \supseteq (\mathcal{H}_1^{LR} + H_2) \cap (\mathcal{H}_1^{RL} + H_2) \neq \emptyset$. A similar argument holds for any two options H_2^L and H_2^R. Hence $H_1 + H_2$ is separable.

Finally, let $G \in \mathbb{G}$ be separable, and consider a position $-H$ of $-G$. Noting that $(-\mathcal{H})^L = -\mathcal{H}^R$ and $(\mathcal{H})^R = -\mathcal{H}^L$, that all positions in $\mathcal{H}^L$ and $\mathcal{H}^R$ are separable, and that the definition of separability is fully symmetric, we conclude that also $-H$ must be separable.

(ii) By the reasoning above.

□

For a separable combinatorial game, any combination of two legal combinatorial moves can always be executed simultaneously in some order. Hence, the following definition is natural.

Definition 2.3.15. Let G be a separable combinatorial game. We inductively construct a *synchronized version of* G, say $\hat{G} = \{\widehat{\mathcal{G}^L} \mid \widehat{\mathcal{G}^S} \mid \widehat{\mathcal{G}^R}\}$, as follows:

- $\widehat{\mathcal{G}^L} = \hat{\mathcal{G}}^L$;
- $\widehat{\mathcal{G}^R} = \hat{\mathcal{G}}^R$;
- For every $G_i^L \in \mathcal{G}^L$, $G_j^R \in \mathcal{G}^R$, if $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$, pick $G_{ij}^S \in \mathcal{G}^{LR} \cap \mathcal{G}^{RL}$ and set $\widehat{G_{ij}^S} = \hat{G}_{ij}^S$. Otherwise, if $G_i^L \in \mathcal{G}^{RL}$, set $\widehat{G_{ij}^S} = \hat{G}_i^L$. Otherwise $G_j^R \in \mathcal{G}^{LR}$ and set $\widehat{G_{ij}^S} = \hat{G}_j^R$.

Example 2.3.16. Consider $G = \{0 \mid 1\}$. There is only one synchronized version of this game, being $\hat{G} = \{0 \mid 0 \mid 1\}$. ◁

Note that, though Definition 2.3.15 gives a way to synchronize a formal combinatorial game, it is not always directly applicable to games defined via a ruleset. Consider the game defined by the Hackenbush position in Example 1.2.2. Written as a formal combinatorial game, the moves on both stalks being indistinguishable in the sets of options, the game amounts to $\{0 \mid 1\}$, with both players effectively having only one option. However, in constructing the synchronized version of this game, we do consider the two possible moves for both players as being distinct options, effectively using the game tree rather than the set-theoretic definition of a game.

Synchronized versions of separable rulesets as defined in this way are always unique. However, for synchronized versions of formal combinatorial games, this does not always need to be the case.

Example 2.3.17. Let $G = \{\{ \mid 0, 1\} \mid \{0, 1 \mid \}\}$. This game is strongly separable, as $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} = \{0, 1\} \neq \emptyset$, and every position besides the first is decided. Hence, G can be synchronized. However, there are *two* truly different synchronized versions of G: $G_1 = \{G^L \mid (0) \mid G^R\}$ and $G_2 = \{G^L \mid (1) \mid G^R\}$. We find that $G_1 \in \mathcal{D}$, whereas $G_2 \in \mathcal{L}$. ◁

Example 2.3.17 shows that, even for strongly separable games, if the original game lies in $\mathcal{P}$, not much can be said about the outcome class of the synchronized version. However, for the other possible combinatorial outcome classes, we do have the following result.

Theorem 2.3.18. [18] *Let G be a strongly separable game and let $\hat{G}$ be a synchronized version of G.*

(i) *If $G \in \mathcal{L}$, then $\hat{G} \in \mathcal{L}$.*
(ii) *If $G \in \mathcal{R}$, then $\hat{G} \in \mathcal{R}$.*

Proof. We prove (i); the argument for (ii) is the same. Let $G \in \mathcal{L}$ and consider $\hat{G}$. In particular, G is a win for Left moving first. Hence, there is some $G_i^L \in \mathcal{G}^L$ such that for any $G_i^{LR} \in \mathcal{G}^{LR}$, it must then hold that G_i^{LR} is also a win for Left moving first. By induction, $G_{ij}^S \in \mathcal{L}$ for all $G_j^R \in \mathcal{G}^R$. □

The following example demonstrates that the above theorem fails for games which are not strongly separable.

Example 2.3.19. Let $G = \{0 \mid 1\}$ be the separable RB-Hackenbush position as depicted in Example 2.3.8, with synchronized version $\hat{G} = \{0 \mid (0) \mid 1\}$. We find that, while $G \in \mathcal{L}$, it holds that $\hat{G} \in \mathcal{D}$.

Next, consider $G + G$, and its unique synchronized version

$$\widehat{G+G} = \left(\begin{array}{c|cc} & 1+G & G+1 \\ \hline G & G & 1 \\ G & 1 & G \end{array} \right).$$

Like in Example 2.3.3, it is clear that the optimal strategy for both players is to play on either copy of G with probability $\frac{1}{2}$, leading to a win for Left or a draw, both with probability $\frac{1}{2}$. Hence, $\widehat{G+G} \in \mathcal{LD}$. This example also highlights the fact that, for synchronized games, problems with regard to determining the outcome class of sums of games may arise, even if the outcome classes of the components of the sum are known. ◁

For combinatorial games, there is a well-defined notion of (in)equality which aids greatly in speaking of "optimal" strategies for both players, and "values" of a game, even in the context of taking disjunctive sums. However, as illustrated by the above example, as even determining the outcome class of a synchronization of a sum of games may be confusing, it may be expected that finding useful definitions of (in)equality of synchronized games is challenging. We present two ways of approaching this problem, which we will call combinatorial synchronization and Nash synchronization.

2.3.2 Combinatorial synchronization

The first way of defining a notion of value depends on a synchronized version of equality, and is based on [18, 21]. The definition essentially mirrors the combinatorial one, and as such we will call it *combinatorial synchronization* of a game.

Definition 2.3.20. Let G and H be synchronized games. We say $G = H$ if $o(G + X) = o(H + X)$ for all synchronized games X.

Note that this definition indeed bestows an equivalence relation on the set of synchronized games. In practice, we identify a game by the 'simplest' game it is equivalent to, and call this its value. However, though intuitive and allowing for a rich analysis in some cases, this definition of synchronized equality does not enjoy all the properties of combinatorial equality.

Example 2.3.21. We show that $G - G$ need not necessarily equal $0 = \{ \mid \mid \}$. Let $G = 1$ as synchronized game, and take $X = \{-2 \mid 2 \mid -2\}$. Consider $G - G$.

We find that

$$G - G + X = 1 - 1 + X = \left(\begin{array}{c|cc} & 1+X & 1-1-2 \\ \hline -1+X & X & -1-2 \\ 1-1-2 & 1-2 & 1-1+2 \end{array}\right).$$

Looking at the outcome classes for the synchronized moves, we find that Left wins in the top-left and bottom-right entry, and that Right wins in the other two entries. Hence, Left nor Right has a winning strategy; we find $o(G - G + X) = \mathcal{LR}$. However, $o(0 + X) = o(X) = \mathcal{L}$. We thus find $o(G - G + X) \neq o(0 + X)$, so $G - G \neq 0$ by definition. ◁

While the above example shows that equality might fail to hold in instances where we would expect it to, the following example shows that sometimes equality holds while we may not want it to.

Example 2.3.22. Consider the synchronized games

$$G = \left(\begin{array}{c|cc} & 0 & 0 \\ \hline 0 & 1 & -1 \\ 0 & -1 & 1 \end{array}\right)$$

and

$$H = \left(\begin{array}{c|ccc} & 0 & 0 & 0 \\ \hline 0 & 1 & -1 & -1 \\ 0 & -1 & 1 & -1 \\ 0 & -1 & -1 & 1 \end{array}\right).$$

To conclude that $G = H$, note that, for any arbitrary synchronized game X, any strategy for $G + X$ can be converted to a strategy for $H + X$ and vice versa. If a player plays on G in $G + X$, it is always best to play any available move with equal probability, in this case $\frac{1}{2}$. If this is the case in some position of $G + X$, the corresponding strategy for $H + X$ is to make any of the three available moves in H with probability $\frac{2}{3}p$. The possible outcomes of the game then remain unchanged, showing that $o(G + X) = o(H + X)$ and thus $G = H$ in synchronized sense.

However, considering the games as zero-sum games, we find that G would have Nash value 0, as both players win with equal probability, whereas H has Nash value $-\frac{1}{3}$ with Right winning with probability $\frac{2}{3}$. Hence, even though $G = H$ by definition, the games do not truly have the same behavior. ◁

Like for combinatorial games, we can define a partial order on the outcome classes of synchronized games, as shown in Figure 2.4. This order on the outcome classes implies a natural definition of a partial order on the set of synchronized games.

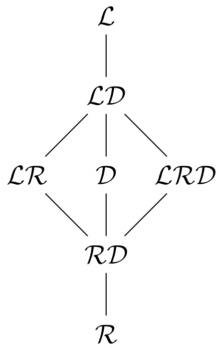


Figure 2.4: The partial order on the synchronized outcome classes.

Definition 2.3.23. Let G and H be synchronized games. We say $G \leq H$ if $o(G + X) \leq o(H + X)$ for all synchronized games X.

We will further explore combinatorial synchronization in Chapter 6. We will see that, for some games, a more useful analysis can be obtained if we gather the outcome classes $\mathcal{D}$, $\mathcal{LD}$, $\mathcal{RD}$, $\mathcal{LR}$ and $\mathcal{LRD}$ into one outcome class $\mathcal{U}$. By this change, the equivalence classes of synchronized equality become (much) larger, i.e., a game is equal to more other games than before.

2.3.3 Nash synchronization

The second proposed method of defining a notion of value for synchronized games, which we will call *Nash synchronization*, is an attempt to solve the problems encountered in combinatorial synchronization. Moreover, it better explicitly captures the inherent non-determinism in the optimal strategies for synchronized games. The definition relies on the choice of a function which assigns a value to every decided game.

Definition 2.3.24. Consider a synchronized version of a combinatorial ruleset, with decided positions D. We call $f\colon D \to \mathbb{R}$ a *value function* if it has the following four properties:

(i) For $H \in D$ with $H \in \mathcal{L}$, we have $f(H) > 0$. Moreover, if every position of H is a decided win for Left, we have $f(H) = \mathrm{Can}(H)$, identifying the game with its fractional value embedded on the real line.
(ii) For $H \in D$ with $H \in \mathcal{R}$, we have $f(H) < 0$. Moreover, if every position of H is a decided win for Right, we have $f(H) = \mathrm{Can}(H)$.
(iii) For $H \in D$ with $\mathcal{H}^L = \mathcal{H}^R = \mathcal{H}^S = \emptyset$, we have $f(H) = 0$.
(iv) For $H \in D$, we have $f(-H) = -f(H)$.

Definition 2.3.25. Consider a synchronized version of a combinatorial ruleset and let f be a value function for the ruleset. For every game G in the ruleset, we define its *Nash value* $v(G)$ to be $v(G) = f(G)$ if G is decided, or the Nash value of G as a zero-sum game otherwise.

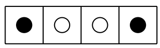
Example 2.3.26. Consider the synchronized game $H := \widehat{G+G}$ as in Example 2.3.19. The decided positions in this game are $\{1 \mid \mid \}$, $\{0 \mid \mid \}$ and $\{ \mid \mid \}$, which should be given values 2, 1 and 0, respectively, by the first three requirements in Definition 2.3.24. Hence, as a zero-sum game, we may write

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The unique Nash equilibrium for this game has value $\frac{1}{2}$, so we conclude that $v(H) = \frac{1}{2}$. ◁

Extending the above example, note that for any RB-Hackenbush game, there is no choice in the definition of the value function: any position consisting of n edges of only one color must be assigned value n or $-n$, depending on the edges being blue or red, respectively. Hence, the Nash value for any RB-Hackenbush game is uniquely determined by our definition of a value function. However, this is not the case for every ruleset.

Example 2.3.27. Consider the following game of Cherries:



Combinatorially, the game reads

$$G = \{\{-2 \mid \{-1 \mid 1\}\} \mid \} \in \mathcal{L};$$

its unique synchronized version is

$$\hat{G} = \{\{-2 \mid -1 \mid \{-1 \mid 0 \mid 1\}\} \mid \mid \} \in \mathcal{L}.$$

Being a decided game, we need to assign a value $f(\hat{G})$ via Definition 2.3.24. However, the only restriction provided by this definition is that $f(\hat{G}) > 0$ should hold. Comparing to the game H defined by

●	○	○	○	●

we find that, even though $G = H$ in combinatorial sense, and $G - 1 < H - 1$ in combinatorially synchronized sense, the values $f(G)$ and $f(H)$ can potentially be ordered in any way. ◁

The example above shows that decided games that contain undecided positions pose a problem in the definition of a useful value function. Hence, we propose the following restriction on the class of games for which defining a value function makes sense.

Definition 2.3.28. Let G be a decided synchronized game. If every position H of G is decided, and $o(H) = o(G)$ for all positions H of G, we say G is *terminal*.

Definition 2.3.29. Let G be a synchronized game. If every decided position of G is terminal, we call G *rebound-free*.

The definition of rebound-free games extends to rulesets: a ruleset is called rebound-free if every game in it is. An example of a rebound-free ruleset is that of synchronized RB-Hackenbush. It is clear that choosing $f(G) = n$ resp. $-n$ for a synchronized RB-Hackenbush position consisting of n blue or red edges is the only valid definition of a value function. The ruleset for synchronized cherries is not rebound-free.

Even a value function for a rebound-free ruleset does not enjoy all properties that one would wish, such as respecting taking sums of games.

Example 2.3.30. Consider RB-Hackenbush with its unique value function, and let $\hat{G}$ be the synchronized game defined in Example 2.3.8. It is clear that $v(\hat{G}) = 0$; both players pick their own edge with probability 1 on the first and only turn, resulting in the empty game and thus a draw. However, for the game $\hat{G} + \hat{G} = \widehat{G + G}$, we have seen in Example 2.3.26 that $v(\hat{G} + \hat{G}) = \frac{1}{2} \neq v(\hat{G}) + v(\hat{G})$. ◁

We do have the following useful properties.

Proposition 2.3.31. *Let R be a synchronized version of a separable combinatorial ruleset, let f be a value function and let $G \in R$. Then $v(G - G) = 0$.*

Proof. Viewing the game $G - G$ as a zero-sum game, we find $v(G - G) = v(-(G - G)) = -v(G - G)$, so $v(G - G) = 0$. ☐

Theorem 2.3.32. *Let R be a synchronized version of a separable combinatorial ruleset, let f be a value function and let $G \in R$. Then for every $G^L \in \mathcal{G}^L$ and for every $G^R \in \mathcal{G}^R$, we have $v(G^L) \leq v(G) \leq v(G^R)$.*

Proof. We prove the first inequality. Let $G^L \in \mathcal{G}^L$ be arbitrary. Pick $G^R \in \mathcal{G}^R$ such that $v(G^{L+R})$ is minimized, denoting G^{L+R} for the synchronized move associated to Left picking G^L and Right G^R. Then $v(G^{L+R}) \leq v(G)$. First, note that if $G^L \in \mathcal{G}^{RL}$, we have $G^{L+R} = G^L$ and we are done.

Hence, suppose that this is not the case, so that $G^R \in \mathcal{G}^{LR}$ or $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$ by G being separable. In either case, $G^{L+R} = G^{LR}$ is legal. Now, let G^{LL} be arbitrary, and consider $G^{L(L+R)}$. Again, $G^{L(L+R)} = G^{LLR}$ or $G^{L(L+R)} = G^{LRL}$ must hold (or both).

If $G^{L(L+R)} = G^{LRL}$, then, by induction,

$$v(G^{L(L+R)}) = v(G^{LRL}) = v((G^{LR})^L) \leq v(G^{LR}) \leq v(G).$$

Otherwise, $G^{L(L+R)} = G^{LLR} = G^{LR}$ must hold, so again $v(G^{L(L+R)}) = G^{LR} \leq v(G)$. Hence, for any Left move from G^L, we find that $v(G^{L(L+R)}) \leq v(G)$, so $v(G^L) \leq v(G)$. ☐

In Chapter 7, we will examine some separable games in more detail. The results from this chapter give rise to the following conjectures.

Conjecture 2.3.33. *Let R be a rebound-free synchronized version of a separable combinatorial ruleset, let f be a value function, let $G \in R$ be arbitrary and let $H \in R$ be terminal. Then $v(G + H) = v(G) + v(H)$.*

Conjecture 2.3.34. *Let R be a rebound-free synchronized version of a separable combinatorial ruleset, let f be a value function and let $G \in R$. Then*

$$\lim_{n \to \infty} \frac{v(n \cdot G)}{n} = \mathrm{Can}(G).$$

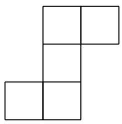
Hence, it seems that, when looking at many copies, some synchronized games behave very much like their combinatorial counterparts. The intuition behind this could be that the probability of the two players playing on the same component will be small, for a large number of components. Hence, the game behaves as being combinatorial.

2.3.4 Synchronization of non-separable games

So far, we have only considered the synchronization of separable combinatorial games, as attempting to synchronize non-separable games may prove problematic, demonstrated by Example 2.3.6. However, it is not impossible. One needs to find a way to deal with the players trying to execute two combinatorial moves simultaneously which cannot be executed legally in any order.

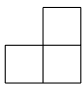
Starting from a ruleset, it is often possible to give a natural interpretation to allowing both moves to be executed anyway, even if this does not lead to a legal position in the underlying combinatorial game. For example, in Domineering, we can allow the placement of two overlapping dominoes if placed simultaneously during the same turn. Under this regime, the position in Example 2.3.6 would be synchronized to $\{0 \mid 0 \mid 0\} \in \mathcal{D}$. Though natural, there are drawbacks, such as the fact that Conjecture 2.3.34 might fail to hold for Nash synchronization of the game.

Example 2.3.35. Let G be the synchronized version of the Domineering position



We look at the variant in which simultaneously placing overlapping dominoes is allowed, and the value function f is uniquely defined by Definition 2.3.24. Then $v(n \cdot G) = -\frac{n}{2}$.

Indeed, we can proceed by induction on n. The base cases $n = 1, 2$ are easy to check. For the induction step, note that, from $n \cdot G$, the game is moved to $(n-1) \cdot G - 1$ or $(n-1) \cdot G$ if both players play on the same copy of G, or $(n-2) \cdot G + H - 1$, with

$H =$ 

if the players play on a different copy. We will show, again by induction, that for $(n-2) \cdot G + H$, there is a Nash equilibrium in which both players play on H with probability 1. Note that the induction hypothesis implies that $v(k \cdot G + H) = -\frac{k}{2}$ for $k < n$. The base case $n = 3$ is easily checked.

If both players play on the same copy of G, the result is either $(n-3) \cdot G + H - 1$, with value $-\frac{n}{2} + \frac{1}{2}$, or $(n-3) \cdot G + H$, with value $-\frac{n}{2} + \frac{3}{2}$. If both players play on a different copy of G, the result is always $(n-4) \cdot G - 1 + 2 \cdot H$. Note that

Right can force a value of at most $-\frac{n}{2}+1$ by playing on one of the copies of H; the result is $(n-5)\cdot G-2+H$ with value $-\frac{n}{2}+\frac{1}{2}$ if Left plays on a copy of G, $(n-4)\cdot G-1$ with value $-\frac{n}{2}+1$ if Left plays on the other copy of H, or $(n-4)\cdot G-1+H$ with the same value if Left plays on the same copy of G.

If both players play on H, the result is $(n-2)\cdot G$, with value $-\frac{n}{2}+1$. If Left plays on a copy of G and Right on H, the result is $(n-3)\cdot G-1$ with value $-\frac{n}{2}+\frac{1}{2}$. Finally, if Left plays on H and Right on a copy of G, the result is $(n-3)\cdot G+H$ with value $-\frac{n}{2}+\frac{3}{2}$.

Comparing these results, we see that, for Left, it is profitable to always play on H. Knowing this, the same holds for Right. Hence, we indeed have that, for $(n-2)\cdot G+H$, it is optimal for both players to play to $(n-2)\cdot G$ and continue from there. Now, writing $v_n = v(n\cdot G)$, we may thus conclude that

$$\begin{aligned} v_n &= \tfrac{1}{n}(\tfrac{1}{2}\cdot -1+\tfrac{1}{2}\cdot 0+v_{n-1})+\tfrac{n-1}{n}(v_{n-2}-1)\\ &= \tfrac{1}{n}(v_{n-1}-\tfrac{1}{2})+\tfrac{n-1}{n}(v_{n-2}-1). \end{aligned}$$

By induction, it follows that

$$\begin{aligned} v_n &= \tfrac{1}{n}(v_{n-1}-\tfrac{1}{2})+\tfrac{n-1}{n}(v_{n-2}-1)\\ &= \tfrac{1}{n}(-\tfrac{n-1}{2}-\tfrac{1}{2})+\tfrac{n-1}{n}(-\tfrac{n-2}{2}-1)\\ &= -\tfrac{n}{2}. \end{aligned}$$

◁

Moreover, if not working with games from a ruleset, but formal games, there is no intuition as to how to define the synchronized moves. Therefore, we propose the following, extending Definition 2.3.15.

Definition 2.3.36. Let G be a combinatorial game. We inductively construct a *synchronized version of* G, named $\hat{G} = \{\widehat{\mathcal{G}^L} \mid \widehat{\mathcal{G}^S} \mid \widehat{\mathcal{G}^R}\}$, as follows:

- $\widehat{\mathcal{G}^L} = \hat{\mathcal{G}}^L$;
- $\widehat{\mathcal{G}^R} = \hat{\mathcal{G}}^R$;
- For every $G_i^L \in \mathcal{G}^L$, $G_j^R \in \mathcal{G}^R$, if $\mathcal{G}^{LR} \cap \mathcal{G}^{RL} \neq \emptyset$, pick $G_{ij}^S \in \mathcal{G}^{LR} \cap \mathcal{G}^{RL}$ and set $\widehat{G_{ij}^S} = \hat{G}_{ij}^S$. Otherwise, if $G_i^L \in \mathcal{G}^{RL}$, set $\widehat{G_{ij}^S} = \hat{G}_i^L$. Otherwise, if $G_j^R \in \mathcal{G}^{LR}$, set $\widehat{G_{ij}^S} = \hat{G}_j^R$. Otherwise, set $\widehat{G_{ij}^S} = \hat{G}$.

If both players pick moves that cannot be executed legally in any sequential order, we disallow the move, letting the players try again. By this definition, a synchronized version of a short combinatorial game may become loopy. In the spirit of loopy games, if, on a Nash synchronized game G, both players play to G with probability 1 in every Nash equilibrium, we declare the game a draw. The game is decided, we set $G \in \mathcal{D}$, and we assign $v(G) = 0$.

Example 2.3.37. Consider H as in Example 2.3.35. Under Definition 2.3.36, the synchronized game will be $H = \{0 \mid H \mid 0\}$. With the only possibilities of the players being to move to H together, we declare H decided, and set $H \in \mathcal{D}$ and $v(H) = 0$.

The synchronized version of two copies of H reads

$$H + H = \left(\begin{array}{c|cc} & H & H \\ \hline H & H+H & 0 \\ H & 0 & H+H \end{array}\right).$$

Any pair of strategies is now a Nash equilibrium: in particular the strategy pair in which both players play on either copy of H with probability $\frac{1}{2}$. For this strategy pair, the players play to $0 \ncong H + H$ with positive probability. Hence, the value of the game is determined by the value of 0, being zero; the game is still a draw, but now not because endless repetition of the position $H + H$ would ensue. ◁

Though the above examples stem from non-separable combinatorial games, we may use the idea of repeating a game to extend the definition of general zero-sum games. We denote such a *repeatable* game by writing at least one $*$ as an entry in the payoff matrix; if the players pick the row and column corresponding to this $*$, the players play the game again. If both players pick a $*$ with probability 1 according to their strategies, we define the value of the game to be 0. With this introduction, unfortunately, games no longer always have a Nash equilibrium.

Example 2.3.38. Consider the zero-sum game given by the payoff matrix

$$G = \begin{pmatrix} * & -1 \\ -1 & 10 \end{pmatrix},$$

signifying that if Left picks the first row and Right the first column, the players try again. We claim that G does not have a Nash equilibrium, i.e., for any pair of strategies (p, q), p denoting the probability for Left picking the first row and

q denoting the probability for Right picking the first column, either player can improve their outcome by deviating. This is summarized in Table 2.3, where the P' column indicates which player deviates and the μ' column shows the new strategy followed by this player.

Hence, there is no pair of strategies for which neither player can gain from deviating. The example can be extended to a payoff matrix of arbitrary size by defining $g_{11} = *$, $g_{1j} = g_{i1} = -1$ and $g_{ij} = 10$ for all i, j. $\triangleleft$

We return to these repeatable games in Section 7.5.

Strategy	Value	P'	μ'	New value
$p = q = 1$	$v = 0$	R	$q \in [0, 1)$	$v = -1$
$p = q = 0$	$v = 10$	R	$q = 1$	$v' = -1$
$p = 1, q = 0$	$v = -1$	L	$p = 0$	$v' = 10$
$p = 0, q = 1$	$v = -1$	L	$p = 1$	$v' = 0$
$p \in (0, 1), q = 1$	$v = -1$	L	$p = 1$	$v' = 0$
$p \in (0, 1), q = 0$	$v = -p + 10(1 - p) \in (-1, 10)$	R	$q = 1$	$v' = -1$
$p = 1, q \in (0, 1)$	$v = -1$	L	$p = 0$	$v' = -q + 10(1 - q) \in (-1, 10)$
$p = 0, q \in (0, 1)$	$v = -q + 10(1 - q) \in (-1, 10)$	R	$q = 1$	$v' = -1$
$p, q \in (0, 1)$	$v = \frac{-p(1-q)-(1-p)q+10(1-p)(1-q)}{1-pq} \in (-1, 10)$	R	$q = 1$	$v' = -1$

Table 2.3: The values obtained for every possible strategy pair, the player P' who can deviate, their new strategy μ', and the new value obtained.

Chapter 3

Hackenbush variants

In this chapter, we explore two combinatorial variants of Red-Blue Hackenbush. In Section 3.1, we consider Childish Hackenbush, briefly introduced in [1]. The contents are largely based on joint work with Nienke Burgers [14]. In Section 3.2, we consider the new variant Uncolored Hackenbush.

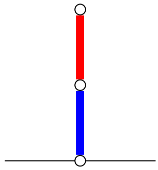
3.1 Childish Hackenbush

In this variant of the game, described in [1], moves that disconnect a part of the configuration from the ground are not allowed. The results in the following sections are largely adapted from [14].

3.1.1 Rules

A position of Childish Hackenbush is the same as one in regular Hackenbush. However, both players are only allowed to cut an edge if this does not result into a part of the graph becoming disconnected from the floor. We focus only on the Red-Blue version of the game.

Example 3.1.1. Consider the following position:

$G =$ 

In the regular version of Hackenbush, we know that $G = \frac{1}{2}$. However, for the Childish rules, we find that $G = \{ \mid 1\} = 0$, as Left is not allowed to cut her blue edge on the first turn, as this would result in the red edge being disconnected from the ground. ◁

3.1.2 Stalks

In this section, we give an exhaustive characterization of Childish Hackenbush stalks, that is, positions essentially consisting of a line graph. We make extensive use of what we will call the *sign expansion* of a Hackenbush stalk, which also lies at the base of Thea van Roode's method to compute the number value of a Red-Blue Hackenbush stalk under regular rules [29].

Definition 3.1.2. Let G be a Hackenbush stalk. Its unique *sign expansion*, denoted by $G_\pm$, is a string of $+$'s and $-$'s: a $+$ for every blue edge, and a $-$ for every red edge, reading from the ground.

Example 3.1.3. For the game G in Example 3.1.1, we have $G_\pm = +-$. ◁

We call every sequence of ≥ 1 consecutive copies of the same symbol ($+$ or $-$) a *block*, and the number of identical symbols in a block its *length*. A block of length at least 2 is called a *series*. A series not being the last block in the sign expansion is called a *non-terminal* series.

Theorem 3.1.4. *Let G be a Childish Hackenbush stalk and let $G_\pm = x_1x_2\ldots x_n$ be its sign expansion. Let a be the length of the last block of $G_\pm$, and let x be the symbol occurring in the last non-terminal series, if any. If no non-terminal series exist, $x = x_1$. Then, concatenating signs and numbers into a string expression, we find*

$$G = \begin{cases} x_n a & \text{if } x = x_n, \\ x_n a x 1 & \text{if } x \neq x_n. \end{cases}$$

Example 3.1.5. Let $G_\pm = + - + - - + - + - + +-$. In the terminology of Theorem 3.1.4, we have $a = 1$, $x = +$ (occurring in the second to last block, being a non-terminal series of length 2) and $x_n = -$. Hence, $G = -1+1 = 0$. ◁

Proof of Theorem 3.1.4. Assume first that $x_n = +$. We proceed by induction on the size of the last block in $G_\pm$, the base case being size 1.

First, suppose $x = -$. If Right starts, he immediately loses. If Left starts, the players alternate turns until the last edge defining x is taken, leaving Left to move on either a string ending in a red edge, or the empty game, losing regardless. Hence, the starting player loses, so $G = 0$, in accordance with the theorem.

Next, suppose $x = +$ and consider $G - 1$. If Right starts, he must move on -1 in the first turn, after which the game proceeds on G as above, resulting in a loss for Right. If Left starts, after having played on G, Right can respond by playing on -1 and win. Hence, $G - 1 = 0$, so $G = 1$.

Now let the size of the last block be $k > 1$. Again, first suppose that $x = -$. Note that $\mathcal{G}^L = \{G'\}$, where $G'_\pm = x_1 \dots x_{n-1}$, and $\mathcal{G}^R = \emptyset$. By induction, noting that the configuration of blocks in $G_\pm$ does not change by removing the last edge, $G' = k - 2$, so $G = \{k-2 \mid \} = k - 1$. If $x = +$, by a similar argument, $G = \{k - 1 \mid \} = k$.

The argument for $x_n = -$ is symmetric. □

3.1.3 Trees

We continue by giving a complete characterization of Childish Hackenbush trees, determining their value in an algorithmic way. For a Childish Hackenbush tree, we call a sequence of edges connecting two nodes of degree larger than 2, or connecting such a node to the ground or a leaf node, a *string*. Note that a stalk consists of a single string. A string originating in a leaf node is called a *branch*.

Algorithm 3.1.6. *Let G be a Childish Hackenbush tree. We compute G as follows:*

(i) *Assign a value to every branch of G using Theorem 3.1.4, acting as if the node of degree larger than 2 in which the branch originates is the ground, if applicable.*

(ii) *If, in a vertex in G of degree k,*

- *$k - 1$ of the outgoing edges are part of a string which has already been assigned a value;*
- *these strings do not contain a non-terminal series in their sign expansions;*
- *the first symbol in the sign expansion of these strings is not equal to the last symbol of the sign expansion of the k-th string,*

then we assign a value the k-th string. If it contains a series, the value is 1 if both the last symbol and the sign of the last series in the sign expansion of the string is $+$, -1 if both are $-$, and 0 otherwise. If not, go to step (iii).

(iii) *Assign a value to the k-th string according to Theorem 3.1.4, again regarding the vertex at the other end as the ground. If this vertex now meets the requirements of step (ii), go to step (ii).*

(iv) *The sum of the computed values is the value of G.*

Theorem 3.1.7. *Algorithm 3.1.6 is correct.*

Proof. To prove the theorem, we start by proving two claims.

Claim 1. Any path from a leaf to some vertex with degree at least 3 in the tree consisting of valued strings has total value 0, 1 or -1. The total value of the path is 0 or 1 if the first non-zero string has value 1, and 0 or -1 if it has value -1.

Proof of Claim 1. Consider such a path, consisting of k strings, say. For all strings to be valued by the above procedure, at least the first $k-1$ strings must not have a series. Moreover, any string of which the sign expansion ends in $+$ must be followed by a string with sign expansion starting with $-$ and vice versa. Hence, by Theorem 3.1.4, after a string of value 1 (starting and ending in $+$), we must encounter at least one string being valued -1 (starting and ending in $-$) before encountering a string of value 1 again. A similar arument holds for the value of the k-th valued string.

Claim 2. Let $w(G)$ be the value assigned to G by the above procedure. Then $w(G^L) < w(G)$ for all $G^L \in \mathcal{G}^L$, and $w(G^R) > w(G)$ for all $G^R \in \mathcal{G}^R$.

Proof of Claim 2. Without loss of generality, consider $\mathcal{G}^L$. Making a move on G, Left has three truly different options.

First, Left may move to G^L by removing an edge from a branch having a non-terminal series, resulting in the branch still having a non-terminal series. Now, in computing the value of G^L, the only difference is the value of the branch in which Left moved. If the sign expansion still ends in $+$, the value is 1 lower. Otherwise, its value becomes 0 (if the last non-terminal series has sign $+$, the branch having value 1 in G) or negative (if the last non-terminal series has sign $-$, the branch having value 0 in G). In any case, the claim holds.

Second, Left may move to G^L by removing an edge from a branch B having a non-terminal series, resulting in B no longer having a non-terminal series. This can only be the case if $B_\pm$ ended in a series of $-$, followed by a single $+$, with no other non-terminal series present. The branch B now goes from having value 0 to having a value at most equal to -1. If in G^L no previously unvalued string is now valued, this proves the claim.

Hence, suppose at least one more string is valued in G^L compared to G, ending in the vertex where B begins. If $B_\pm$ begins with $-$, having value at most -2, the newly valued series ends with $+$, so that the path of all potentially newly valued strings has value at most 1 by Claim 1. Hence, the claim is true. Similarly, if $B_\pm$ begins with $+$, B having value at most -1, the newly valued path adds at most 0 to the value of the tree.

Third, Left may move to G^L by removing an edge from a branch which does not have a non-terminal series. Again, in computing the value of G^L, the only difference is the value of this branch in which Left moved, which always decreases by exactly 1.

Proof of the theorem. Induction on the number of edges in the tree, the base case being the empty game having value 0. Consider a tree G and suppose first that $w(G) > 0$. If there is a branch with a positive terminal series in its sign expansion, Left may play on this branch to $w(G) - 1$. If there is a branch with a blue end of which the last non-terminal series is positive, or which does not have a non-terminal series and starts with a blue edge, Left may also play on this branch to $w(G) - 1$.

If there are no such branches, note that all branches have value 0 or lower. Hence, for $w(G) > 0$ to hold, there must be some path from a leaf to an internal vertex of degree at least 3 having value 1. The first non-zero string encountered starting at the leaf must have value 1. Hence, the edge connected to the leaf must be blue. By playing on this edge, Left can play to $w(G) - 1$.

Hence, we see that $w(G) - 1 \in \mathcal{G}^L$ regardless. Moreover, note that $w(G)$ and therewith $w(G) - 1$ is an integer by construction, and, $w(G^L) < w(G)$ holding by the second claim, moving to $w(G) - 1$ is dominating for Left. Now, if $\mathcal{G}^R = \emptyset$, we find that $G = \{w(G) - 1 \mid \} = w(G)$ by induction. Otherwise, by the second claim, we have that $w(G^R) > w(G)$ for all $G^R \in \mathcal{G}^R$. Now, G^R again being an integer by construction, we have that $G = w(G)$ by induction.

The case $w(G) < 0$ is symmetrical. Finally, if $w(G) = 0$, by induction and the second claim, it immediately follows that $G = 0 = w(G)$. □

Example 3.1.8. Consider the Childish Hackenbush position G as depicted in Figure 3.1.

We first value the branches using Theorem 3.1.4, finding values of 1, -2, 3, 0 and 0, going from left to right. Now, both the bottommost edge of the second

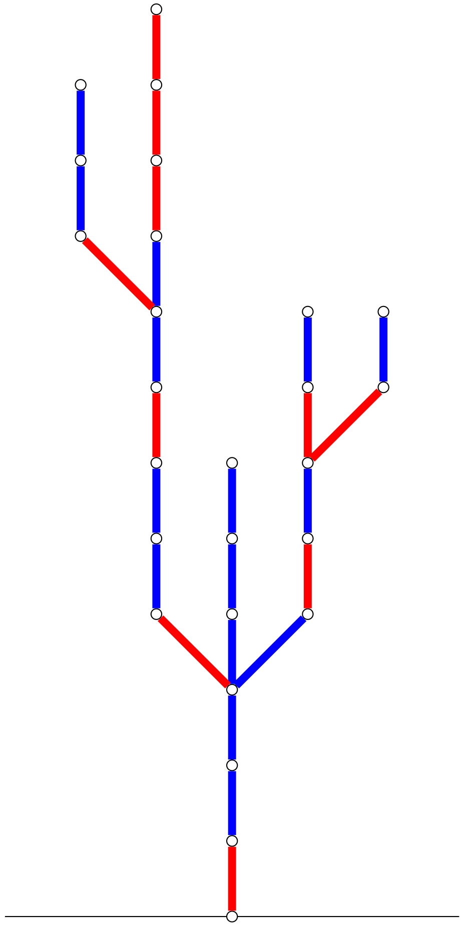


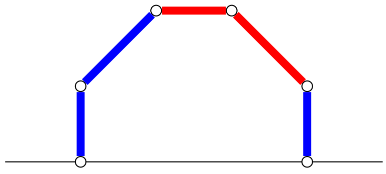
Figure 3.1: A large Childish Hackenbush position.

string and the topmost edge of the string below that are blue, so by the third bullet of step 2, the lower string is not assigned a value. The rightmost vertex rooting the last two strings does meet the requirements of step 2. It does not contain a series, so we proceed to step 3 and value it using Theorem 3.1.4, finding a value of 1. Summing the computed values yields a result of 3, hence $G = 3$. ◁

3.1.4 Cycles

By the results in the previous sections, in Childish Hackenbush, stalks and trees are all integer-valued. This turns out not to be the case if we allow cycles in the graph.

Example 3.1.9. Consider the following position G:

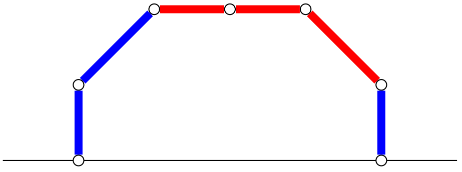


It is readily verified that

$$G = \{0, 0, -1 \mid 2, 1\} = \{0 \mid 1\} = \tfrac{1}{2}.$$

◁

Example 3.1.10. Consider the following position G:



We compute

$$G = \{0, -1, -2 \mid 1, 0, 0\} = \{0 \mid 0\} = *.$$

◁

Unlike in regular Red-Blue Hackenbush, we thus see that Childish Hackenbush allows for non-numeric values.

In [1], it is shown that Red-Blue Hackenbush is NP-hard by a reduction from the Steiner tree problem to that of determining the value of a *Redwood bed*. Such a Redwood bed is a Hackenbush position of the form as shown in Figure 3.2, where G is any graph consisting of solely red edges.

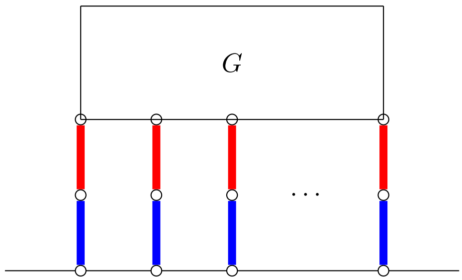


Figure 3.2: The form of a redwood bed. G is a graph consisting only of red edges.

However, determining the value of these beds turns out to be simple in the childish version of the game.

Proposition 3.1.11. *Let G be a Childish Redwood bed with m blue edges and n red edges. Then $G = m - n$.*

Proof. Note that $m \leq n$. We proceed by induction on m and n. For $m = n = 1$, we have $G = \{ \mid 1\} = 0$, as shown in Example 3.1.1. For $n > 1$ fixed, we find $G = \{ \mid G^R\}$, where G^R is a bed with 1 blue and $n - 1$ red edges. Hence $G = \{ \mid m - (n-1)\} = m - n$.

Now, let G be some bed with $m > 1$ blue edges. If $m = n$, we find $G = 0$, as it is the disjunctive sum of m copies of the bed with $m = n = 1$.

Next, let the bed have $n > m$ red edges. Left can only cut an edge which does not disconnect a part of the red mattress from the floor, resulting in a bed having $m - 1$ blue and n red edges. Hence, $\mathcal{G}^L = \emptyset$ or $\mathcal{G}^L = \{m - n - 1\}$.

A move of red can either remove a single red edge, leading to a bed with value $m-(n-1)$, or it can split the bed into two. In the latter case, say the move results in one bed with x blue and y red edges, and the other in a bed with $m - x$ blue and $n - y - 1$ red edges, so that the total game has m blue and n red edges. These two beds together then have value $x - y + (m - x) - (n - y - 1) = m - (n - 1)$. Hence $G = \{m - n - 1 \mid m - n + 1\} = m - n$. □

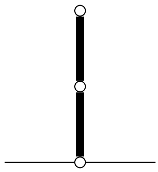
3.2 Uncolored Hackenbush

In this variant of the game, a given graph first needs to be colored before commencing normal play. We start out with an explanation of the rules of the game, followed by results on increasingly complex classes of graphs.

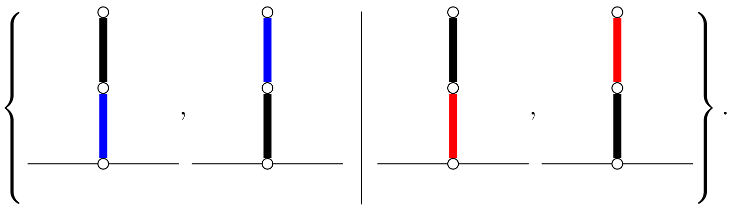
3.2.1 Rules

An initial game position for Uncolored Hackenbush is a Hackenbush position in which none of the edges is colored — or, in other words, each edge is colored black. As long as there is still at least one black edge left, a legal move of Left is to color a black edge blue, and a legal move of Right is to color a black edge red. Once all the edges in the graph have been assigned a color (blue or red), game continues as in the regular combinatorial version of Hackenbush.

Example 3.2.1. Consider the following starting configuration G of a game of Uncolored Hackenbush:

$G =$ 

Writing out the options, we find

$G =$ 

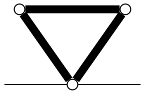
Numbering these options G_1 through G_4, by inspection, we find that $G_1 = \{2 \mid \frac{1}{2}\}$, $G_2 = \{2 \mid -\frac{1}{2}\}$, $G_3 = \{-\frac{1}{2} \mid -2\}$ and $G_4 = \{\frac{1}{2} \mid -2\}$. Removing dominated options, we conclude that

$$G = \{G_1 \mid G_3\} = \left\{2 \mid \tfrac{1}{2} \parallel -\tfrac{1}{2} \mid -2\right\} = \pm\left\{2 \mid \tfrac{1}{2}\right\},$$

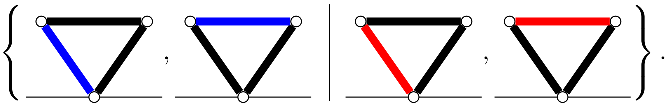
using the nested notation $\left\{2 \mid \frac{1}{2} \parallel -\frac{1}{2} \mid -2\right\} = \left\{\left\{2 \mid \frac{1}{2}\right\} \mid \left\{-\frac{1}{2} \mid -2\right\}\right\}$. ◁

We see that coloring the edge closest to the ground is an optimal move for both players. In the following sections, we will find that this is always the case in some sense.

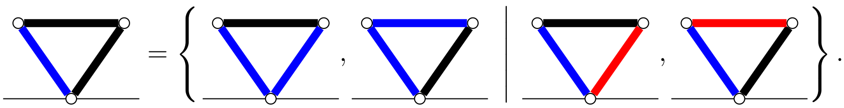
Example 3.2.2. Consider

$G =$ 

Writing out the options, using symmetry to exclude two, we find

$G =$ 

We further analyze the options.



Numbering these options G_1 through G_4, we find that $G_1 = G_2 = \{3 \mid 1\}$ and $G_3 = G_4 = \{1 \mid -1\}$. Hence, the position above is $\{3 \mid 1 \parallel 1 \mid -1\} = 1$. The same holds for the position in which the bottom two edges of the cycle are black and the top edge is blue.

The same holds for the other side, with the colors reversed, so that $G = \pm 1$. ◁

In contrast with games of regular Red-Blue Hackenbush, which are all numbers, we see that the games depicted above are both switches. We note that this is the case for every fully Uncolored Hackenbush position.

Theorem 3.2.3. *Let G be an Uncolored Hackenbush position containing only black edges. Then $G = \pm H$ for some game H.*

The statement follows from the following lemma, which is also useful in its own right.

Lemma 3.2.4. *Let G be an Uncolored Hackenbush position. Then $-G$ is G with all red edges colored blue and all blue edges colored red.*

Proof. Trivial. □

3.2.2 Stalks

In this section, we compute the value of an arbitrary Uncolored Hackenbush stalk. We start by proving that the optimal strategy for both players is to color the black edge closest to the ground.

Proposition 3.2.5. *Let G be an Uncolored Hackenbush stalk with n black edges, $n > 0$. For $k = 0, \ldots, 2^n$, denote by G_k the position in which the i-th black edge counting from the ground is colored blue if the binary expansion of k has a 0 as i-th digit and red if it has a 1 as i-th digit. Then*

$$G = \{G_0 \mid G_1 \parallel G_2 \mid \ldots \mid G_{2^n-1}\}, \tag{3.1}$$

where the $| \ldots |$ symbols are nested so that the games are successively paired.

Example 3.2.6. Let G be the Uncolored Hackenbush stalk consisting of three black edges. Proposition 3.2.5 gives that

$$\begin{aligned} G &= \{G_0 \mid G_1 \mid\mid G_2 \mid G_3 \mid\mid\mid G_4 \mid G_5 \mid\mid G_6 \mid G_7\} \\ &= \{\{\{G_0 \mid G_1\} \mid \{G_2 \mid G_3\}\} \mid \{\{G_4 \mid G_5\} \mid \{G_6 \mid G_7\}\}\}, \end{aligned}$$

where, for example, G_6 represents the stalk in which the bottom two edges are red and the top edge is blue, as $6 = 110$ in binary notation. Filling in the values we know from regular Red-Blue Hackenbush, we obtain

$$G = \{3 \mid \tfrac{3}{2} \mid\mid \tfrac{3}{4} \mid \tfrac{1}{4} \mid\mid\mid -\tfrac{1}{4} \mid -\tfrac{3}{4} \mid\mid -\tfrac{3}{2} \mid -3\} = \pm\{3 \mid \tfrac{3}{2} \mid\mid \tfrac{3}{4} \mid \tfrac{1}{4}\}.$$

◁

Proof of Proposition 3.2.5. We show by induction on n that coloring the edge closest to the ground is the dominating option for both players, the base case $n = 1$ being trivial. Note first that G indeed represents the game in which moving on the lowest edge is the only possible option for both players, which is most easily seen by picturing the possible moves as a binary tree. We label every left child of a node 0, corresponding to a move by Left coloring the lowest black edge blue, and every right child of a node 1, corresponding to a move by Right coloring the lowest black edge red.

Remains to show that coloring a different edge is not beneficial for both players. We only give the argument for Left. Suppose Left chooses to color another edge than the lowest edge blue on the first move, say edge i, playing to G'. We need to show that

$$G' \le \{G_0 \mid G_1 \mid \ldots \mid G_{2^{n-1}-1}\},$$

i.e, that $G' - \{G_0 \mid G_1 \mid \ldots \mid G_{2^{n-1}-1}\}$ is won by Right playing second. Whenever Left colors some edge other than 1 or i blue in G', Right responds on $-\{G_0 \mid G_1 \mid \ldots \mid G_{2^{n-1}-1}\}$ by coloring the corresponding edge blue, and vice versa. When Left colors edge 1 in G' blue, Right responds by coloring edge i in $-\{G_0 \mid G_1 \mid \ldots \mid G_{2^{n-1}-1}\}$ blue. The result will then eventually be $G'' - H$, where G'' and H are identical; hence, Left loses playing first. When Left colors edge i in $-\{G_0 \mid G_1 \mid \ldots \mid G_{2^{n-1}-1}\}$ red, Right responds by coloring edge 1 in G' red. The result will then be $G'' - H$ in which the only difference between G'' and H is that in G'', edge 1 is colored red and i blue, and vice versa in $-H$. Now, Left starting, Right can always respond by mirroring, except when Left cuts edge i in either game, in which case Left responds by cutting edge 1 in the same game. The result is a win for Right. □

We can, in fact, be more specific. Recall the definition of the ordinal sum from Definition 2.1.37.

Theorem 3.2.7. *Let S_n be an Uncolored Hackenbush stalk consisting of n black edges. Then S_n is determined by the following recurrence relation:*

$$\begin{cases} S_0 = 0, \\ S_n = \pm(1 : S_{n-1}), \quad n \ge 1. \end{cases}$$

The solution is (3.1), which is in canonical form. It may be rewritten explicitly as

$$S_n = \pm\{n \mid \tfrac{2(n-1)-1}{2} \mid \tfrac{4(n-2)-1}{4} \mid \tfrac{4(n-2)-3}{4} \mid \tfrac{8(n-3)-1}{8} \mid \ldots$$
$$\ldots \mid \tfrac{8(n-3)-7}{8} \mid \ldots \mid \tfrac{2^k(n-k)-i}{2^k} \mid \ldots\}.$$

Proof. It is clear that $S_0 = 0$. By Proposition 3.2.5, the dominating move for both players in S_n, $n > 0$, is to color the edge closest to the floor. Hence, $S_n = \{S_n^L \mid S_n^R\}$, where in S_n^L the lowest edge is colored blue, and likewise in S_n^R the lowest edge is colored red. By the theory on ordinal sums in [4], we find that

$$S_n = \{1 : S_{n-1} \mid -1 : S_{n-1}\}.$$

Using Lemma 3.2.4, and the properties of ordinal sums described in [2], we find that

$$\begin{aligned} S_n &= \{1 : S_{n-1} \mid -1 : S_{n-1}\} \\ &= \{1 : S_{n-1} \mid -(1 : (-S_{n-1}))\} \\ &= \{1 : S_{n-1} \mid -(1 : S_{n-1})\} \\ &= \pm(1 : S_{n-1}). \end{aligned}$$

The fact that (3.1) is a solution, is Proposition 3.2.5. To show that G is in canonical form, it suffices to show that neither the move of Left nor the move of Right is reversible. For the former, we need to prove that

$$\{G_{2^{n-2}} \mid G_{2^{n-2}+1} \mid \ldots \mid G_{2^{n-1}-1}\} > G,$$

i.e., that $G - \{G_{2^{n-2}} \mid G_{2^{n-2}+1} \mid \ldots \mid G_{2^{n-1}-1}\}$ is a win for Right, regardless of who makes the first move. Suppose first that Left starts. If Left plays on G, Right responds on G^L, playing to

$$\{G_{2^{n-2}} \mid G_{2^{n-2}+1} \mid \ldots \mid G_{2^{n-1}-1}\} - \{G_{2^{n-2}} \mid G_{2^{n-2}+1} \mid \ldots \mid G_{2^{n-1}-1}\} = 0$$

with Left moving and losing. If Left plays on $-\{G_{2^{n-2}} \mid G_{2^{n-2}+1} \mid \ldots \mid G_{2^{n-1}-1}\}$, note that all options in this game are negative, as all indices start with a 0, and all represented games are thus blue-based. By responding on G to $\{G_{2^{n-1}} \mid G_{2^{n-1}+1} \mid \ldots \mid G_{2^n-1}\}$, Right guarantees that also this component becomes negative, winning the game. To show that Right's move on G is not reversible follows a similar argument.

Finally, to show that the explicit solution is correct, we note that G_i is the sign expansion obtained by replacing every 0 in the binary expansion of i by $+$ and every 1 by $-$. Indeed, note that the G_i are precisely the number values encountered in the game tree of G. To compute $1 : G$, we replace every node in which G_i is encountered by $1 : G_i$, which is equivalent to adding a $+$ at the left side of the sign expansion represented by G_i. Similarly, to compute $-1 : G$, we replace every G_i by $-1 : G_i$, which amounts to adding a $-$ at the left side of the sign expansion. □

3.2.3 Trees

We continue by assessing trees. We start out by giving two helpful lemmas, in which we abuse the notation of $G : H$ to mean H connected to some leaf node of G.

Lemma 3.2.8. *Let $G = H : S$ be an Uncolored Hackenbush position, where S is a stalk. Coloring the edge in S closest to H dominates all other moves in S.*

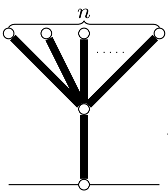
Proof. Analogous to the proof of Proposition 3.2.5. □

Lemma 3.2.9. *Let $G = S : H$ be an Uncolored Hackenbush position, where S is a stalk. Coloring the edge in S closest to the ground dominates all other moves.*

Proof. Analogous to the proof of Proposition 3.2.5. □

We first consider, in a sense, the simplest trees around.

Proposition 3.2.10. *Let*

$G =$  .

Then

$$G = \begin{cases} \pm 1, & n \text{ even}, \\ \pm\{2 \mid \frac{1}{2}\}, & n \text{ odd}. \end{cases}$$

Proof. First, let n be even and consider $G + \pm 1$, which we will show to be losing for the starting player. We only consider the case in which Left starts. If Left plays on G, the optimal move is to color the base of the tree blue. Right reacts by playing ± 1 to -1. Consequently, the players alternate turns until both players have colored half of the n branches of the tree, resulting in $(1:0) - 1 = 1 - 1 = 0$ with Left moving losing. If Left plays on ± 1 on her first move, Right reacts by claiming the base of the tree, and the result is similar.

Next, suppose $n > 1$ is odd and consider $G \pm \{2 \mid \frac{1}{2}\}$. Again, suppose Left starts. If Left colors the base of the tree G, Right reacts by playing on $\pm\{2 \mid \frac{1}{2}\}$ to $\{-\frac{1}{2} \mid -2\}$. Now, continuing play, the result is either that $\frac{n+1}{2}$ of the branches of G are colored blue, and $\frac{n-1}{2}$ red, and $\{-\frac{1}{2} \mid -2\}$ is played to -2; or that the amount of blue and red branches in G is reversed and $\{-\frac{1}{2} \mid -2\}$ is played to $-\frac{1}{2}$. In both cases, it is Left's turn to move. The colored G in the first option has value $1 : 1 = 2$, so that the game as a whole has value $2 - 2 = 0$ and is thus losing for Left. Similarly, the colored G in the second scenario has value $1 : -1 = \frac{1}{2}$, so the total game is losing for the starting Left. □

Alternative proof. Instead of arguing in terms of playing games, we can also prove the proposition by using Lemma 3.2.9 and the arithmetic of ordinal sums. For n even, we find

$$\begin{aligned} G &= \left\{1 : \sum_{k=1}^{n} \pm 1 \mid -1 : \sum_{k=1}^{n} \pm 1\right\} \\ &= \{1 : 0 \mid -1 : 0\} \\ &= \{\{0 \mid\} \mid \{\mid 0\}\} \\ &= \pm 1. \end{aligned}$$

For n odd, we have

$$\begin{aligned} G &= \left\{1 : \sum_{k=1}^{n} \pm 1 \mid -1 : \sum_{k=1}^{n} \pm 1\right\} \\ &= \{1 : \pm 1 \mid -1 : \pm 1\} \\ &= \{\{0, 1 : 1 \mid 1 : -1\} \mid \{-1 : 1 \mid 0, -1 : 1\}\} \\ &= \{\{0, 2 \mid \tfrac{1}{2}\} \mid \{-\tfrac{1}{2} \mid 0, -2\} \\ &= \{2 \mid \tfrac{1}{2} \parallel -\tfrac{1}{2} \mid 2\} \\ &= \pm\{2 \mid \tfrac{1}{2}\}. \end{aligned}$$

□

For more complicated trees, we may use the following recursion to determine the value.

Theorem 3.2.11. *Let G be an uncolored tree. We have $G = \pm(1 : G')$, where G' is G with the edge starting in the root node contracted.*

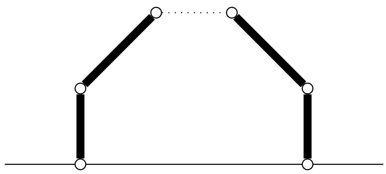
Proof. If the degree of the root of G is $n > 1$, it is straightforward to see that G can be seen as the sum of n trees. Hence, suppose that the root has degree 1. By Lemma 3.2.9, the dominating move for both players is to color the edge originating in the root, from which the statement immediately follows. □

Theorem 3.2.11 provides a linear-time algorithm to determine the value of an arbitrary uncolored tree, albeit not always in canonical form.

3.2.4 Cycles

As for regular Hackenbush, cycles provide more of a challenge. We characterize the value of a single cycle.

Proposition 3.2.12. *Let*

$C_n =$ 

be a cycle with n edges. Then

$$C_n = \begin{cases} 0, & n \text{ even}, \\ \pm 1, & n \text{ odd}. \end{cases}$$

Proof. If n is even, a mirroring strategy yields $C_n = 0$.

Hence, suppose n is odd. We will show that $C_n + \pm 1 = 0$ by constructing an explicit winning strategy for the second player. Throughout, we will represent positions of the game by a string of B's, R's and X's, denoting blue, red and black edges, respectively.

First, we will show that any position of the form

$$G_1 = BBx_1x_2\ldots x_{n-3}R + R \quad \text{or} \quad G_2 = RRx_1x_2\ldots x_{n-3}B + B$$

is a second-player win for Right, where the x_i are coupled such that either $x_i = B$ and $x_{n-2-i} = R$ or vice versa for all $i = 1, \ldots, n-3$. Suppose Left moves first on G_1. If Left removes any of the x_i, Right can respond by mirroring and removing x_{n-2-i}. If Left removes the second B edge, Right responds by removing the last R edge, playing to $B + R = 0$. If Right removes the first B edge, Right plays the same response, leading to $R = -1$, a Right win.

Next, suppose Left plays first on G_2. Again, any move on some x_i can be mirrored in x_{n-2-i}. If Left removes the last B edge in the cycle, Right can respond by removing the second R edge, playing to $R + B = 0$. If Left removes the loose B edge, Right responds by playing on x_i with $|i - \frac{n-3}{2}|$ minimal. If Left mirrors this move in x_{n-2-i}, Right continues making the same response until either Left removes the last B in the cycle, at which point Right wins as before, until Left plays on some other x_j, which Right mirrors in x_{n-2-j}, or until the position $RR + B$ is encountered, which is a win for Right playing first.

Next, we will consider the positions

$$G_3 = RRx_1x_2\dots x_{n-3}R + B, \quad G_4 = RRx_1x_2\dots x_{n-3}B + R,$$
$$G_5 = RBx_1x_2\dots x_{n-3}R + R, \quad G_6 = BRx_1x_2\dots x_{n-3}R + R$$
$$\text{and } G_7 = RRx_1x_2\dots x_{n-3}R + R,$$

where the x_i are again coupled as before, except that now $x_i = x_{n-2-i} = B$ holds for exactly one i. We show that G_3, G_4, G_5, G_6 and G_7 are won by Right playing second.

In G_3 and G_7, regardless of Left's moves, Right may remove the second and last R edge in the cycles on his first two turns, playing to $R + B = 0$. Similarly, in G_5, Right can remove the red ends of the cycle, resulting in R. In G_6, Right removes the first and last R in the cycle, resulting in $B + R = 0$.

Remains to consider G_4. If Left plays on a coupled x_i, then Right mirrors on x_{n-2-i}. If Left removes the last B in the cycle, Right responds by taking the second R, playing to $R + R$, which is certainly a Right win. Next, supppose Left plays on the x_i for which $x_{n-2-i} = B$. If $i < n - 2 - i$, Right responds by playing on x_j with $j > n - 2 - i$ minimal; otherwise, Right responds by playing on x_j with $j < i$ maximal. Now, Right can mirror any move of Left except a move on x_{n-2-j}. If Left makes this move, Right continues replying in the same fashion until $RR + B + R$ is reached with Right moving and winning.

The final step is to prove that the second player can force a game of the form G_1 (if Left goes first) or G_2 (if Right goes first), and this is the best possible. As $G_1 = -G_2 = 0$, we may assume without loss of generality that Left starts. Opening to $BX\dots X + X$, Right responds to $BX\dots XR + X$. If Left plays to $BBX\dots XR + X$ or $BX\dots XR + B$, Right responds to $BBX\dots XR + R$ or $BX\dots XRR + B$, respectively. If Left plays on any other X, Right responds by mirroring, resulting in G_1 or G_2, respectively.

If Left opens to $XBX\dots X + X$, Right responds to $XBX\dots XR + X$. Now, if Left plays to $BBX\dots XR + X$ or $XBX\dots XR + B$, Right responds to $BBX\dots XR + R$ or $XBX\dots XRR + B$, respectively. In the former case, Right can mirror to achieve G_1. In the latter case, Right also mirrors, coupling the first and last X. The result will either be G_2 or G_i for $i = 3, \dots, 7$, won by Right going second in both cases.

If Left opens to $X\dots X + B$, Right responds to $RX\dots X + B$. Next, if Left plays to $RBX\dots X + B$, Right responds to $RBX\dots R + B$ and mirrors afterwards. Right then wins the result playing second, identifying the first R with the loose B, and the second B with the last R. If Left plays to $RX\dots XB + B$, Right responds to $RRX \ldots XB + B$ and continues mirroring, leading to G_2. If Left plays on any other X, Right can respond leading to some G_i, $i = 3, \ldots, 7$.

Finally, if Left opens by coloring any other edge, Right can always respond to create any G_i, $i = 1, \ldots, 7$, winning the game playing second. □

Naturally, further research could focus on determining whether there is some polynomial time algorithm to determine the value of an arbitrary uncolored position, or whether this problem is NP-complete.

Chapter 4

Order versus Chaos

This chapter is based on joint work with Sipke Castelein and Daan van Gent, published at the IEEE Conference on Games 2020 [15], which was in turn an extension of Sipke's bachelor thesis on the game Order and Chaos [30]. Concerning a positional game, it is a slight deviation from the focus on (synchronized versions of) truly combinatorial games in the rest of the thesis, and can therefore be considered as a standalone piece of work.

We study the positional game of Order versus Chaos, which can be considered a maker-breaker variant. The players Order and Chaos take turns placing circles or crosses on a board, in which the goal of Order is to create a consecutive line of identical symbols of a certain length, while Chaos aims to prevent this. In this paper, we provide some theoretical results on winning strategies for both players on finite boards of varying sizes, as well as on infinite boards. The composition of these strategies was aided by the use of Monte-Carlo Tree Search (MCTS) players, as well as a SAT solver. In addition to these theoretical results, we provide some more experimental results obtained using MCTS.

4.1 Introduction

The game of Order versus Chaos is a maker-breaker-like positional game [31]. In the original game 'Order and Chaos', as proposed by Stephen Sniderman in the Games Magazine [16], two players, named Order and Chaos, take turns placing either a circle or a cross on a 6×6 board. Both players are allowed to place either symbol on an empty square. The goal of Order (maker) is to create a horizontal, vertical or diagonal line of (at least) five identical symbols, while the goal of Chaos (breaker) is to prevent this whilst filling the board.

The original version of the game was solved by Benjamin Turner using a brute-force approach [32], showing that Order wins playing first. In this chapter, we will discuss a more sophisticated strategy solving the original game. In order to find this strategy, we constructed two artificial players using Monte-Carlo Tree Search (MCTS) simulations, a popular method for solving combinatorial or positional games [33]. Analyzing the moves prescribed for Order by the MCTS algorithm, we distilled an explicit rule-based strategy.

Moreover, we consider larger games, in which the objective for Order is to make a line of more than five in a row. For winning lines of length at least 9, we model our problem as an instance of the Satisfiability (SAT) problem, for which fast solvers are available [34]. We prove constructively that Chaos always wins if Order needs to align at least 10 symbols, and that Chaos wins if Order needs to align 9 symbols and the amount of squares on the board is of suitable parity.

For games in which Order needs a line of length 6, 7 or 8 to win, we prove that Chaos wins if the board is not much larger than the line to be made. Moreover, we use more MCTS simulations to explore these games, conjecturing that these games are winning for Order if and only if the board is large enough.

We start by introducing some notation. Throughout, for any natural number n, we denote $[n] = \{1, \ldots, n\}$. A *board* is a finite set $B \subseteq \mathbb{Z}^2$ and a *game state* of B is a map $B \to S$, where $S = \{\bigcirc, \times, \square\}$ is the set of *symbols* with $\square$ denoting an *empty square*. For $s = \bigcirc$, we define $\overline{s} = \times$, and vice versa. A *line* is a set of the form $\{(x, y) + k \cdot (a, b) \,|\, 0 \leq k < m\}$ for some $(x, y) \in \mathbb{Z}^2$, $m \in \mathbb{Z}_{>0}$ and non-zero $(a, b) \in \{-1, 0, 1\}^2$, and we call m the length of this line. We call a line L *homogeneous* if either $f[L] = \{\times\}$ or $f[L] = \{\bigcirc\}$, where we write $f[L] = \{f(x) \mid x \in L\}$. The *players* are Order and Chaos.

For a board B, a positive integer m and a player p we define the positional game $\mathrm{ovc}(B, m, p)$ as follows. The players take turns starting with player p and as initial game state f the empty board, i.e., $f(b) = \square$ for all $b \in B$. If the board is full, i.e., $\square \notin f[B]$, the game ends. Otherwise, a turn consists of choosing some $b \in B$ with $f(b) = \square$ and updating f at b such that $f(b) = \bigcirc$ or $f(b) = \times$. In accordance with the terminology for maker-breaker games, We call a line $L \subseteq B$ of length m a *win line*. We say f is in *order* if there exists a homogeneous win line. If f is in order at the end of the game, then Order wins, and otherwise

Chaos wins. The traditional version of Order versus Chaos is thus defined by $\text{ovc}([6]^2, 5, \text{Order})$.

We similarly define the game $\text{ovc}'(B, m, p)$ where the starting player p each turn in addition to his or her usual moves is allowed to pass, i.e., skip their turn. We study this game because it has nice properties with respect to inclusion of boards.

Lemma 4.1.1. *Write $X \preceq Y$ for 'Order wins X implies Order wins Y'. Let $A \subseteq B \subseteq C$ be boards, p a player and $m > 0$. Then*

$$\begin{aligned}\text{ovc}'(A, m, \text{Chaos}) \preceq \text{ovc}'(B, m, \text{Chaos}) &\preceq \text{ovc}(B, m, p)\\ &\preceq \text{ovc}'(B, m, \text{Order}) \preceq \text{ovc}'(C, m, \text{Order}).\end{aligned}$$

The winning result for Order is summarized as follows.

Theorem 4.1.2. *Let B be a board containing $[n]^2$ for some n. Then Order wins $\text{ovc}'(B, m, \text{Chaos})$ for $(m, n) \in \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 6)\}$.*

The result for Chaos for long win lines is as follows.

Definition 4.1.3. For parameters (B, m, p) we say the game $\text{ovc}(B, m, p)$ has *good parity* if $|B|$ is even when p is Order and $|B|$ is odd when p is Chaos.

Theorem 4.1.4. *Let B be a board and let p be a starting player. Then*

(i) *Chaos wins $\text{ovc}(B, m, p)$ for all $m \geq 9$ when the game has good parity.*
(ii) *Chaos wins $\text{ovc}'(B, m, \text{Order})$ for all $m \geq 10$.*

Clearly, good parity can be obtained by passing the first turn, so assuming Theorem 4.1.4 we have that Chaos wins $\text{ovc}'(B, m, \text{Chaos})$ for all boards B and $m \geq 9$.

For small boards, we have the following result for Chaos.

Proposition 4.1.5. *Chaos wins $\text{ovc}'([5 + 2m]^2, 5 + m, \text{Order})$ for all $m \geq 0$.*

We prove Lemma 4.1.1 and discuss some more subtleties considering passing in Section 4.2. In Section 4.4, we constructively prove Theorem 4.1.2, using a strategy inspired by MCTS play. This explicit rule-based strategy leads to a win for Order starting from the empty board, therewith weakly solving the game. Note that this contrasts the solution in [32], where a winning move is listed for every winning position of Order, strongly solving the game.

The proof of Theorem 4.1.4 can be found in Section 4.3, resting on a SAT formulation of the game. In this section, we also prove Proposition 4.1.5. For some of the games not covered by the above theorems, we provide MCTS simulations in Section 4.5. Finally, we discuss some generalizations of these results to infinite boards in Section 4.6.

Throughout, in all strategies discussed, the listed steps are executed in order. When a move is made, after the other player's turn, we start again with the execution of the first step.

4.2 Maker-breaker games and passing

A classic maker-breaker game is defined by a set U and a family of winning sets $\mathcal{F} \subseteq 2^U$. Maker and breaker alternate turns, each turn claiming an unclaimed element from U. Maker wins by claiming all elements in some $F \in \mathcal{F}$, while breaker wins by claiming at least one element from each $F \in \mathcal{F}$.

The Order versus Chaos game $\mathrm{ovc}(B, m, p)$ differs from classic maker-breaker games in that both players do not claim an element from B as their own on their turn, but instead assign an as of yet unassigned element to either $X = \{b \in B \mid f(b) = \times\}$ or $O = \{b \in B \mid f(b) = \bigcirc\}$. The winning sets are the win lines, and Order (maker) wins if either $L \cap X = L$ or $L \cap O = L$ for some win line L, while Chaos (breaker) wins if $L \cap X \neq \emptyset$ and $L \cap O \neq \emptyset$ for all win lines L.

While it is clear that in a classic maker-breaker game, it is disadvantageous to pass for either player, this is not always true in Order versus Chaos.

Example 4.2.1. Consider $\mathrm{ovc}([2] \times [1], 2, \text{Chaos})$. It is clear that Order wins this game. However, the variant $\mathrm{ovc}'([2] \times [1], 2, \text{Chaos})$ is won by Chaos by passing on the first turn. Similarly, the game $\mathrm{ovc}([2] \times [1], 2, \text{Order})$ is won by Chaos, while $\mathrm{ovc}'([2] \times [1], 2, \text{Order})$ is won by Order. ◁

Some relations between the passing variant and the regular game are summarized in the following lemma.

Lemma 4.1.1. *Write $X \preceq Y$ for 'Order wins X implies Order wins Y'. Let $A \subseteq B \subseteq C$ be boards, p a player and $m > 0$. Then*

$$\begin{aligned}\mathrm{ovc}'(A, m, \text{Chaos}) \preceq \mathrm{ovc}'(B, m, \text{Chaos}) &\preceq \mathrm{ovc}(B, m, p) \\ &\preceq \mathrm{ovc}'(B, m, \text{Order}) \preceq \mathrm{ovc}'(C, m, \text{Order}).\end{aligned}$$

Proof. To prove the third inequality, if Order wins $\operatorname{ovc}(B, m, \text{Order})$, it is clear that Order also wins $\operatorname{ovc}'(B, m, \text{Order})$. If Order wins $\operatorname{ovc}(B, m, \text{Chaos})$, then Order wins $\operatorname{ovc}'(B, m, \text{Order})$ by using the same strategy, but passing on the first turn.

For the second inequality, we use a similar argument, but by contrapositive. Hence, suppose that Chaos wins $\operatorname{ovc}(B, m, p)$ for some p. Then Chaos must also win $\operatorname{ovc}'(B, m, \text{Chaos})$ using the same strategy if $p = \text{Chaos}$, and passing on the first turn if $p = \text{Order}$.

For the first inequality, Order can apply her winning strategy for A to B by treating every move by Chaos outside of A as a pass from Chaos on A.

For the last inequality, suppose Order has a winning strategy for the game $\operatorname{ovc}'(B, m, \text{Order})$. Then she can win $\operatorname{ovc}'(C, m, \text{Order})$ by applying this strategy when Chaos moves in B while passing when Chaos moves in $C \setminus B$. □

Example 4.2.1 shows that the implications defined by the second and third inequalities are not equivalences. The next example shows that holds for both other implications, as well.

Example 4.2.2. Consider $\operatorname{ovc}'([1]^2, 2, \text{Chaos})$ and $\operatorname{ovc}'([2]^2, 2, \text{Chaos})$. It is clear that Order cannot win $\operatorname{ovc}'([1]^2, 2, \text{Chaos})$, as there is not enough space for a line of length 2, while it is straightforward to check that Order wins the game $\operatorname{ovc}'([2]^2, 2, \text{Chaos})$. Similarly, Order wins $\operatorname{ovc}'([2]^2, 2, \text{Order})$, but not the game $\operatorname{ovc}'([1]^2, 2, \text{Order})$. ◁

Remark 4.2.3. In ovc', a player never needs to pass on consecutive turns. Suppose a player p passes at turn n, the opposing player makes a move s at b, and then passing is a winning move for p. Instead player p can play s at b turn n, resulting in the same winning game state. Hence the pass was unnecessary.

4.3 Winning strategies for Chaos

We prove Theorem 4.1.4 by explicitly constructing a strategy for Chaos. In the construction, we use a so-called "pairing strategy", the likes of which can be used to solve, e.g., variants of tic-tac-toe, as well as a variant of the original version of Order versus Chaos [31,35].

Throughout, $B \subseteq \mathbb{Z}^2$ will be a board. For a line $L \subseteq \mathbb{Z}^2$ of length $m \geq 2$ there exist two lines L_+ of length $m+1$ such that $L \subseteq L_+$. We choose $L_+ = L \cup \{(x, y)\}$ such that $2y - x$ is maximal among the two possibilities, or equivalently such that (x, y) occurs before the points of L in 'reading order' (left-to-right, top-to-bottom).

Proposition 4.3.1. *There exists a partitioning $\mathcal{P}$ of $\mathbb{Z}^2$ into lines of length 2 such that the following holds:*

(i) *For every line L of length 9 there exists a $P \in \mathcal{P}$ such that $P \subseteq L$.*
(ii) *For every line L of length 10 there exists a $P \in \mathcal{P}$ such that $P_+ \subseteq L$.*

We call $\mathcal{P}$ a 'pairing' and its elements 'pairs'.

Proof. We give a constructive proof.

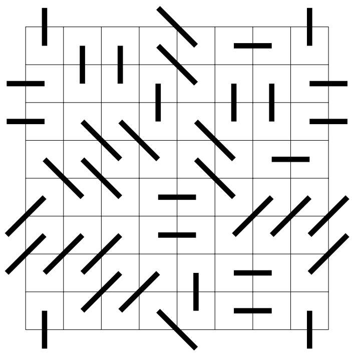


Figure 4.1: Pairing for good parity

Consider Figure 4.1, where each square in the grid represents an element of the flat torus $(\mathbb{Z}/8\mathbb{Z})^2$, and each thick line denotes a pair of two adjacent squares. Observe that every row, column and diagonal of $(\mathbb{Z}/8\mathbb{Z})^2$ contains a pair $\overline{P}$.

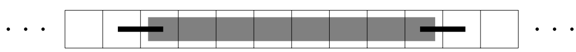


Figure 4.2: From $(\mathbb{Z}/8\mathbb{Z})^2$ to $\mathbb{Z}^2$

Equivalently, Figure 4.1 gives a partitioning $\mathcal{P}$ of $\mathbb{Z}^2$ into lines of length 2 by daisy chaining the pattern. For any line $L \subseteq \mathbb{Z}^2$ of length at least 8 we consider its image $\overline{L}$ in $(\mathbb{Z}/8\mathbb{Z})^2$ and note that this image must contain a pair $\overline{P}$. If L has length exactly 8, it is possible that L intersects two pairs with image $\overline{P}$

non-trivially without containing any, as in Figure 4.2 where L is drawn grey. However, in this case, extending L by one in either direction solves this problem, which proves (i). Proving (ii) goes similarly. □

The pairing given by Figure 4.1 is rather irregular and hard to find. One can easily find pairings for $(\mathbb{Z}/n\mathbb{Z})^2$ for $n > 8$. The reason for this is straightforward combinatorics: there are $4n$ lines that have to contain a pair, which requires $8n$ points, while we have n^2 points available. This also suggests that a pairing for $n = 8$ could just be possible.

To find the pairing in Figure 4.1, we formulated the problem as an instance of the Satisfiability (SAT) problem. An instance of SAT consists of a Boolean expression in conjunctive normal form, and a solution is a true/false assignment of the variables that makes the expression true, or a proof that such an assignment does not exist. While the SAT problem has long been known to be NP-complete, modern-day solvers can still efficiently solve sizeable instances with ease. To find the required pairing, we have used the PicoSAT solver [36].

We introduce a variable $x_{\{p,q\}}$ for each pair of adjacent points $p, q \in (\mathbb{Z}/8\mathbb{Z})^2$. Setting $x_{\{p,q\}}$ to true corresponds to pairing the squares p and q. For every two intersecting pairs $A \neq B$, we add a clause $(\neg x_A) \vee (\neg x_B)$ to guarantee that a square is paired to at most one other square. Now, any line $L \subseteq (\mathbb{Z}/8\mathbb{Z})^2$ of length 8 contains 8 pairs of adjacent points $A_1, \ldots, A_8$, of which at least one pair must be coupled. To do so, we add a clause $x_{A_1} \vee \ldots \vee x_{A_8}$. Any solution to the conjunction of the aforementioned clauses thus corresponds to a pairing as desired, and the pairing in Figure 4.1 is such a solution.

Using this pairing, we now describe a strategy for Chaos.

Strategy 4.3.2. Let $\mathcal{P}$ be a pairing of $\mathbb{Z}^2$ given by Proposition 4.3.1. For $b \in \mathbb{Z}^2$, write $\overline{b}$ for the unique element such that $\{b, \overline{b}\} \in \mathcal{P}$. Let f be the current state, $E = f^{-1}[\{\square\}]$ the set of empty squares and $U = \{b \in B \mid \overline{b} \notin B\}$ the set of unmatched squares.

(i) If, in the previous turn, Order played $s \in S$ at $b \in B$ such that $\overline{b} \in E$, then play $\overline{s}$ at $\overline{b}$.
(ii) If there exists some $b \in E \cap U$, then play anything at b.
(iii) Choose any $b = (x, y) \in E$ such that $2y - x$ is maximal and let $c \in \mathbb{Z}^2$ be such that $\{b, \overline{b}\}_+ = \{b, \overline{b}, c\}$. If $c \in B$, play $\overline{f(c)}$ at b. Otherwise, play anything at b.

Theorem 4.1.4. *Let B be a board and let p be a starting player. Then*

(i) *Chaos wins* $\text{ovc}(B, m, p)$ *for all* $m \geq 9$ *when the game has good parity.*
(ii) *Chaos wins* $\text{ovc}'(B, m, \text{Order})$ *for all* $m \geq 10$.

Proof. We show that Strategy 4.3.2 is well-defined and winning for Chaos. Note that in Strategy 4.3.2, a move at b is only made when $f(b) = \square$, i.e., all moves are legal. In step (iii), note that $f(c) \neq \square$ when $c \in B$, otherwise we would have chosen c instead of b as the square to make our move in. Hence Strategy 4.3.2 is well-defined.

(i) First, we consider $\text{ovc}(B, 9, p)$ with good parity. Note that in this case Chaos will always be last to play in U: if Chaos is the starting player, then $|U|$ is odd and he plays in U his first turn; if Order is the starting player, then $|U|$ is even. Chaos plays in U when Order did, so the last turn $|U \cap E|$ will always be even at the start of Order's turn. Consequently, we never enter step (iii) of Strategy 4.3.2 when the game has good parity. In this case, when the game ends, we have for each $b \in B$ that either $b \in U$ or $\overline{f(b)} = f(\bar{b})$ by step (i) and (ii) of the strategy. Then, by Proposition 4.3.1, every win line contains a pair $\{b, \bar{b}\}$, which we just noted is not homogeneous. Hence the board is not in order and Chaos wins, proving (i).

(ii) Now consider the game $\text{ovc}'(B, 10, \text{Order})$. For $P \in \mathcal{P}$ such that $P_+ \subseteq B$, we consider the first time a player plays at P. If Order is first to play in P, then Chaos follows in step (i), after which P and in particular P_+ becomes non-homogeneous. When Chaos is first to play in P, then this must happen in step (iii), after which P_+ becomes non-homogeneous. Then, at the end of the game, by Proposition 4.3.1, every win line contains a P_+ for some $P \in \mathcal{P}$, none of which are homogeneous. Hence again Chaos wins.

The rest of the statement now follows from Lemma 4.1.1. $\square$

Strategy 4.3.2 shows that the game is winning for Chaos if Order needs to make a long homogeneous line to win. For shorter win line length, Proposition 4.1.5, which we prove next, gives some specific results.

Proposition 4.1.5. *Chaos wins* $\text{ovc}'([5 + 2m]^2, 5 + m, \text{Order})$ *for all* $m \geq 0$.

Proof. We begin by showing that Chaos wins $\text{ovc}([5]^2, 5, \text{Order})$ similarly to Theorem 4.1.4.ii, namely by partitioning the board so that some of the squares are matched. Consider the (partial) pairing $\mathcal{P}$ as displayed in the center 5×5 subboard B of Figure 4.3, where squares with the same number are paired and non-numbered squares remain unpaired. Again, for $b \in \mathbb{Z}^2$, we write $\bar{b}$ for the unique element such that $\{b, \bar{b}\} \in \mathcal{P}$. Note that every line $L \subseteq B$ of length 5 contains a pair. We now give a slight modification of Strategy 4.3.2 for Chaos.

			40	41		41	42			
			30	31		31	32			
			20	21		21	22			
45	35	25	3	1	6	1	0	23	33	43
47	37	27	10	2	5	5	4	24	34	44
			9	2		8	9			
47	37	27	10	11	11	8	4	24	34	44
42	32	22	0	7	6	7	3	20	30	40
			23	26		26	25			
			33	36		36	35			
			43	46		46	45			

Figure 4.3: (Partial) pairing for $[5]^2$ and $[11]^2$

(i) If in the previous turn Order played s at b such that $\{b, \bar{b}\} \in \mathcal{P}$ and $\bar{b}$ is empty, play $\bar{s}$ at $\bar{b}$.
(ii) If there is an empty $b \in B$ such that there is no empty $\bar{b} \in B$, play anything at b.
(iii) If there is a pair $\{b, \bar{b}\} \in \mathcal{P}$ with b empty such that the corresponding win line contains $s \in \{\times, \bigcirc\}$, play $\bar{s}$ at b.

Analogous to the proof of Theorem 4.1.4, this is a winning strategy for Chaos if step (iii) is well-defined, i.e., if we can always find such a pair. We enter step (iii) only when the center is filled, in which case we can play in the pair marked 0, assuming it is not already filled. Note that every pair numbered $n \geq 1$ has in its associated win line a square of the pair numbered $n - 1$. Therefore, inductively, we can always play in the pair marked n with n minimal among the empty pairs. Hence Chaos wins $\mathrm{ovc}'([5]^2, 5, \mathrm{Order})$.

Note that only $m = 1, 2, 3, 4, 5$ remain, as the rest follows from Theorem 4.1.4. We give a proof for $m = 3$ as the rest goes analogously and is left as an exercise for the reader. For the board $[11]^2$ we apply the previous strategy to the center 5×5 squares. Then note that almost all lines of length 8 have 5 squares in the center subboard and are already taken care of. The only extra lines are a few off-diagonals, and the orthogonal lines contained completely in the border, and, if we label the border as in Figure 4.3, all of them contain pairs. When the unmatched squares and the center subboard are completely filled, each of the remaining pairs on the border has a filled square in between them, so we may block the corresponding lines as in step (iii) of the strategy for $[5]^2$. Hence Chaos wins $\mathrm{ovc}'([11]^2, 6, \mathrm{Order})$.

To generalize this to other m one needs to extend or restrict the given border for the 11×11 board in the obvious way. □

A different approach to designing strategies for maker-breaker games is using a potential function, which assigns a value in $[0, 1]$ to every win line [37]. A line containing both symbols is assigned the value 0, a homogeneous line is mapped to 1 and any other win line is assigned a value non-decreasing in the amount of empty squares. If the total potential of all win lines is strictly less than 1 at the start of the game, to show that Chaos wins playing first, it suffices to show that after every pair of moves of Chaos and Order, the total potential has not increased. While this is a straightforward argument for true maker-breaker games, it is hard for Order versus Chaos, as a move by Chaos can increase the potential gained by a subsequent move of Order.

4.4 Winning strategies for Order

We continue by proving Theorem 4.1.2, first taking care of the small cases.

Lemma 4.4.1. *Order wins* $\mathrm{ovc}'([n]^2, n, \mathrm{Chaos})$ *for* $n = 1, 2, 3$.

Proof. For $n = 1, 2$ this is trivial, so consider $n = 3$. If Chaos chooses to move we may assume by rotating the board that he moves anywhere in the lower triangular subboard B colored white in Figure 4.4. If Chaos passes we may move at the square marked with a dot. Then, regardless of whether Chaos passes the next turn we may apply rotations to the board such that precisely one square of B is filled and it is Order's turn.

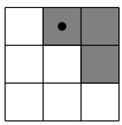


Figure 4.4: 3×3 board

Now Order can force Chaos to keep playing in B by repeatedly forming a line of length 2. It can easily be verified that this always results in a win for Order. $\square$

We continue by assessing $\mathrm{ovc}'([4]^2, 4, \text{Chaos})$. For this game, we constructed a player for both Order and Chaos using Monte-Carlo Tree Search (MCTS) with Upper Confidence Bounds applied to Trees (UCT) as selection method [38,39], further explained in Section 4.5. Pitting these MCTS players against each other, we analyzed the strategy employed by Order in a myriad of games. From this analysis, an explicit rule-based strategy for Order was distilled [30].

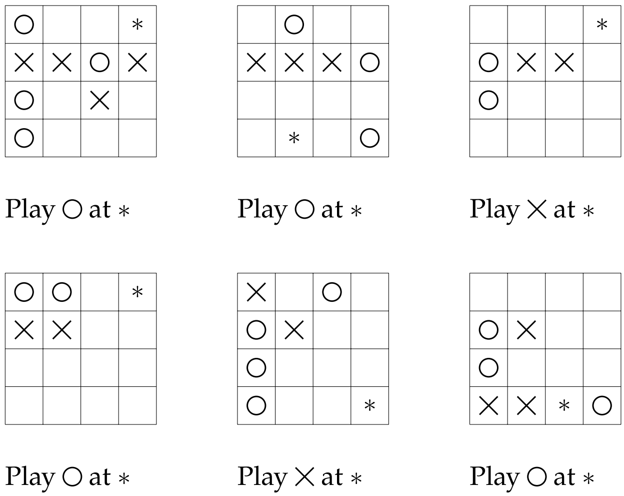


Figure 4.5: Exceptions

Strategy 4.4.2. Let $B = [4]^2$, f the current state and $E = f^{-1}[\{\square\}]$ the set of empty squares. Let $\mathcal{L}$ be the set of win lines, and for $b \in B$, let $\mathcal{L}_b = \{L \mid b \in L \in \mathcal{L}\}$. We call $L \in \mathcal{L}$ *broken* when $\{\bigcirc, \times\} \subseteq f[L]$ and *unbroken* otherwise. For an unbroken line L we define its *weight* $w(L) = |\{b \in L \mid f(b) \neq \square\}|$.

Whenever there is a choice between playing in different $b \in B$, we pick the lexicographically minimal, where the northwest square is numbered $(1, 1)$. In addition to this, there are six exceptional boards. These boards and the corresponding winning moves for Order are in Figure 4.5.

(i) If the board is in Figure 4.5, play the defined move.
(ii) If $f[B] = \{\square\}$, play $\times$ at $(2,2)$.
(iii) If there exists an unbroken $L \in \mathcal{L}$ with $w(L) = 3$, we win by playing in L.
(iv) If there exist a $b \in E$ and distinct unbroken $L_1, L_2 \in \mathcal{L}_b$ of weight 2 such that $f[L_1] = f[L_2] = \{\square, s\}$ for some $s \in \{\bigcirc, \times\}$, play s at b.
(v) Let $\mathcal{E}_s = \{b \in E \mid (\forall L \in \mathcal{L}_b)\ \overline{s} \in f[L] \Rightarrow s \in f[L]\}$ be the set of squares in which playing s does not break a line. If $\mathcal{E} = \mathcal{E}_\times \cup \mathcal{E}_\bigcirc$ is non-empty, let L be a win line intersecting $\mathcal{E}$ of maximal weight. For any $b \in L \cap \mathcal{E}_s$ for some $s \in \{\bigcirc, \times\}$, play s at b.
(vi) Let $\mathcal{L}_s$ be the set of win lines containing $s \in \{\bigcirc, \times\}$ and let $\mathcal{L}_s^*$ be the set of lines in $\mathcal{L}_s$ of maximal weight w^* among all unbroken lines. For any square b, let $w_s(b) = \max\{w(L) \mid b \in L \in \mathcal{L}_s\}$ be the weight of the longest unbroken line through b. We play s in b as to maximize $w_s(b)$ under the constraint that there is no $L \in \mathcal{L}_{\overline{s}}^*$ for which $b \in L$, so that we do not break any line of weight w^*.

For illustration, we draw what will happen when we encounter exception 4. The number above the symbol is the turn in which the symbol was played, and the letter below identifies the player that makes this move. Note that, to verify that the strategy is winning for Order, we need to check all possible moves of Chaos. Here, we show only two, as means of example.

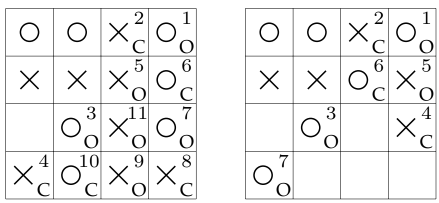


Figure 4.6: Working out two possible outcomes of Exception 4

Lemma 4.4.3. *Order wins* $\mathrm{ovc}'([4]^2, 4, \mathrm{Chaos})$.

Proof. We verify that Strategy 4.4.2 is winning for Order by straightforward computer proof: we check that the strategy is weakly winning against a brute-force player for Chaos. See the Appendix for a reference to the source code. □

To complete the proof of Theorem 4.1.2, we provide the following lemma, based on the result in [32], solving the final case $\mathrm{ovc}'([6]^2, 5, \mathrm{Chaos})$.

Lemma 4.4.4. *Let* $n \geq 1$. *If Order wins* $\mathrm{ovc}'([n]^2, n, \mathrm{Chaos})$, *then Order also wins* $\mathrm{ovc}'([n+2]^2, n+1, \mathrm{Chaos})$.

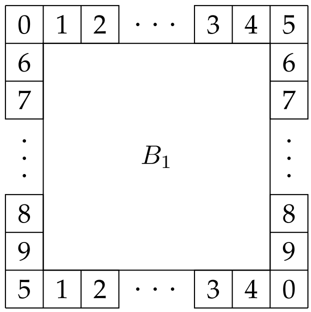


Figure 4.7: Mirroring strategy on B

Proof. We partition $B = [n+2]^2$ into its $n \times n$ center B_1 and its border $B_2 = B \setminus B_1$. As in Figure 4.7 we consider the pairing $\mathcal{P}$ of B_2 that pairs opposing squares, i.e., $\{u, v\} \in \mathcal{P}$ if and only if $u, v \in B_2$ are distinct and $\{u, v\} \subseteq L$ for some line $L \subseteq B$ of length $n+2$ intersecting B_1. We consider the following strategy for Order:

(i) If we can win by completing a win line, do so.
(ii) If Chaos plays in B_2, play the opposing symbol in the paired square.
(iii) Apply the winning strategy for $\mathrm{ovc}'(B_1, n, \mathrm{Chaos})$ to B_1.

We show that the strategy is well-defined and winning. Note that we only play in B_2 in response to Chaos in step (ii) or when we win in step (i). Hence if Chaos plays in B_2, then B_2 will always contain an odd number of filled squares and since $|B_2| = 4(n+1)$ is even there must be an empty square left. Thus step (ii) is well-defined. If B_1 is filled at the start of our turn, then due to the strategy applied it contains a homogeneous line L of length n. Then there exists a pair $P \in \mathcal{P}$ such that $L \cup P$ is a line of length $n+2$. We either have that P is empty, in which case we can in fact win in step (i), or both squares are filled with opposing symbols due to step (ii), in which case a homogeneous line of length $n+1$ already exists. Hence step (iii) is well-defined, and since B_1 will be filled at some point in the game, the strategy is also winning. $\square$

Theorem 4.1.2. *Let* B *be a board containing* $[n]^2$ *for some* n. *Then Order wins* $\mathrm{ovc}'(B, m, \mathrm{Chaos})$ *for* $(m, n) \in \{(1,1), (2,2), (3,3), (4,4), (5,6)\}$.

Proof. Combine the statements of Lemma 4.4.1, that of Lemma 4.4.4 applied to Lemma 4.4.3, and Lemma 4.1.1. □

4.5 Monte Carlo results

Monte-Carlo Tree Search (MCTS) is an algorithm that, for every move, iteratively builds a game tree in search of good states, starting with the current state as root [39]. The search is done in four steps per iteration.

(i) *Selection.* In the current state s, we select a state t reachable by one move which has not been visited yet. If such a state does not exist, we pick a reachable state t for which

$$v_t + C\sqrt{\frac{\ln n_s}{n_t}}$$

is maximized, where n_s is the amount of times state s has been visited (analogous for n_t), v_t is the percentage of visits that eventually led to a win for the current player, and C is a chosen constant. We continue selecting until we reach an as-of-yet unvisited state or a terminal state in which either player has won.

(ii) *Play-out.* The game is finished by making random moves until a player has won.

(iii) *Expansion.* The new state which was encountered for the first time is added to the tree.

(iv) *Backpropagation.* All the states that have been visited in the current iteration are updated to incorporate the results of the played-out game.

After a set amount of iterations, cq. play-outs, the current player performs the move leading to the state t with the highest v_t, and the search is continued.

To find Strategy 4.4.2, the algorithm was run with $C = \sqrt{2}$ and $5{,}000$ play-outs per move. After running the algorithm numerous times, we discovered a pattern in Order's moves, which was used to synthesize the steps of the strategy. While the first two rules of Strategy 4.4.2 are straightforward, the latter ones are more involved. By testing the algorithm without the exceptional step (i), the game was weakly solved except for in the six cases shown in Figure 4.5. Adding these boards as exceptions, the strategy weakly wins the game for Order.

For the game of $\mathrm{ovc}([n]^2, m, p)$, we have exhaustive theoretical results for $m \leq 5$ in Theorem 4.1.2 and for $m \geq 10$ in Theorem 4.1.4. However, for the values of

m in between, our only provable result is for small boards in Proposition 4.1.5, and for boards of good parity in Theorem 4.1.4. For these values of m, we conducted additional MCTS experiments.

# itr.	100,000									60,000	
$m \diagdown n$	3	4	5	6	7	8	9	10	11	12	13
3	100	100	100	-	-	-	-	-	-	-	-
4	-	100	100	100	-	-	-	-	-	-	-
5	-	-	0	91	99	100	100	-	-	-	-
6	-	-	-	0	0	22	50	91	99	100	-
7	-	-	-	-	0	0	0	0	1	23	64
8	-	-	-	-	-	-	-	-	0	0	1

Table 4.1: Win percentage of Order in MCTS simulation of one hundred $\mathrm{ovc}([n]^2, m, \mathrm{Order})$ games.

# itr.	100,000									60,000	
$m \diagdown n$	3	4	5	6	7	8	9	10	11	12	13
3	100	100	100	-	-	-	-	-	-	-	-
4	-	100	100	100	-	-	-	-	-	-	-
5	-	-	0	95	100	100	100	-	-	-	-
6	-	-	-	0	0	18	55	90	100	100	-
7	-	-	-	-	0	0	0	0	2	27	61
8	-	-	-	-	-	-	-	-	0	0	0

Table 4.2: Win percentage of Order in MCTS simulation of one hundred $\mathrm{ovc}([n]^2, m, \mathrm{Chaos})$ games.

# itr.	100,000									60,000	
$m \diagdown n$	3	4	5	6	7	8	9	10	11	12	13
3	100	100	100	-	-	-	-	-	-	-	-
4	-	100	100	100	-	-	-	-	-	-	-
5	-	-	0	86	100	100	100	-	-	-	-
6	-	-	-	0	0	15	61	92	99	100	-
7	-	-	-	-	0	0	0	0	1	14	50
8	-	-	-	-	-	-	-	-	0	0	7

Table 4.3: Win percentage of Order in MCTS simulation of one hundred $\mathrm{ovc}'([n]^2, m, \mathrm{Order})$ games.

# itr.	100,000									60,000	
$m \diagdown n$	3	4	5	6	7	8	9	10	11	12	13
3	100	100	100	-	-	-	-	-	-	-	-
4	-	100	100	100	-	-	-	-	-	-	-
5	-	-	0	95	100	100	100	-	-	-	-
6	-	-	-	0	0	22	52	92	98	100	-
7	-	-	-	-	0	0	0	0	1	33	67
8	-	-	-	-	-	-	-	-	0	0	0

Table 4.4: Win percentage of Order in MCTS simulation of one hundred $\mathrm{ovc}'([n]^2, m, \text{Chaos})$ games.

The results of the experiments can be found in Table 4.1 and Table 4.2. The algorithm was run with $C = \sqrt{2}$ for all values of m and n. The amount of play-outs allowed per move is dependent on n and shown in the first row of the tables. Each game was simulated 100 times; the tables show the amount of games won by Order.

Although we must note that it is unknown whether MCTS inherently favors one of the asymmetric players, we formulate some conjectures looking at the results. For $m = 6$, we know that the game on the board $[n]^2$ is winning for Chaos for $n \leq 7$, which is also the result found by MCTS. For $n = 8$, the game also seems to be winning for Chaos. For $n = 9$, the simulation results are unclear. For $n \geq 10$, the game appears to be winning for Order.

For $m = 7, 8$, MCTS again shows an increase in win rate for Order when the board becomes larger, leading to the conjecture that $\mathrm{ovc}([n]^2, m, p)$ is winning for Order if and only if $n \geq N_m$ for some fixed N_m. However, the results do not show a sharp threshold with the current amount of iterations, and thus do not give an indication for the values of N_m.

In Section 4.2, we remarked that, theoretically, passing may be a beneficial move for either player. To investigate the impact of allowing a player to pass, we ran MCTS simulations for the game ovc', the results of which can be found in Table 4.3 and Table 4.4.

Note that the results are very similar to those in Table 4.1. Hence, it appears that allowing the starting player to pass does not have a large impact; investigating the moves made by the MCTS players, we find that passing is rarely done. It thus seems that, in the games we consider here, passing is not advantageous to either player. Finally, note that for these games, also the starting player does not seem to have a noticeable effect.

4.6 Infinite boards

We can generalize $\mathrm{ovc}(B, m, p)$ and $\mathrm{ovc}'(B, m, p)$ to non-finite boards B. Here, Order wins as usual when there is a homogeneous line of length m. As there is no way to fill the board, Chaos cannot win in the traditional sense. Hence, we define $\mathrm{ovc}(B, m, p)$ for infinite B to be won by Chaos if no winning strategy for Order exists.

First note that Lemma 4.1.1 generalizes to infinite boards almost perfectly. The only exception is $\mathrm{ovc}'(B, m, \text{Order}) \preceq \mathrm{ovc}'(C, m, \text{Order})$, which only holds when $C \setminus B$ is finite, as Order wants to ensure they do not keep playing in $C \setminus B$ forever when Order plans to win on B. Now Theorem 4.1.2 is applicable and Order wins $\mathrm{ovc}'(B, m, \text{Chaos})$ for small m when B contains a sufficiently large square subboard. For $m = 2$ we can do slightly better.

We say $a, b \in B$ are *neighbours* if $\{a, b\}$ is a line of length 2. The *connected* relation is then the transitive closure of the neighbour relation.

Lemma 4.6.1. *Let B be a possibly infinite board. Then Order wins $\mathrm{ovc}'(B, 2, \text{Chaos})$ if and only if B contains a connected component of size at least 3.*

Proof. Note that a connected component of size at least 3 contains a pair of distinct intersecting win lines L_1 and L_2. Since Chaos can never play in an empty win line, Order can play in $L_1 \setminus L_2$ and win in her next turn. If all connected components of B are of size at most 2, then all win lines are disjoint. Chaos can simply pass until Order moves in a win line and counter. □

For Chaos, we extend Theorem 4.1.4.

Definition 4.6.2. We say $\mathrm{ovc}(B, m, p)$ has *good parity*

(i) in case B is finite, when $p = \text{Chaos}$ if and only if $|B|$ is odd.
(ii) in case B is co-finite, i.e., $\mathbb{Z}^2 \setminus B$ is finite, when $p = \text{Chaos}$ if and only if $|\mathbb{Z}^2 \setminus B|$ is odd.
(iii) in case B and $\mathbb{Z}^2 \setminus B$ are infinite, always.

Lemma 4.6.3. *If $B \subseteq \mathbb{Z}^2$ is neither finite nor co-finite, then there are infinitely many lines $L \subseteq \mathbb{Z}^2$ of length 2 for which $|L \cap B| = 1$.*

Proof. Suppose there are only finitely many lines $L \subseteq \mathbb{Z}^2$ of length 2 for which $|L \cap B| = 1$ and let E be their union. Then, up to a translation of B, there exists some $n \geq 0$ such that $E \subseteq [n]^2$. Since B is not finite, $B \setminus [n]^2$ is non-empty.

For each $b \in B \setminus [n]^2$ all its neighbours a are in $B \setminus [n]^2$, otherwise $\{a, b\} \in E$. Since $\mathbb{Z}^2 \setminus [n]^2$ is connected we have $\mathbb{Z}^2 \setminus [n]^2 \subseteq B$, so B is co-finite. The lemma follows from contradiction. □

Theorem 4.6.4. *Let B be a possibly infinite board and let p be a starting player. Then*

(i) *Chaos wins* $\text{ovc}(B, m, p)$ *for all* $m \geq 9$ *when the game has good parity.*
(ii) *Chaos wins* $\text{ovc}'(B, m, \text{Order})$ *for all* $m \geq 10$.

Proof. (i) When B is finite this is Theorem 4.1.4.i. When B is co-finite, Strategy 4.3.2 is still applicable: there are only finitely many unmatched squares, and because of the parity, Chaos is never first to play in a pair. We never enter step (iii), so as in the proof of Theorem 4.1.4, no homogeneous line of line of length 9 can ever exist.

Now consider the case where B is neither finite nor co-finite. Partition the lines of length 2 in $\mathbb{Z}^2$ by their image in $(\mathbb{Z}/8\mathbb{Z})^2$. At least one of these partitions $\mathcal{L}$ must contain infinitely many $L \in \mathcal{L}$ such that $|L \cap B| = 1$ by Lemma 4.6.3. After translating B we may assume $\mathcal{L}$ is contained in the pairing $\mathcal{P}$ given by Proposition 4.3.1. Then, applying Strategy 4.3.2 we have infinitely many unmatched squares, so we never enter step (iii) of the strategy, and again no homogeneous line of length 9 can occur. Thus Chaos wins $\text{ovc}(B, m, p)$.

(ii) Again, the finite case was already shown. The non-finite, non-co-finite case, proceeds the same as the proof of (i), since there is an infinite number of unmatched squares. For the co-finite case, we can find a pairing of $(\mathbb{Z}/9\mathbb{Z})^2$ as in Proposition 4.3.1 with ample unmatched squares to show that Chaos wins $\text{ovc}'(\mathbb{Z}^2, 10, \text{Order})$ and thus $\text{ovc}'(B, 10, \text{Order})$. □

4.7 Conclusions and future research

In this chapter, we proved that the game of Order versus Chaos is winning for Order if her objective is to make a line of length at most 5, and the board is suitably large. For the proof, we constructed explicit strategies, fuelled by MCTS simulations. For Chaos, we proved that he wins the game if Order needs to make a line of length at least 10, using a SAT solver to find a winning strategy. For winning lines of length between 6 and 9, we showed that Chaos wins if the board is of suitable size or parity. Furthermore, we generalized some of the theoretical results to infinite boards.

For boards which do not meet these requirements, we ran MCTS simulations to develop conjectures. The results of these simulations suggest that Order wins the game $\mathrm{ovc}([n]^2, m, \text{Order})$ if and only if n is sufficiently larger than m. Note that for general boards B and C with $B \subseteq C$, it is not necessarily the case that $\mathrm{ovc}(B, m, p)$ being won by Order implies that $\mathrm{ovc}(C, m, p)$ is won by Order. However, from the MCTS simulations, one might conjecture that this statement does hold for the games $\mathrm{ovc}([n]^2, m, \text{Order})$ with $m = 6, 7, 8$. It would be interesting to see whether this could be proven.

Besides drawing conclusions from the generated MCTS results, it might be fruitful to explore other AI techniques, such as deep learning, in order to derive more information. For the games $\mathrm{ovc}([n]^2, m, \text{Order})$ with $m = 7$ or $m = 8$, for example, based on the current results, nothing can be said on the threshold for n (if this exists) at which the game becomes winning for Order. Moreover, different techniques may show different strategic behaviour, of which the analysis may lead to new theoretical insights.

Finally, the question of whether passing is advantageous for either player is an interesting one to further investigate, it being a crucial difference between Order versus Chaos and classic maker-breaker games. In Section 4.2, we discussed that passing once can theoretically be an advantage for both players to solve parity problems, while passing twice in a row is never necessary to win. It is unknown whether passing more often offers an advantage to either player. However, the simulations in Section 4.5, showing that Tables 4.1 through 4.4 are roughly the same, strongly suggest that passing is not beneficial except for in edge cases like Example 4.2.1.

Acknowledgment

The authors would like to thank Wessel van Woerden, Walter Kosters and Flora Spieksma for their useful comments, as well as Ludo Pulles and Pim Spelier for their help in constructing the SAT instance for Proposition 4.3.1.

Appendix

Some of our proofs rely on intensive computation. We made our source code available through GitHub at www.github.com/MadPidgeon/Order-versus-Chaos.

We briefly describe the files:

`dump.cc` generates files `order.txt` and `chaos.txt` consisting of zeros and ones indicating for each of the 3^{16} game states of $\mathrm{ovc}'([4]^2, 4, \text{Chaos})$ whether it is winning for the respective player. Both of these files are of size 43 MB and can be used as input for programs to quickly verify whether a strategy is winning for a player.

`verify.py` implements an Order player using Strategy 4.4.2, which is verified against all possible moves of Chaos to prove Lemma 4.4.3.

`sat/` contains the code generating the pairing for Proposition 4.3.1, written by Ludo Pulles and Pim Spelier.

`table/` contains the code generating the computational MCTS results for Section 4.5.

Chapter 5

Nim variants

This chapter is based on work published in [17]. We introduce and analyze three imperfect information variants of the game of Nim. In these variants, the opponent only receives partial information on the move executed by the opponent. We model the variants as games in extensive form as introduced in Section 2.2 and compute Nash equilibria for different starting configurations. For one variant, this provides a full characterization of the game. For the other variants, we prove some partial and structural results, but a full characterization remains elusive.

5.1 Introduction

In this chapter, we introduce three variants of a non-perfect information version of the impartial game Nim. As a first example, consider a simple Nim configuration, depicted in Fig. 5.1. In regular Nim, the winning strategy for the first player is to remove one chip from any heap.

Figure 5.1: A simple Nim position with three heaps of sizes 3, 3 and 1, respectively.

The first variant we consider is *Schrödinger Nim*. The difference with regular Nim is that the opponent is only told from which heap chips have been removed, but not how many. Hence, a move now consists of first selecting a heap to inspect, and then removing any amount of chips from it. If a heap is emptied by a move, this is communicated to the opponent.

Because of the introduction of imperfect information, a strategy for a player now consists of a probability distribution over their possible moves. We assign value 1 to a position which is won by the first player with probability 1, and -1 if the game is won by the second player. The position in Fig. 5.1 has value 1/3; a pair of optimal strategies is illustrated in Fig. 5.2. Here, "optimal" refers to a Nash equilibrium: we will show that the players cannot improve by one-sided deviation from their strategy.

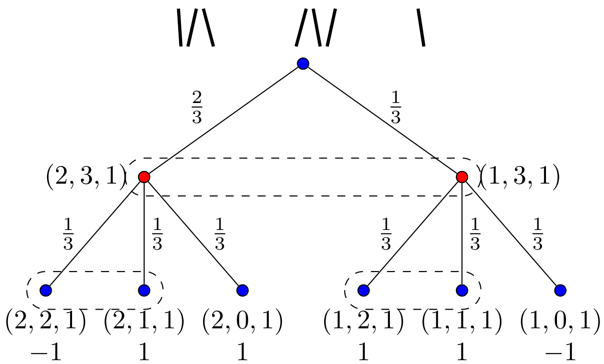


Figure 5.2: Optimal strategies for Schrödinger Nim played on heaps of sizes 3, 3 and 1. The states in the dotted ovals cannot be distinguished by the current player. In the leaves, final values are depicted, omitting further play.

The first player either removes 1 or 2 chips from the first heap, with probability 2/3 and 1/3, respectively. The second player, not knowing how many chips are still in the first heap, removes any number of chips from the second heap with equal probability. If the second player emptied the second heap, the game values are easily computed. Otherwise, if the first player removed one chip from the first heap on the first move, she later proceeds by emptying this heap, winning if and only if the second heap contains 1 chip. Finally, if the first player removed two chips on the first turn, she later proceeds to empty the second heap, winning the game.

We show that, if all heaps consist of at most two chips, Schrödinger Nim positions are equivalent to regular Nim positions. Conversely, if all heaps consist of at least three chips, Schrödinger Nim positions have value 0, i.e., they do not favor either player (see Theorem 5.3.1). The truly interesting positions are those in which there are some heaps of size at most two, and some of size at least three, as in the above example. We consider such positions with three heaps, and provide several results.

The second variant we consider is *Fuzzy Schrödinger Nim*, following the same rules as Schrödinger Nim, except that emptying a heap is no longer signalled to the opponent. In Theorem 5.4.1 and Theorem 5.4.2 we give a complete characterization of this game, as well as its misère variant. Finally, we consider a third version, called *Kriegspiel Nim*, for which we use a set of rules inspired by the chess variant Kriegspiel. Theorem 5.5.1 characterizes games with two heaps.

For both combinatorial games and games without perfect information, research has been done in the field of artificial intelligence to create powerful agents. Examples include the application of Monte-Carlo Tree Search (MCTS) and deep neural networks to Go in [33], using MCTS and meta position based agents to play Kriegspiel in [10, 11], and exploring Counterfactual Regret Minimization and deep neural networks in the context of Heads-up No-limit Poker in [40] and [41]. Whereas these methods produce powerful artificial players also in the context of non-perfect information, we are interested in developing a theory akin to that for combinatorial games, for this class of games.

Research in this direction has been done in the context of synchronizing combinatorial games in [21] and [13]. In these synchronized versions, both players communicate an intended move to an umpire, after which these moves are executed simultaneously. Though results are promising for several partisan games, this approach might prove problematic for impartial games, as both players may have selected the same move to execute.

The (Fuzzy) Schrödinger imperfect information variant of the rules of Nim can be generalized to any (impartial) game in which the amount of disjunctive components is non-increasing. Variants of Nim such as arbitrary finite subtraction games come to mind, but also a game like Push (see [2]) can be altered in a natural way such that the opponent only hears the component on which a move was made, but not the move itself. The Kriegspiel rule variant can be applied to any combinatorial game.

In Section 5.2, we more formally outline the rules of the Nim variants, as well as their modelling as games in extensive form for computational purposes, as outlined in [12,20,25,42]. In Section 5.3, Section 5.4 and Section 5.5, we analyse the three variants, finding a full characterization for Fuzzy Schrödinger Nim, and partial results for the other two. We conclude in Section 5.6.

5.2 Game rules and notation

In this section, we introduce the rules and notation for three variants of Nim. Furthermore, we discuss notation for games in extensive form, which will be used for the analysis of the variants.

5.2.1 Nim and its variants

The classic combinatorial game of Nim starts with a configuration of d heaps, the i-th heap consisting of n_i chips. Two players L and R alternate turns, each player being allowed to take any number of chips from any one heap every turn. The player taking the last chip wins. Nim being an impartial game, for the remainder of the chapter, we assume without loss of generality that L starts. Denoting the height of the heaps by $n = (n_1, \ldots, n_d)$, we write $\textsc{Nim}(n)$ for the game of Nim with these heaps as starting configuration.

In the first variant of the game, dubbed *Schrödinger Nim*, both players know the starting configuration. During the game, however, when the opponent moves, a player is only told which heap the move was on, not the amount of chips removed. An exception is when a heap is emptied, which is always communicated to the opponent. The height of (some of) the heaps may thus be unknown to a player during the game. A move now consists of first selecting a heap, at which point the height of the heap becomes known, followed by making a decision on how many chips to remove from the heap. We denote a game of Schrödinger Nim with starting configuration $n = (n_1, \ldots, n_d)$ by SN(n). Note that the game is impartial and short.

The second variant, which we will call *Fuzzy Schrödinger Nim*, is very much like the first, again with both players knowing the starting configuration. However, in the fuzzy version, emptying a heap is not signalled to the other player. Now, a player may happen to select an empty heap to move on. In this case, the player is told the heap is empty, and they must pick another heap to try and remove chips from. A game of this variant with starting configuration $n = (n_1, \ldots, n_d)$ is denoted by FSN(n). We will return to this variant in Section 5.4.

The third variant which we will consider is inspired by Kriegspiel and is therefore named *Kriegspiel Nim*. Yet again, both players know the starting configuration. During a player's turn, a move consists of trying to remove i chips from the j-th heap. If this is possible, it is done, and the turn is passed, the other player not being informed of anything except the fact that a successful move has been executed. In particular, if a pile is emptied by a move, this is not explicitly communicated to either of the players. If the requested move is not possible, the player must try another move, continuing until a legal move is tried and thus performed. For $n = (n_1, \ldots, n_d)$, we denote this variant of the game with starting position n by KN(n).

To illustrate the three game variants, again consider the Nim position from Fig. 5.1. Suppose that the first player removes one chip from the first heap. In both Schrödinger and Fuzzy Schrödinger Nim, the second player only knows that the first heap has been altered, whereas in Schrödinger Nim he also knows that the heap is non-empty. The possible moves for the second player are now either to remove any number of coins from the second or third heap, or to choose to inspect the first heap. If he chooses to inspect the first heap, he discovers there are two chips left, and may then proceed to choose to remove one or both of them.

In Kriegspiel Nim, after the first player has removed a chip from the first heap, the second player receives no information at all. His move now consists of trying to remove any amount of chips from any heap. If he would try to remove three chips from the first heap, the umpire would respond negatively, revealing that the first player has removed at least one chip from the first heap. Any other attempted move that was legal from the (known) starting position can and will be executed. See Fig. 5.3 for an illustration of this example.

5.2.2 Extensive form games

As the players no longer have perfect information in the variants of Nim described above, the variants are no longer combinatorial games. Instead, we model them as games in extensive form as in [12, 20, 25], already expanded upon in Section 2.2. In such a game, we have a set of states V. The set $V_0 \subseteq V$ is the set of terminal states, each having value $f(v) \in \mathbb{R}$, $v \in V_0$, and the non-terminal states are partitioned into (disjoint) information sets. For every information set S, we define an action set $A = A(S)$. We write $N^+(v)$ for all

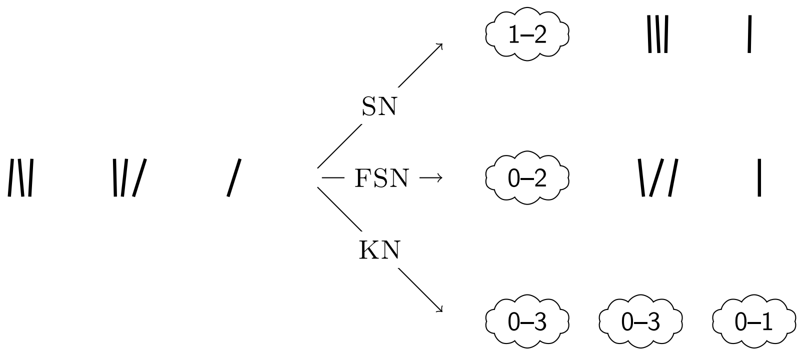


Figure 5.3: The starting position of a game with three heaps of sizes 3, 3 and 1, respectively, as seen by R after L's first move, in the three different variants of Nim. The clouds contain the possible heights of the heaps. Notice that in Kriegspiel Nim, R knows a little more: exactly one of the heaps has been touched.

states reachable by one action from v. To every state $w \in N^+(v)$, we assign the unique action $\alpha_v(w)$ used to reach it, and we denote by w^- the state preceding it. An action sequence is a list of consecutive actions, starting from the initial state: $\sigma = [a_1, \ldots, a_n]$. The part of σ consisting of only L's actions is denoted by σ_L; the definition of σ_R is analogous.

An assumption that is commonly made is that of *perfect recall*: no two different action sequences by one player can lead to the same information set. Under this assumption, we can denote a player's unique action sequence leading to a node v in some information set by $\sigma_P(v)$. Note that by recording the sequence leading to an information set in the description of the set, i.e., by assuming that both players do not forget earlier moves they did, we can always guarantee perfect recall.

A (*behavior*) *strategy* μ defines on every action set a probability distribution. The probability of choosing any $a \in A(S)$ for $S \in \mathcal{S}$ is denoted by $\mu(a)$. The value of a strategy pair $\mu = (\mu_L, \mu_R)$ for the two players L and R is denoted by $v(\mu_L, \mu_R)$. The value of a strategy pair when starting the game in a state $w \in V$ is denoted by $v_w(\mu)$.

For the Nim variants, we take as states the current height of the heaps, and the current player to move. The terminal states are those with only empty heaps, having value 1 if it is R's turn (so L, the starting player, wins) and -1 otherwise, giving rise to a zero-sum game. For the division of the rest of the states into information sets, we keep track of the current player, the action sequence for that player so far, and the knowledge of the player of the height of the heaps. For every heap k, this can either be an exact height, say h, or an upper bound, denoted by $\overline{h}$, if the other player has moved on the heap last.

For the games of (Fuzzy) Schrödinger Nim, the current player can choose to either remove any amount of chips from a known heap, or to take a look at an unknown heap. In the latter case, the player sees the amount of chips left, entering another information state, from which they can only choose to remove any amount of chips from the chosen heap. We denote the move of removing i chips from the (known) k-th heap by $\mathbf{k}_i$.

5.3 Schrödinger Nim

In this section, we analyze the game of Schrödinger Nim. We first argue that configurations in which either all heaps are sufficiently small or all heaps are sufficiently large, the outcome is straightforward. Then, we continue by partially analyzing the game on three heaps of varying size, which proves to be more complicated.

5.3.1 Basic cases

First, note that if the other player has moved on a heap i times since the first player has observed its height to be h, their knowledge of the heap will be $\overline{h-i}$. Furthermore, we have that $\overline{1}$ is equivalent to 1, as the emptying of a heap is signalled. Therefore, if a heap contains two chips, and both players know, the height of that heap will be known to both players for the rest of the game. Hence, SN(n) is equivalent to Nim(n) if $n_i \leq 2$ for all $i = 1, \ldots, d$. Naturally, in the trivial case that the game starts with only one heap, Schrödinger Nim is also equivalent to regular Nim. Finally, if all remaining heaps have height 1 at some point in the game, regardless of the information both players have, the result will be the same as for regular Nim, and is determined by the parity of the number of non-empty heaps.

In these cases, both players can use a deterministic strategy when playing optimally, and the starting player L will either always win or always lose if both players play optimally. In other cases, the game may not be always won or lost by the starting player. Instead, the players will employ a Nash equilibrium of strategies, having a real value $v \in [-1, 1]$, where $\frac{v+1}{2}$ is the probability of L winning. We denote by $v(\text{SN}(n))$ the value of the game, which is the unique value of some Nash equilibrium for the two players.

Theorem 5.3.1. *Let $n \in \mathbb{N}^d_{\geq 3}$, $d \geq 2$. Then $v(SN(n)) = 0$.*

Proof. We proceed by induction on d. For the base case $d = 2$, we provide an explicit strategy pair and prove that it is Nash. On the first turn, L reduces one of the heaps to one or two chips, both with probability $\frac{1}{2}$. Without loss of generality, let this be the first heap. Next, R does the same for the other heap. Now, if L reduced her heap to two chips, she takes one of the remaining chips there. Otherwise, she takes one chip from the other heap. R again mirrors. Note that the value of the strategy pair is indeed 0.

The equilibrium is depicted in Fig. 5.4. The nodes are states, the dotted boxes show information sets. Note that in fact every node is contained in an information set, but only the most relevant sets are displayed. By the discussion above, we may interpret states in which both heaps consist of one chip as terminal states, yielding a value of either 1 or -1.

The description of the strategies above is not exhaustive. Therefore, in the sequel, if we encounter an information set in which no probability distribution over the actions has been defined yet, we must and will do so, making sure that this does not contradict an earlier definition. For example, it has not been decided how Left should play in the information set $[1, n_2]$, which may be reached by Right playing on the first heap during his first turn. However, it is clear that, in this case, $\mathbf{2}_{n_2-1}$ is an optimal move, which we would have filled in only when encountering this information set for the first time.

We prove that the provided strategy pair is indeed Nash. First, suppose R deviates. If R checks and plays on the first heap during the first move, L is given the state $(1, n_2)$ or $(0, n_2)$ with perfect knowledge, winning the game by removing all but one or all chips from the second heap, respectively. Similarly, emptying the second heap leads to a win for L. Therefore, we may suppose that R plays $\mathbf{2}_i$ with probability p_i, with $\sum_{i=1}^{n_2-1} p_i = 1$, $p_{n_2} = 0$.

If R was in $(2, n_2)$, L will respond by moving to $(1, n_2 - i)$, which is a win for R if and only if $n_2 - i \neq 1$. If R was in $(1, n_2)$, L will inspect and play on the second heap, which is a win for R if and only if $n_2 - i = 1$. Both cases occurring with probability $\frac{1}{2}$, we find that R obtains a value of 0 regardless of the p_i. Noting that it is clear that R cannot gain by deviating during his second turn, we conclude that R cannot improve his score by deviating at all.

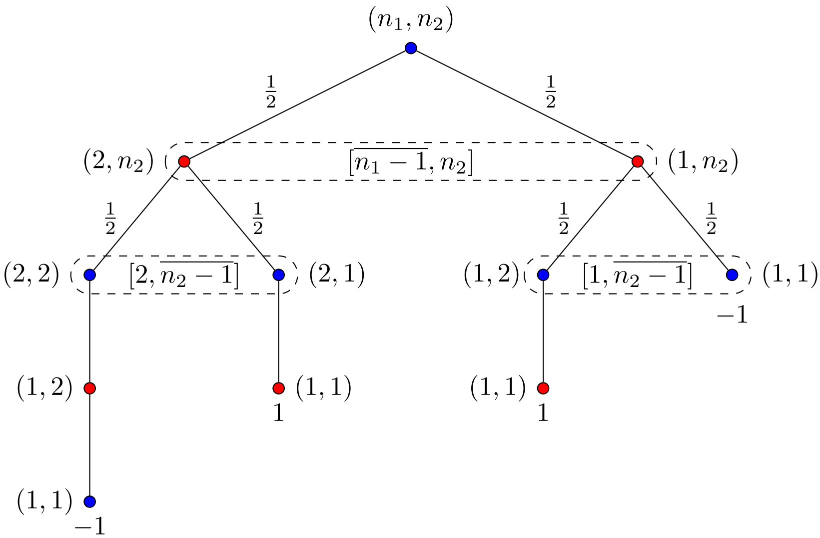


Figure 5.4: The Nash equilibrium in action. In black nodes, it is L's turn; in white nodes, it is R's turn. The action probabilities are shown on the edges. The labels of the information sets display the information as known to the active player.

Next, suppose L deviates. Emptying a heap leads to a win for R, so assume that L plays $\mathbf{1}_i$ with probability p_i, where $\sum_{i=1}^{n_1-1} p_i = 1$, $p_{n_1} = 0$. Next, R moves to either $(n_1 - i, 2)$ or $(n_1 - i, 1)$ with probability $\frac{1}{2}$. If $n_1 - i = 1$, L must play on the second heap to not immediately lose, winning if and only if R left 2 chips. Otherwise, playing on the second heap allows R to certainly win, so suppose L plays $\mathbf{1}_j$ with probability q_j, where $\sum_{j=1}^{n_1-i-1} q_j = 1$, q_{n_1-i}. If R initially left 2 chips, he moves to $(n_1 - i - j, 1)$, which is winning for L if and only if $n_1 - i - j \neq 1$. Otherwise, he inspects and plays on the first heap, which is winning for L if and only if $n_1 - i - j = 1$. Hence, L obtains value 0 regardless of the p_i and q_j and cannot gain by deviating. We conclude that the given strategy pair is Nash.

Finally, suppose $v(\text{SN}(n)) = 0$ for any $n \in \mathbb{N}_{\geq 3}^d$ and consider the game $\text{SN}(n)$ with $n \in \mathbb{N}_{\geq 3}^{d+1}$. If L empties any heap in the first move, this guarantees a value of 0. Otherwise, R is faced with d heaps of known height and one of unknown height. Now, by checking and emptying the heap of unknown height, R can again guarantee a value of 0. By playing on any other heap, he might force a value of less than 0. Therefore, the best L can do is to prevent this by emptying any heap on the first move, obtaining value 0. □

The above theorem shows that either player wins with probability $\frac{1}{2}$ if all heaps contain at least 3 chips at the start of the game. For the game with two heaps, if either heap is initially of height 1 or 2, L can win by reducing the other heap to the same size. Games with three or more heaps, in which some heaps have at most 2 chips and some at least 3, however, turn out to be more difficult to analyze, as we will see in the next subsection.

5.3.2 Three heaps

In this section, we give a partial characterization of $v(\text{SN}(n))$ for $n \in \mathbb{N}^3$. We start by computing the values for all starting configurations except for $(n_1, n_2, 1)$ with $n_1, n_2 \geq 4$, which are summarized in Table 5.1. Note that the heaps may be permuted in any way without changing the value of the game. The game $(1, n_1, 1)$, for example, also has value 1 for all $n_1 \geq 3$.

Theorem 5.3.2. *Let $n \in \mathbb{N}$. Then $v(SN(n, 1, 1)) = v(SN(n, 2, 2)) = 1$.*

Proof. On her first turn, L moves to $(0, 1, 1)$ or $(0, 2, 2)$ respectively, from which point the game is equivalent to two-pile regular Nim. As the piles are of the same size, the current player R loses. □

Table 5.1: Values of three-heap games.

$(n_1, 1, 1)$ $n_1 \geq 1$	$(3, 2, 1)$	$(n_1, 2, 1)$ $n_1 \geq 4$	$(n_1, 3, 1)$ $n_1 \geq 3$
1	0	1/2	1/3
Thm. 5.3.2	Thm. 5.3.3	Thm. 5.3.4	Thm. 5.3.5

$(n_1, 2, 2)$ $n_1 \geq 3$	$(n_1, n_2, 2)$ $n_1, n_2 \geq 3$	(n_1, n_2, n_3) $n_1, n_2, n_3 \geq 3$
1	1/3	0
Thm. 5.3.2	Thm. 5.3.6	Thm. 5.3.1

Theorem 5.3.3. $v(SN(3, 2, 1)) = 0$.

Proof. First, note that if L makes a move on the second or third pile, she loses the game. Indeed, $(3, 1, 1)$ is winning for the active player by Theorem 5.3.2 and the other two cases are trivial. Furthermore, L must not take all chips from the first pile in her first move by Theorem 5.3.1.

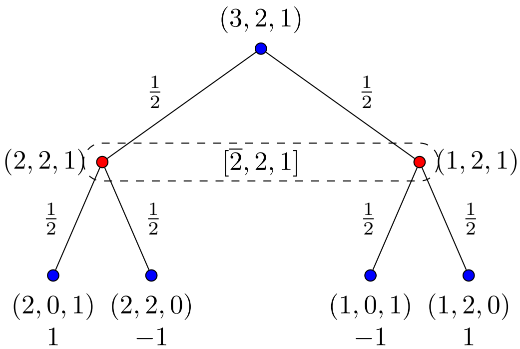


Figure 5.5: A Nash equilibrium for SN$(3, 2, 1)$.

Now, consider the strategy pair depicted in Figure 5.5. Note that the pair has value 0. It is easy to see that L cannot improve by deviating. Indeed, following the reasoning above, only the two actions depicted should be assigned a non-zero probability. Suppose L picks one chip with probability p_1 and two chips with probability p_2, then her payoff becomes

$$p_1(\tfrac{1}{2} - \tfrac{1}{2}) + p_2(-\tfrac{1}{2} + \tfrac{1}{2}) = 0.$$

Next, suppose R deviates. Inspecting and playing on the first pile results in the state (1,2,1) or (0,2,1) with complete information for L, both yielding a certain loss for R. Removing one chip from the second pile gives one of the states (2,1,1) or (1,1,1), again with complete information for L, resulting in a loss for R. Hence, the two actions depicted for R are the only ones potentially worthwhile. Picking from the second pile with probability q_1 and the third with probability q_2, we find a value of

$$\tfrac{1}{2}(q_1 - q_2) + \tfrac{1}{2}(-q_1 + q_2) = 0.$$

Hence, neither player can improve by deviating from the shown strategy, so the strategy pair is indeed Nash with value 0. □

Theorem 5.3.4. *Let $n \in \mathbb{N}$, $n \geq 4$. Then $v(SN(n,2,1)) = \frac{1}{2}$.*

Proof. Again, note that if L makes a move on the second or third pile, she loses the game. Indeed, $(n,1,1)$ is winning for the active player by Theorem 5.3.2 and the other two cases are trivial. Hence, we conclude that to reach a value strictly larger than -1, L must move on the first pile.

Consider the pair of strategies described in the Kuhn tree of Figure 5.6. To improve readability, we have not indicated all the information sets. Note that this strategy pair has value $\frac{1}{2}$. We will prove that the pair is a Nash equilibrium.

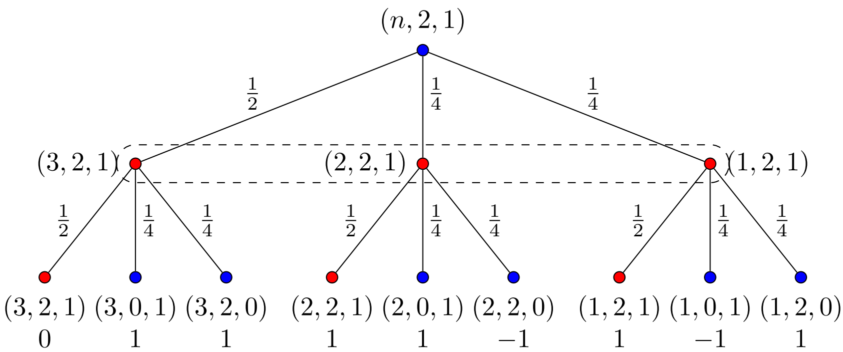


Figure 5.6: A Nash equilibrium for SN$(n, 2, 1)$.

First, suppose R deviates. Note that taking one chip from the second pile in his move leads to a state in $\{(3,1,1),(2,1,1),(1,1,1)\}$ with perfect information for L, which is a certain loss for R by Theorem 5.3.2. Therefore, we only need to consider a strategy (q_1, q_2, q_3) over the three moves described. Following this strategy, the payoff for L is

$$q_1(\tfrac{1}{2} \cdot 0 + \tfrac{1}{4} \cdot 1 + \tfrac{1}{4} \cdot 1) + q_2(\tfrac{1}{2} \cdot 1 + \tfrac{1}{4} \cdot 1 + \tfrac{1}{4} \cdot -1) + q_3(\tfrac{1}{2} \cdot 1 + \tfrac{1}{4} \cdot -1 + \tfrac{1}{4} \cdot 1)$$
$$= \tfrac{1}{2}(q_1 + q_2 + q_3) = \tfrac{1}{2}.$$

Hence, no matter the strategy of R, the value remains $\frac{1}{2}$.

Now, suppose L deviates and chooses to pick i chips from the first pile with probability p_i, $0 < i < n$; taking all chips results in a certain loss. Let v_i denote the value obtained if R chooses to inspect and play on the first pile, then the value of the strategy pair $(p_1, \ldots, p_{n-1})$ and the fixed strategy of R is

$$\sum_{i=1}^{n-3} p_i(\tfrac{1}{2}v_i + \tfrac{1}{2}) + \tfrac{1}{2}p_{n-2}v_{n-2} + \tfrac{1}{2}p_{n-1}v_{n-1}. \tag{5.1}$$

We explain this as follows. By removing i chips from the first pile, $1 \leq i \leq n-3$, L moves to $(n-i, 2, 1)$ with $3 \leq n-i < n$. In these states, touching the second or third pile causes a certain loss for R, while inspecting the first pile yields v_i. Hence, the value obtained from this state is $\frac{1}{2}v_i + \frac{1}{4} + \frac{1}{4} = \frac{1}{2}v_i + \frac{1}{2}$. If L removes $n-2$ or $n-1$ chips from the first pile, we end up in (2,2,1) or (1,2,1). From Figure 5.6, it follows that by the fixed strategy of R, the value obtained in these states is $\frac{1}{2}v_i + \frac{1}{4} - \frac{1}{4} = \frac{1}{2}v_i$. Hence the formula indeed holds.

Next, we determine v_i. Again, by Figure 5.6, we immediately see that $v_{n-3} = 0$ and $v_{n-2} = v_{n-1} = 1$, where the former follows from Theorem 5.3.3. To determine v_i for $1 \leq i < n-3$, note that, by the reasoning at the start of the proof, playing on the first pile is optimal for the first player if starting in $(k, 2, 1)$ with $k \geq 4$. To determine the value for these games, we proceed by induction on k.

For the game starting in (4,2,1), again consider the tree in Figure 5.6. We already saw that R cannot improve his payoff by deviating. Hence, suppose now that L picks i chips from the first pile with probability p_i, $i = 1, 2, 3$. The value of the game then becomes

$$p_1(\tfrac{1}{4} + \tfrac{1}{4}) + p_2(\tfrac{1}{2} + \tfrac{1}{4} - \tfrac{1}{4}) + p_3(\tfrac{1}{2} - \tfrac{1}{4} + \tfrac{1}{4}) = \tfrac{1}{2}(p_1 + p_2 + p_3) = \tfrac{1}{2}.$$

Hence, L can also not improve, so the given strategy pair is Nash and the value of the game is $\frac{1}{2}$. Now, by induction, it follows that $v_i = -\frac{1}{2}$ for $i = 1, \ldots, n-4$.

Substituting the values of the v_i into (5.1) yields

$$\frac{1}{4}\sum_{i=1}^{n-4} p_i + \frac{1}{2}\sum_{i=n-3}^{n-1} p_i.$$

Maximizing over all probability distributions $(p_1, \ldots, p_{n-1})$ yields a value of $\frac{1}{2}$. Hence, L can indeed not improve her payoff by deviating. □

Theorem 5.3.5. *Let $n \in \mathbb{N}$, $n \geq 3$. Then $v(SN(n, 3, 1)) = \frac{1}{3}$.*

Proof. We consider the strategy pair in Figure 5.7, where again we denote only the state. The value of the pair is $\frac{1}{3}$. Note that this strategy pair does not fix the strategies followed in the states not encountered in the tree. If we do encounter such a state when one of the players deviates, we fill in the blanks at that time. For the values of the leaves, note that in the leftmost information set for L, she

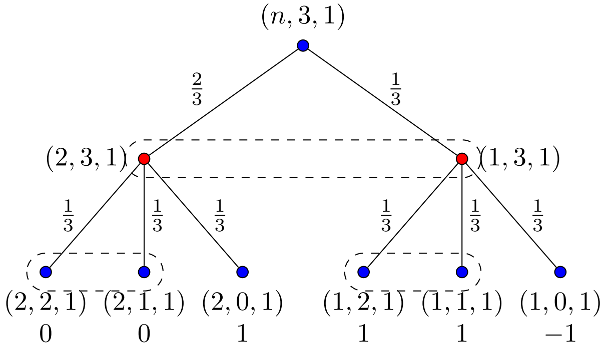


Figure 5.7: A Nash equilibrium for SN$(n, 3, 1)$.

must empty the first pile, leading to a win half of the times. In the rightmost set, emptying the second pile is always a winning strategy.

Now, suppose R deviates. Note that inspecting and playing on the first pile leads to (1,3,1) or (0,3,1) with perfect information for L, yielding a loss for R. Emptying the third pile leads to (2,3,0) or (1,3,0) with perfect information for L, also giving a loss for R. Hence, the three actions depicted in Figure 5.7 are indeed the only interesting ones. Picking i chips from the second pile with probability q_i, the value becomes

$$\tfrac{2}{3}q_3 + \tfrac{1}{3}(q_1 + q_2 - q_3) = \tfrac{1}{3}(q_1 + q_2 + q_3) = \tfrac{1}{3}.$$

Hence, deviation by R does not lead to a smaller value.

Next, suppose L deviates. First, note that emptying the last pile leads to a payoff of 0. Therefore, any strategy which does not empty the last pile on the first move which yields a non-negative payoff cannot be improved by choosing to empty the last pile with a non-zero probability.

Similarly, consider a strategy for L in which she moves on the second pile in the first turn. Note that emptying the pile leads to $(n, 0, 1)$, which is a loss for L. Not emptying the pile, L moves to $(n, 2, 1)$ or $(n, 1, 1)$. We depict these choices and a fixed counterstrategy for R in Figure 5.8.

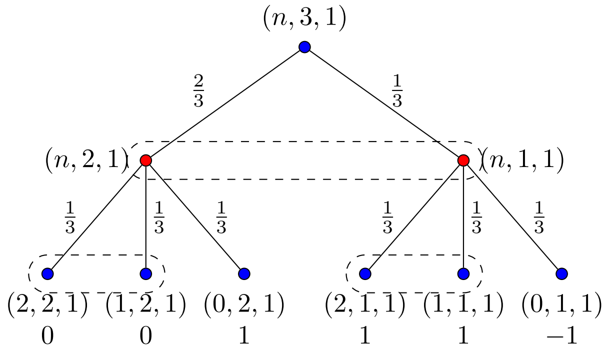


Figure 5.8: Player L moving on the second pile and the response of R.

Note that in the information set $[\overline{2}, 2, 1]$ for L, her best response is to empty the second or third pile, both with probability $\frac{1}{2}$, leading to a value of 0 as shown in the figure. Picking both branches with probabilities p_1 and p_2 respectively in the first move, L obtains a payoff of

$$\tfrac{1}{3}p_1 + p_2(\tfrac{1}{3} + \tfrac{1}{3} - \tfrac{1}{3}) = \tfrac{1}{3}(p_1 + p_2) = \tfrac{1}{3}.$$

Hence, L cannot obtain a value higher than $\frac{1}{3}$ by playing on the second pile. Therefore, we may assume that L plays on the first pile, taking i chips with probability p_i, $1 \leq i < n$; note that taking all chips results in a certain loss. Suppose that R consequently takes any number of chips from the second pile with probability $\frac{1}{3}$.

Now, L finds herself in some information set $[n-i, \overline{2}, 1]$ or $[n-i, 0, 1]$. The latter means a loss for L if and only if $i = n-1$. Suppose therefore that L is in the set $[n-i, \overline{2}, 1]$. She now chooses to pick j chips from the first pile with probability

q_j^{n-i}, $j = 1, \ldots, n-i$, chooses to inspect the second pile with probability r_1^{n-i}, and to empty the third pile with probability r_2^{n-i}.

First, consider the information set $[1, \overline{2}, 1]$. In this case, inspecting and emptying the second pile is always a winning move, so we set $q_1^1 = r_2^1 = 0$ and $r_1^1 = 1$. For $[k, \overline{2}, 1]$ with $k \geq 3$, emptying the third pile is always losing, so $r_3^k = 0$ for $k = 2, \ldots, n-1$. The same goes for inspecting and moving on the second pile for $k \geq 2$, so also $r_1^k = 0$ for $k = 2, \ldots, n-1$.

From the information set $[k, \overline{2}, 1]$ for $k \geq 2$, by removing $1 \leq j < k$ chips from the first pile, we arrive in the information set $[\overline{n-2}, 2, 1]$ or $[\overline{n-2}, 1, 1]$ for R. In the latter case, inspecting and emptying the first pile always results in a win for R. In the former case, things are more complicated. Taking one chip from the second pile always results in a loss for R. Taking two (and thus emptying it) results in a win for R if and only if he is in the state $(1, 2, 1)$. Emptying the third pile leads to the information set $[k, \overline{2}, 0]$ for L. However, as R would always empty the first pile if the second pile has size 1, L may conclude to be in the state $(k, 2, 0)$. Therefore, emptying the third pile is winning for R if and only if $k = 2$.

The final option for R is to inspect the first pile and act accordingly to what he finds. Going from $[\overline{n-2}, 2, 1]$, emptying it is always a loss for R, and hence choosing the inspection action in $(1, 2, 1)$ results in a certain loss. Not emptying the first pile reveals to L that the height of the second pile is 2, like before. Therefore, choosing the inspection action in the state (2,2,1) leads to the state (1,2,1) or (0,2,1) with perfect information for L, both being a win. If R finds himself to be in a state $(k, 2, 1)$ for $k \geq 3$, we fix his strategy to taking all but one or two chips from the first pile, both with probability $\frac{1}{2}$. L then finds herself either in (2,2,1) or (1,2,1) with equal probability, which, like assessed earlier, leads to a payoff of 0.

Now, suppose R, in the information set $[\overline{n-2}, 2, 1]$, chooses to inspect the first pile with probability s_1, empty the second pile with probability s_2 and empty the third pile with probability s_3. The value of the strategy pair of L and R can then be computed as follows:

$$
\begin{aligned}
&\tfrac{1}{3}p_{n-1}(1+1-1) + \tfrac{1}{3}p_{n-2}(q_1^2(s_1-s_2+s_3) - q_2^2 + r_1^2 - q_1^2 + q_2^2 - r_1^2 + 1) \\
&\quad + \tfrac{1}{3}p_{n-3}(q_1^3(s_1+s_2-s_3) + q_2^3(s_1-s_2+s_3) - q_3^3 - q_1^3 - q_2^3 + q_3^3 + 1) \\
&\quad + \sum_{k=4}^{n-1} \tfrac{1}{3}p_{n-k}\left(\sum_{j=1}^{k-3} q_j^k(s_2+s_3) + q_{k-2}^k(s_1+s_2-s_3) + q_{k-1}^k(s_1-s_2+s_3)\right. \\
&\quad \left. -q_k^k - \sum_{j=1}^{k-1} q_j^k + q_k^k + 1\right) \\
&= \tfrac{1}{3}p_{n-1} + \tfrac{1}{3}p_{n-2}(q_1^2(s_1-s_2+s_3-1)+1) \\
&\quad + \tfrac{1}{3}p_{n-3}(q_1^3(s_1+s_2-s_3-1) + q_2^3(s_1-s_2+s_3-1)+1) \\
&\quad + \sum_{k=4}^{n-1} \tfrac{1}{3}p_{n-k}\left(\sum_{j=1}^{k-3} q_j^k(s_2+s_3-1) + q_{k-2}^k(s_1+s_2-s_3-1)\right. \\
&\quad \left. +q_{k-1}^k(s_1-s_2+s_3-1)+1\right)
\end{aligned}
$$

Now, note that $s_1+s_2+s_3=1$ so that $s_1 \pm s_2 \pm s_3 - 1 \leq 0$ no matter the signs of the s_i. Therefore, maximizing over the possible choices, L will assign value $q_j^k = 0$ to those q_j^k multiplied by a factor $s_1 \pm s_2 \pm s_3 - 1$. The value of the best response strategy therefore becomes

$$\tfrac{1}{3}p_{n-1} + \tfrac{1}{3}p_{n-2} + \tfrac{1}{3}p_{n-3} + \sum_{k=4}^{n-1} \tfrac{1}{3}p_{n-k} = \tfrac{1}{3}\sum_{k=1}^{n-1} p_k = \tfrac{1}{3}.$$

This proves that L also cannot gain more by deviating. Hence, the depicted strategy pair is indeed Nash with value $\frac{1}{3}$. □

Theorem 5.3.6. *Let $n_1, n_2 \in \mathbb{N}$, $n_1, n_2 \geq 3$. Then $v(SN(n_1,n_2,2)) = \frac{1}{3}$.*

Proof. Consider the strategy pair in Figure 5.9. Note that in the fourth row, the two nodes circled by a dashed line together form an information set, as well as the two nodes circled by a dotted line. Again, we denote only the states, and again, the strategies are not yet exhaustive.

Suppose R deviates. In his first turn, he finds herself in one of the three states $(3,n_2,2)$, $(2,n_2,2)$ or $(1,n_2,2)$ with probability $\frac{1}{3}$. Emptying the third pile leads

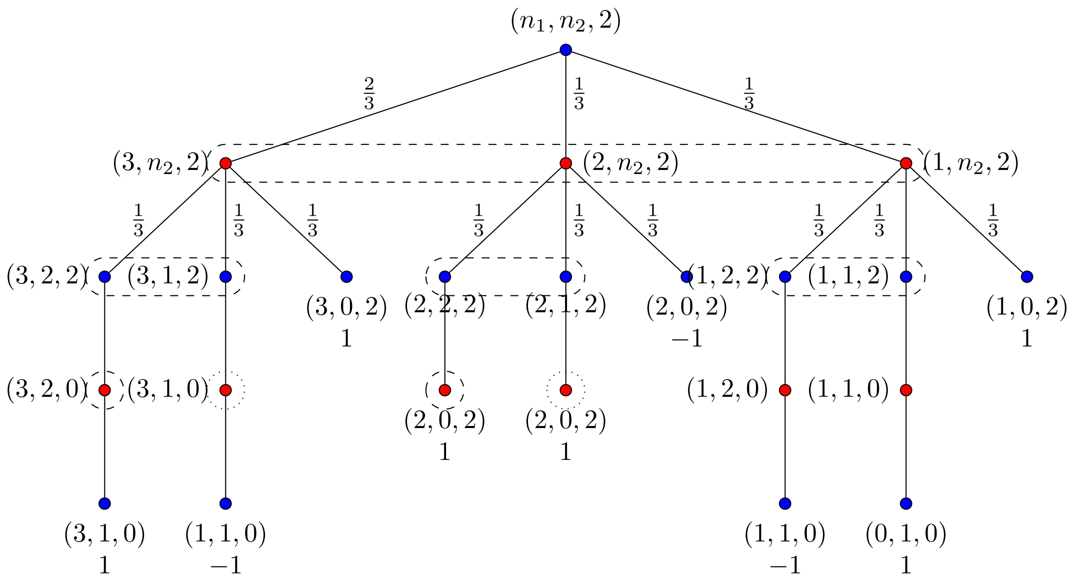


Figure 5.9: A Nash equilibrium for $\mathrm{SN}(n_1, n_2, 2)$.

to a value of 0 in the case of $(3, n_2, 2)$ and a win for L in the other two cases. Taking one chip for the third pile leads to a value of at least $\frac{1}{3}$, $\frac{1}{2}$ and 1, respectively, by Theorems 5.3.5, 5.3.4 and 5.3.2. Note that L has perfect information after R's move, while R has not, hence the values might be larger. Therefore, R will not play on the third pile.

Next, consider R inspecting the first pile. Emptying the pile leads to a win for L, so this move leads to a win for L if done in $(1, n_2, 2)$. From $(2, n_2, 2)$, we move to $(1, n_2, 2)$ with perfect information for both players, yielding a value of $\frac{1}{2}$. From $(3, n_2, 2)$, say R moves to $(2, n_2, 2)$ or $(1, n_2, 2)$ with probability q_1 and q_2, respectively. This confronts L with the information set $[\overline{2}, n_2, 1]$. We define her strategy in this set to play the winning strategy for starting from $(2, n_2, 2)$ or $(1, n_2, 2)$ both w.p. $\frac{1}{2}$, following the proofs of Theorems 5.3.2 and 5.3.4. Hence, with probability $\frac{1}{2}$, L will empty the second pile, w.p. $\frac{1}{4}$ she will reduce the second pile to 3 chips and w.p. $\frac{1}{8}$ she will reduce it to 2 or 1 chip. This gives a total value of

$$q_1(\tfrac{1}{2} \cdot -1 + (\tfrac{1}{4} + \tfrac{1}{8} + \tfrac{1}{8}) \cdot \tfrac{1}{2}) + q_2(\tfrac{1}{2} \cdot 1 + (\tfrac{1}{4} + \tfrac{1}{8} + \tfrac{1}{8}) \cdot -1) = -\tfrac{1}{4}q_1.$$

Minimizing this value, R will thus pick $q_1 = 1$ and $q_2 = 0$ to enforce $-\frac{1}{4}$. Thus, by choosing to inspect the first pile, the value becomes $\frac{1}{3}(-\frac{1}{4} + \frac{1}{2} + 1) = \frac{5}{12}$. As this is larger than $\frac{1}{3}$, R cannot gain from choosing this option.

Therefore, suppose R chooses to remove i chips from the second pile with probability p_i, $i = 1, \dots, n_2$. Removing n_2 chips and thus emptying the pile leads to a win for R if and only if the other piles are of size 2. Removing any other amount of chips brings L into the info set $[3, \overline{n_2 - 1}, 2]$, $[2, \overline{n_2 - 1}, 2]$ or $[1, \overline{n_2 - 1}, 2]$. Her strategy for these sets is determined by Figure 5.9. From $[3, \overline{n_2 - 1}, 2]$ and $[1, \overline{n_2 - 1}, 2]$, L empties the third pile. From $[2, \overline{n_2 - 1}, 2]$, L inspects and empties the second pile, moving to the state $(2, 0, 2)$ and winning the game. Hence, after L's move, if R is not losing the game, he finds herself in some state $(3, k, 0)$ or $(1, k, 0)$, having full knowledge on the second pile, but not on the first. Hence, abusing notation, we can describe his information set as $I_k = [\{1, 3\}, k, 0]$.

From the set I_k, let R inspect the first pile with probability r^k. If there was only one chip, he must empty the pile and loses the game. If there are three chips left, he picks one with probability s_1^k and two with probability s_2^k; picking all three leads to a loss. This move by R brings L into the information set $[\overline{2}, \overline{n_2 - 1}, 0]$. We define her strategy in this info set by inspecting the second pile and reducing it to either one or two chips, both with probability $\frac{1}{2}$, if possible. Now, L wins if and only if the two remaining piles are of the same size after this move.

Therefore, the best R can do is to pick $s_1^k = s_2^k = \frac{1}{2}$ for $k = 1, \dots, n_2 - 3$, forcing a value of 0. For $k = n_2 - 2$, R picks $s_1^k = 1$ and $s_2^k = 0$ to move to $(2, 2, 0)$ and win the game. Similarly, for $k = n_2 - 1$, R picks $s_1^k = 0$ and $s_2^k = 1$ to move to $(1, 1, 0)$ and win.

Finally, R chooses to pick i chips from the second pile with probability q_i^k, $i = 1, \dots, k - 1$; note again that emptying the pile leads to a loss. This moves to the information set $[3, \overline{n_2 - 2}, 0]$ or $[1, \overline{n_2 - 2}, 0]$ for L. In the first case, L will always reduce the first pile to one chip, winning if and only if the second pile is also of size one and losing otherwise. In the second case, L will inspect the second pile and reduce it to size 1 if possible, losing if and only if it already contained exactly one chip, and winning otherwise.

In summary, the value of the game obtained when R plays his deviating strategy becomes

$$\begin{aligned}
&\frac{1}{3}\Big[\sum_{k=1}^{n_2-3} p_k\big(-\sum_{i=1}^{n_2-2-k} q_i^k + q_{n_2-1-k}^k + 1 + r^k + \sum_{i=1}^{n_2-2-k} q_i^k - q_{n_2-1-k}^k\big) \\
&\quad + p_{n_2-2}(-r^{n_2-2} + q_1^{n_2-2} + 1 + r^{n_2-2} - q_1^{n_2-2}) + p_{n_2-1}(-r^{n_2-1} + r^{n_2-1}) \\
&\quad + p_{n_2}(1 - 1 + 1)\Big] \\
&= \frac{1}{3}\Big[\sum_{k=1}^{n_2-3} p_k(1 + r^k) + p_{n_2-2} + p_{n_2-1} + p_{n_2}\Big].
\end{aligned}$$

Minimizing this value, R will choose $r^k = 0$ for $k = 1, \dots, n_2 - 3$ so that we obtain

$$\frac{1}{3}\sum_{k=1}^{n_2} p_k = \frac{1}{3}$$

as value. Hence, R can indeed not improve by choosing a different strategy.

Now, suppose L deviates. Without loss of generality, we may assume that L does not play on the second pile. If L removes one chip from the second pile, R can remove the second chip on his turn, moving to $(n_1, n_2, 0)$. Hence, R can force an outcome of at most 0. Similarly, by emptying the third pile, L obtains a payoff of 0. Therefore, if a value of more than 0 can be obtained by playing on the first pile, L will always choose to do so.

Hence, assume L picks i chips from the first pile with probability p_i, $i = 1, \dots, n_1 - 1$; emptying leads to $(0, n_2, 2)$ and thus a loss for L. Next, R will reduce the second pile to size 0, 1 or 2, each with probability $\frac{1}{3}$. If R emptied the second pile, L will find herself in the state $(k, 0, 2)$ with $k = n_1 - i$, losing if and only if $k = 2$ and winning otherwise. Otherwise, L is faced with the info set $[k, \{1, 2\}, 2]$.

First, note that if $k = 2$, L can always win by emptying the second pile. We therefore further assume that $k \neq 2$. Now, L can pick i chips from the first pile with probability q_i^k, $i = 1, \ldots, k$. If the second pile is of size 2, L wins if and only if she empties the first pile, and loses otherwise. If the second pile is of size 1, L loses if she empties the first pile. Otherwise, she moves to the information set $[\overline{n_1 - 2}, 1, 2$ for R. For this set, we define the strategy of R to be the equilibrium strategy for the game $(n_1, 1, 2)$ as described in the proof of Theorem 5.3.4, which plays on the first pile in the first move. Therefore, if L reduced the first pile to size 1 or 2, she wins; size 3 yields a value of at most 0 and otherwise, a value of at most $-\frac{1}{2}$ is obtained.

Next, L can choose to inspect the second pile with probability r^k. If the pile was of size 1, it is now emptied, moving to the information set $[\overline{n_1 - 1}, 0, 2; \sigma_1]$ for *R*, with σ_1 being the choice to reduce the second pile to size 1 on R's first turn. We define R's strategy in this set to inspect the first pile and reduce it to size 2 if possible. This strategy is losing for R if and only if the first pile was of size 1, and winning otherwise. If the second pile was of size 2, L can choose to pick 1 or 2 chips from it, with probability s_1^k and s_2^k respectively. Picking one chip leads to the info set $[\overline{n_1 - 1}, 1, 2]$ for R. We let R empty the third pile in this case, winning if and only if $k = 1$ and losing otherwise. Picking two chips leads to $[\overline{n_1 - 1}, 0, 2; \sigma_2]$, where σ_2 denotes R's choice to reduce the second pile to size 2 on his first turn. In this set, we let R pick one chip from the third pile, again winning if and only if $k = 1$ and losing otherwise.

Finally, L can pick one or two chips from the third pile with probability t_1^k and t_2^k respectively. First, consider the case where the second pile is of size 2. Picking one chip from the third pile then leads to $[\overline{n_1 - 1}, 2,]$ for R, from where we let R play like in Theorem 5.3.4 again. Hence, we obtain a value of 1 if $k = 1$, at most 0 if $k = 3$ and at most $-\frac{1}{2}$ otherwise. Picking two chips leads to $[\overline{n_1 - 1}, 2, 0]$, from where R must pick one chip from the second pile as defined by Figure 5.9. Hence, he wins if $k = 1$ and loses otherwise. Next, consider the second pile being of size 1. Removing one chip from the third pile then leads to $[\overline{n_1 - 1}, 1, 1]$. Inspecting and emptying the first pile always leads to a win for R. Finally, removing both chips leads to $[\overline{n_1 - 1}, 1, 0]$, from where R can win if and only if $k \neq 1$.

Combined, L can obtain at most the following value:

$$
\begin{aligned}
&\frac{1}{3}\Bigg[\sum_{k=1}^{n_1-5} p_k \Bigg(-\sum_{i=1}^{n_1-1-k} q_i^{n_1-k} + q_{n_1-k}^{n_1-k} + r^{n_1-k} - \tfrac{1}{2}t_1^{n_1-k} \\
&\qquad + t_2^{n_1-k} - \tfrac{1}{2}\sum_{i=1}^{n_1-4-k} q_i^{n_1-k} + q_{n_1-2-k}^{n_1-k} + q_{n_1-1-k}^{n_1-k} - q_{n_1-k}^{n_1-k} \\
&\qquad - r^{n_1-k} - t_1^{n_1-k} - t_2^{n_1-k} + 1 \Bigg) \\
&\quad + p_{n_1-4}(-q_1^4 - q_2^4 - q_3^4 + q_4^4 + r^4 - \tfrac{1}{2}t_1^4 + t_2^4 \\
&\qquad + q_2^4 + q_3^4 - q_4^4 - r^4 - t_1^4 - t_2^4 + 1) \\
&\quad + p_{n_1-3}(-q_1^3 - q_2^3 + q_3^3 + r^3 + t_2^3 + q_1^3 + q_2^3 - q_3^3 - r^3 - t_1^3 - t_2^3 + 1) \\
&\quad + p_{n_1-2}(1 + 1 - 1) \\
&\quad + p_{n_1-1}(q_1^1 - r^1 + t_1^1 - t_2^1 - q_1^1 + r^1 - t_1^1 + t_2^1 + 1)\Bigg] \\
&= \frac{1}{3}\Bigg[\sum_{k=1}^{n_1-5} p_k(-\tfrac{3}{2}\sum_{i=1}^{n_1-4-k} q_i^{n_1-k} - \tfrac{3}{2}t_1^{n_1-k} + 1) \\
&\quad + p_{n_1-4}(-q_1^4 - \tfrac{3}{2}t_1^4 + 1) + p_{n_1-3}(-t_1^3 + 1) + p_{n_1-2} + p_{n_1-1}\Bigg].
\end{aligned}
$$

Maximizing this value, L will pick all probabilities remaining in this expression equal to zero, leading to a value of

$$\frac{1}{3}\sum_{k=1}^{n_1-1} p_k = \frac{1}{3}.$$

This finishes the proof. □

The games SN$(n_1, n_2, 1)$ with $n_1, n_2 \geq 4$ do not have a constant value. Some computed values for small n_1 and n_2 are shown in Table 5.2.

For this configuration of the game, we present some structural results. The proofs of these results (partly) rely on the following observation, stating the principle of indifference for games in extensive form. Note that this is true for these games in general, not only for Schrödinger Nim.

Table 5.2: Values for the game SN$(n_1, n_2, 1)$.

$n_2 \backslash n_1$	4	5	6	7
4	$1/5$	$1/4$	$1/4$	$1/4$
5		$3/13$	$5/21$	$5/21$
6			$4/17$	$13/55$

Theorem 5.3.7. *Let $\mu = (\mu_L, \mu_R)$ be a Nash equilibrium for a two-player zero-sum game in extensive form with perfect recall, in which L moves first and R moves second. Write $v = v(\mu)$, and let r be the root of the Kuhn tree of the game. Then*

(1) *$v_w(\mu) = v$ for all $w \in N^+(r)$ with $\mu_L(\alpha_r(w)) > 0$.*
(2) *$\sum_{w\in S} \mu_L(\alpha_r(w))v_{\alpha_w^{-1}(a)}(\mu) = \sum_{w\in S} \mu_L(\alpha_r(w))v$ for all $a \in A(S)$ with $\mu_R(a) > 0$, where S is an information set for which $\sigma_R(S) = \emptyset$.*

Proof. (1) Let $w \in N^+(r)$ be such that $\mu_L(\alpha_r(w)) > 0$. Suppose now that $v_w(\mu) \neq v$ and assume first that $v_w(\mu) > v$. Define the new strategy μ'_L for L by $\mu'_L(\alpha_r(x)) = \mathbb{1}_{\{x=w\}}$ for $x \in N^+(r)$ and $\alpha'_L(a) = \alpha_L(a)$ for all other actions a, and let $\mu' = (\mu'_L, \mu_R)$. By definition,

$$v_r(\mu') = \sum_{x\in N^+(r)} \mu'_L(\alpha_r(x))v_x(\mu') = v_w(\mu') = v_w(\mu) > v,$$

which contradicts the fact that μ is Nash. Now, suppose $v_w(\mu) < v$. As $v = \sum_{x\in N^+(r)} \mu_L(\alpha_r(x))v_x(\mu)$, there must be some $w' \in N^+(r)$ with $v_{w'}(\mu) > v$; otherwise we find

$$v = \sum_{x\in N^+(r)} \mu_L(\alpha_r(x))v_x(\mu) < \sum_{x\in N^+(r)} \mu_L(\alpha_r(x))v = v.$$

Now, repeating the above argument for w' instead of w completes the proof of (1).

(2) Let S be an information set for which $\sigma_R(S) = \emptyset$ and let $a \in A(S)$ be such that $\mu_R(a) > 0$. Suppose first that

$$\sum_{w\in S} \mu_L(w^-)v_{\alpha_w^{-1}(a)}(\mu) < \sum_{w\in S} \mu_L(w^-)v$$

Define μ'_R by $\mu'_R(k) = \mathbb{1}_{\{k=a\}}$ on $A(S)$ and $\mu'_R \equiv \mu_R$ elsewhere. Then

$$\begin{aligned}
v_r(\mu') &= \sum_{w \in N^+(r)} \mu_L(\alpha_r(w)) v_w(\mu') \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v_w(\mu') + \sum_{w \in S} \mu_L(\alpha_r(w)) v_w(\mu') \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v_w(\mu) \\
&\quad + \sum_{w \in S} \mu_L(\alpha_r(w)) \sum_{k \in A(S)} \mu'_R(k) v_{\alpha_w^{-1}(k)}(\mu') \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v + \sum_{w \in S} \mu_L(\alpha_r(w)) v_{\alpha_w^{-1}(a)}(\mu') \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v + \sum_{w \in S} \mu_L(\alpha_r(w)) v_{\alpha_w^{-1}(a)}(\mu) \\
&< \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v + \sum_{w \in S} \mu_L(\alpha_r(w)) v = v,
\end{aligned}$$

contradiction. Suppose next that

$$\sum_{w \in S} \mu_L(\alpha_r(w)) v_{\alpha_w^{-1}(a)}(\mu) > \sum_{w \in S} \mu_L(\alpha_r(w)) v.$$

Then there must be some $a' \in A(S)$ for which

$$\sum_{w \in S} \mu_L(\alpha_r(w)) v_{\alpha_w^{-1}(a')}(\mu) < \sum_{w \in S} \mu_L(\alpha_r(w)) v;$$

indeed, otherwise we find

$$\begin{aligned}
v &= \sum_{w \in N^+(r)} \mu_L(\alpha_r(w)) v_w(\mu) \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v_w(\mu) + \sum_{w \in S} \mu_L(\alpha_r(w)) v_w(\mu) \\
&= \sum_{w \in N^+(r) \setminus S} \mu_L(\alpha_r(w)) v_w(\mu) + \sum_{w \in S} \mu_L(\alpha_r(w)) \sum_{k \in A(S)} \mu_R(k) v_{\alpha_w^{-1}(k)}(\mu)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{w \in N^+(r)\setminus S} \mu_L(\alpha_r(w))v + \sum_{k \in A(S)} \mu_R(k) \sum_{w \in S} \mu_L(\alpha_r(w)) v_{\alpha_w^{-1}(k)}(\mu) \\
&> \sum_{w \in N^+(r)\setminus S} \mu_L(\alpha_r(w))v + \sum_{k \in A(S)} \mu_R(k) \sum_{w \in S} \mu_L(\alpha_r(w)) v \\
&= \sum_{w \in N^+(r)\setminus S} \mu_L(\alpha_r(w))v + \sum_{w \in S} \mu_L(\alpha_r(w)) \sum_{k \in A(S)} \mu_R(k) v \\
&= \sum_{w \in N^+(r)\setminus S} \mu_L(\alpha_r(w))v + \sum_{w \in S} \mu_L(\alpha_r(w)) v = v,
\end{aligned}
$$

contradiction. This completes the proof of (2).

□

Lemma 5.3.8. *Let $n_2 \in \mathbb{N}$, $n_2 \geq 3$. Then $v(SN(n_1, n_2, 1))$ is non-decreasing in n_1, for $n_1 \geq n_2$.*

Proof. Consider the game $\mathrm{SN}(n_2, n_2, 1)$. Suppose first that there exists a Nash equilibrium (μ_L, μ_R) for which $\mu_L(\mathbf{3}_1) = 0$ with payoff $v := v(\mathrm{SN}(n_2, n_2, 1))$. Then, for the game $\mathrm{SN}(n_1, n_2, 1)$ with $n_1 > n_2$, L may use the same strategy as for the game $\mathrm{SN}(n_2, n_2, 1)$, playing $\mathbf{1}_i$ with $i > n_1 - n_2$ on the first turn to obtain a value of v in this game, as well.

If such an equilibrium does not exist, let $\mu = (\mu_L, \mu_R)$ be Nash such that $\mu_L(\mathbf{3}_1) > 0$. Note that $0 = v(\mathrm{SN}(n_2, n_2, 0)) = v_{\mathbf{3}_1}(\mu) = v(\mathrm{SN}(n_2, n_2, 1))$ by Theorem 5.3.7. Therefore, we may define $\mu' = (\mu'_L, \mu_R)$ with $\mu'_L(\mathbf{3}_1) = 1$ as another Nash equilibrium strategy pair. Now, for $\mathrm{SN}(n_1, n_2, 1)$ with $n_1 > n_2$, L may also force a value of 0 by picking $\mathbf{3}_1$ with probability 1 as first move. Hence, indeed $v(\mathrm{SN}(n_1, n_2, 1)) \geq v(\mathrm{SN}(n_2, n_2, 1))$ for $n_1 > n_2$. □

Lemma 5.3.9. *Let $n_1, n_2 \in \mathbb{N}$, $n_1, n_2 \geq 4$. Then $v(SN(n_1, n_2, 1)) \in [0, \frac{1}{4}]$.*

Proof. By playing $\mathbf{3}_1$ as first move, L guarantees a value of 0 by Theorem 5.3.1. Therefore, $v(\mathrm{SN}(n_1, n_2, 1)) \geq 0$.

To prove the upper bound of $\frac{1}{4}$, we give an explicit strategy for R against which L cannot obtain a value of more than $\frac{1}{4}$. Suppose without loss of generality that L plays on the first heap during her first move. Emptying this heap leads to a win for R, so we may assume that L plays $\mathbf{1}_i$ for some $i \in \{1, \ldots, n_1 - 1\}$.

Now, let R play $\mathbf{2}_{n_2-3}$ with probability $\frac{1}{8}$, $\mathbf{2}_{n_2-2}$ or $\mathbf{2}_{n_2-1}$ both with probability $\frac{2}{8}$ and $\mathbf{2}_{n_2}$ with probability $\frac{3}{8}$. If L played $\mathbf{1}_{n_1-1}$, this yields a value of $\frac{1}{8}+\frac{2}{8}+\frac{2}{8}-\frac{3}{8}=\frac{2}{8}$. If L played $\mathbf{1}_i$ for any $i \in \{1,\dots,n_1-2\}$, R's move $\mathbf{2}_{n_2}$ leads to a win for L, whereas any of his other moves leads to the information set $[n_1-i, \overline{n_2-1}, 1]$ for L. We continue by showing that L cannot obtain more than a value of $-\frac{1}{8}$ in any of these information sets, so the total value obtained will be at most $-\frac{1}{8}+\frac{3}{8}=\frac{2}{8}$.

First, note that in these information sets, inspecting the second heap will lead to a loss for L if it was of height 2 or 1, which is the case with probability $\frac{4}{8}$. Hence, this move will yield a value of at most $\frac{1}{8}-\frac{4}{8}=-\frac{3}{8}<-\frac{1}{8}$. Similarly, playing $\mathbf{1}_{n_1-i}$ yields a win for L if and only if the height of the second heap was 1, so this move results in a value of $-\frac{1}{8}-\frac{2}{8}+\frac{2}{8}=-\frac{1}{8}$. We may thus assume that L plays $\mathbf{1}_j$ with probability q_j for $j \in \{1,\dots,n_1-i-1\}$, facing R with the information set $[\overline{n_1-2}, \ell, 1]$ with $\ell \in \{1,2,3\}$; or that L plays $\mathbf{3}_1$ with probability r, leading to $[\overline{n_1-1}, \ell, 0]$ with $\ell \in \{1,2,3\}$.

In $[\overline{n_1-2}, 3, 1]$, let R play $\mathbf{2}_1$. In $[\overline{n_1-2}, 2, 1]$, R plays $\mathbf{2}_1$ or $\mathbf{2}_2$ both with probability $\frac{1}{2}$. This leaves L either in $[n_1-i-j, \overline{n_2-2}, 1]$ or $[n_1-i-j, 0, 1]$. In the latter case, L wins if and only if $n_1-i-j \neq 1$. In the former case, choosing to empty the first heap, say with probability s, leads to a win for L if the second heap consists of one chip, and a loss if it consists of two. Doing anything else leads to a loss for L if the second heap contains one chip. Finally, in $[\overline{n_1-2}, 1, 1]$, R inspects and empties the first heap, which leads to a loss for L.

In $[\overline{n_1-1}, 3, 0]$, let R play $\mathbf{2}_1$. Consequently, R will play as if he leaves L with two heaps of height 2, winning if and only if this is the case. In $[\overline{n_1-1}, 2, 0]$, R inspects the first heap, and reduces it to height 2 if possible, winning if this is the case, and losing otherwise. In $[\overline{n_1-1}, 1, 0]$, R also inspects the first heap, and consequently reduces it to one chip, winning the game.

Altogether, starting from the information set $[n_1-i, \overline{n_2-1}, 1]$ with $n_1-i \geq 3$, L will thus obtain a value of at most

$$\frac{1}{8}\left(\sum_{j=1}^{n_1-i-2} q_j\left((1-s)-s\right) + q_{n_1-i-1} + r\right) + \frac{2}{8}\left(\sum_{j=1}^{n_1-i-2} q_j\left(\tfrac{1}{2}(-(1-s)+s)+\tfrac{1}{2}\right) + q_{n_1-i-1}(\tfrac{1}{2}-\tfrac{1}{2}) - r\right) - \frac{2}{8}$$

$$= \frac{1}{8} \sum_{j=1}^{n_1-i-1} q_j - \frac{1}{8}r - \frac{2}{8} \leq -\frac{1}{8}.$$

If $n_1 - i = 2$, the signs in front of the r's are flipped, also leading to a total of at most $-\frac{1}{8}$. Hence, we indeed see that L indeed cannot obtain a value of more than $\frac{2}{8}$. □

Corollary 5.3.10. *Let $n_1 \in \mathbb{N}$, $n_1 \geq 5$. Then $v(SN(n_1, 4, 1)) = \frac{1}{4}$.*

Proof. We compute that $v(\mathrm{SN}(5, 4, 1)) = \frac{1}{4}$ using a sequence form linear programming formulation of the game. The result then immediately follows from Lemma 5.3.8 and Lemma 5.3.9. □

Corollary 5.3.11. *Let $n_1, n_2 \in \mathbb{N}$, $n_1, n_2 \geq 3$, $n_1 \geq n_2$. Then there exists a Nash equilibrium $\mu = (\mu_L, \mu_R)$ for which*

$$1 - 2\mu_L(\mathbf{1}_{n_1-1}) = 1 - 2\mu_R(\mathbf{2}_{n_2})$$

for the first turns of L and R. Moreover, if $\mu_L(\mathbf{1}_{n_1-1}) > 0$, it holds that

$$v(SN(n_1, n_2, 1)) = 1 - 2\mu_L(\mathbf{1}_{n_1-1}).$$

Proof. If it holds that $\mu_L(\mathbf{1}_{n_1-1}) = \mu_R(\mathbf{2}_{n_2}) = 0$, the statement is trivial. If $v(\mathrm{SN}(n_1, n_2, 1)) = 0$, there exists an equilibrium for which this holds, with $\mu_L(\mathbf{3}_1) = 1$. Therefore, we may assume for the remainder of the proof that $v(\mathrm{SN}(n_1, n_2, 1)) > 0$ and $\mu_L(\mathbf{3}_1) = 0$.

Next, by Theorem 5.3.7, if $\mu_L(\mathbf{1}_i) > 0$ for some $i \in \{1, \ldots, n_1 - 1\}$, as all these moves lead to the information set $[\overline{n_1 - 1}, n_2, 1]$ for Right, there exists a Nash equilibrium for which $\sum_{i=1}^{n_1-1} \mu_L(\mathbf{1}_i) = 1$.

If $\mu_L(\mathbf{1}_{n_1-1}) > 0$, by Theorem 5.3.7, $v = v_{(1,n_2,1)}(\mu)$ with R moving. Noting that R wins if and only if he removes all chips from the second heap, we find

$$v = v_\mu(1, n_2, 1) = -\mu_R(\mathbf{2}_{n_2}) + (1 - \mu_R(\mathbf{2}_{n_2})) = 1 - 2\mu_R(\mathbf{2}_{n_2}).$$

Conversely, if $\mu_R(\mathbf{2}_{n_2}) > 0$, again by Theorem 5.3.7, we find

$$\sum_{i=1}^{n_1-1} \mu_L(\mathbf{1}_i)v = \sum_{i=1}^{n_1-1} \mu_L(\mathbf{1}_i)v_{(i,0,1)}(\mu) = \sum_{i=1}^{n_1-2} \mu_L(\mathbf{1}_i) - \mu_L(\mathbf{1}_{n_1-1}).$$

If also $\mu_L(\mathbf{1}_{n_1-1}) > 0$, then we may assume that $\sum_{i=1}^{n_1-1} \mu_L(\mathbf{1}_i) = 1$ so that $v = 1 - 2\mu_L(\mathbf{1}_{n_1-1})$. Hence, if both $\mu_L(\mathbf{1}_{n_1-1}) > 0$ and $\mu_R(\mathbf{2}_{n_2}) > 0$, we are done.

Therefore, suppose that $\mu_L(\mathbf{1}_{n_1-1}) > 0$, but $\mu_R(\mathbf{2}_{n_2}) = 0$. Then $v = 1$. However, this is a contradiction with Lemma 5.3.9 and the values for SN$(n_1, n_2, 1)$ with $n_1 = 3$ or $n_2 = 3$.

Finally, suppose that $\mu_R(\mathbf{2}_{n_2}) > 0$, but $\mu_L(\mathbf{1}_{n_1-1}) = 0$. Then $\sum_{i=1}^{n_1-2} \mu_L(\mathbf{1}_i)v = \sum_{i=1}^{n_1-2} \mu_L(\mathbf{1}_i)$, so again $v = 1$, which is a contradiction. This completes the argument. □

5.4 Fuzzy Schrödinger Nim

In this section, we consider the variant of the game in which the emptying of a heap is not signalled to the other player. Now, it may happen that a player inspects the height of a heap only to find out that it is empty. In this case, the player must choose another heap to perform a move on, continuing until the player has successfully removed at least one chip. Recall that a game of this variant with starting configuration $n \in \mathbb{N}^d$ is denoted by FSN(n).

We have the following complete characterization.

Theorem 5.4.1. *Let $n \in \mathbb{N}^d$. Then*

$$v(FSN(n)) = \begin{cases} -1 & \text{if } \max_{k=1,\dots,d} n_k = 1 \text{ and } d \text{ is even,} \\ 1 & \text{if } \max_{k=1,\dots,d} n_k = 1 \text{ and } d \text{ is odd,} \\ & \text{or if } n_k > 1 \text{ for precisely one } k, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. If we have $\max_{k=1,\dots,n} n_k = 1$, the game is equivalent to regular Nim, so L wins if and only if there is an odd number of chips available. If $n_k > 1$ for precisely one k, L can reduce this heap to either 1 or 0 chips, depending on the parity of d, winning the game. This settles the first two cases.

For the last case, we distinguish between two situations. First, suppose that $n_k \geq 2$ for all $k = 1, \dots, d$. The proof for this situation is analogous to that of Theorem 5.3.1, employing induction on d. For the base case $d = 2$, we use a similar Nash strategy pair. Now, L reduces the first heap to size 0 or 1 both with probability $\frac{1}{2}$, and R does the same to the second heap. L wins precisely if one of the heaps has size 1 after the initial moves, which happens with probability

$\frac{1}{2}$, hence the value is indeed 0. Proving that the pair is indeed Nash is analogous to the proof in Theorem 5.3.1.

For the induction step, suppose $v(\text{FSN}(n)) = 0$ for all $n \in \mathbb{N}$ as above and consider the game FSN(n) for some $n \in \mathbb{N}^{d+1}$ with $n_k \geq 2$ for all k. Without loss of generality, let L play on the first heap on her first turn. Now, by also playing on the first heap during his first turn, R can guarantee a value of at most 0. Indeed, if the heap was already emptied by L, R will find out and be in the case $n \in \mathbb{N}^d$. Otherwise, R can empty the heap himself. Therefore, the best L can do is to empty any heap, giving value 0. This concludes the first situation.

For the second situation, in which at least one of the n_k equals 1, we split the proof into three parts. First, consider the game FSN$(n_1, n_2, 1)$ with $n_1, n_2 \geq 2$. We will show that the value of this game is 0. First, note that emptying the third heap on the first move of L leads to a value of 0. Remains to show that L cannot achieve a payoff of more than 0 by choosing another strategy. Hence, without loss of generality, let L pick i chips from the first heap with probability p_i. We give an explicit strategy for R which prohibits L from obtaining a payoff larger than 0.

On his first move, R is faced with the information set $[\overline{n_1 - 1}, n_2, 1]$. With probability $\frac{1}{2}$, R reduces the second heap to height 1, and with equal probability, R empties the heap. L is thus given the info set $[n_1 - i, \overline{n_2 - 1}, 1]$.

For the case $n_1 - i = 1$, this yields either the state $(1, 1, 1)$ or $(1, 0, 1)$, which gives value 1 or -1, respectively. For $n_1 - i = 0$, it results in the state $(0, 1, 1)$ or $(0, 0, 1)$, giving value -1 or 1, respectively.

Now, suppose $n_1 - i > 1$, and let L take j chips from the first heap with probability $q_j^{n_1-i}$, $j \in \{1, \ldots, n_1 - i\}$. L inspects the second heap with probability $r_1^{n_1-i}$ and L empties the third heap with probability $r_2^{n_1-i}$. For $j = 1, \ldots, n_1 - i - 2$, R ends up in the state $(n_1 - i - j, 1, 1)$ or $(n_1 - i - j, 0, 1)$ with $n_1 - i - j \geq 2$. Hence, by inspecting the first heap and reducing it to height 0 or 1, respectively, R can win. For $j = i - 1$, R ends up in $(1, 1, 1)$ or $(1, 0, 1)$, having value -1 or 1, respectively. Taking $j = i$ leads to $(0, 1, 1)$ or $(0, 0, 1)$, having values 1 and -1.

If L chooses to inspect the second heap, she must empty it if it was of height 1, leading to $(n_1 - i, 0, 1)$ for R, who will win as $n_1 - i > 1$ by assumption. If it was already empty, L may choose another move on $(n_1 - i, 0, 1)$ and thus win. Emptying the third heap leads to either $(n_1 - i, 1, 0)$ or $(n_1 - i, 0, 0)$, which are both winning for R.

Hence, in summary, the value obtained by L playing any strategy against the fixed strategy of R is

$$\begin{aligned} &p_{n_1}(-\tfrac{1}{2}+\tfrac{1}{2})+p_{n_1-1}(\tfrac{1}{2}-\tfrac{1}{2}) \\ &\quad+\sum_{i=1}^{n_1-2} p_i \left(\tfrac{1}{2}\left(\sum_{j=1}^{n_1-i-2} -q_j^{n_1-i} - q_{n_1-i-1}^{n_1-i} + q_{n_1-i}^{n_1-i} - r_1^{n_1-i} - r_2^{n_1-i}\right)\right. \\ &\qquad\qquad\left. + \tfrac{1}{2}\left(\sum_{j=1}^{n_1-i-2} -q_j^{n_1-i} + q_{n_1-i-1}^{n_1-i} - q_{n_1-i}^{n_1-i} + r_1^{n_1-i} - r_2^{n_1-i}\right)\right) \\ &= -\sum_{i=1}^{n_1-2} p_i \left(\sum_{j=1}^{n_1-i-2} q_j^{n_1-i} + r_2^{n_1-i}\right) \end{aligned}$$

Maximizing her payoff, L will thus choose the variables such that the value becomes 0. This proves that $v(\text{FSN}(n_1, n_2, 1)) = 0$ and concludes the first part.

Next, let $n \in \mathbb{N}^d$, $d \geq 3$ be such that $n_1, n_2 \geq 2$ and $n_3 = \ldots = n_d = 1$. We will again show that the value of this game is 0. We apply induction on d. The base case is FSN$(n_1, n_2, 1)$, which was shown to have value 0 above. Hence, let $d \geq 4$. If L picks any of the heaps except the first two, R finds himself in the situation $d-1$ immediately and we are done. Remains to show that L cannot obtain a payoff larger than 0 by choosing any other strategy.

Denote by 1^k the vector $(1, 1, \ldots, 1) \in \mathbb{N}^k$. Abusing notation, we may thus represent the starting configuration by $(n_1, n_2, 1^{d-2})$. Without loss of generality, suppose L picks i chips from the first heap in the first turn with probability p_i, $i \in \{1, \ldots, n_1\}$. Again, we give an explicit strategy for R which prevents L from scoring higher than 0.

On his first move, R is given the information set $[\overline{n_1-1}, n_2, 1^{d-2}]$. With probability $\frac{1}{2}$, R reduces the second heap to height 1, and with equal probability, R empties the heap. L is thus faced with the information set $[n_1-i, \overline{n_2-1}, 1^{d-2}]$.

For the case $n_1 - i = 1$, this gives either the state $(1, 1, 1^{d-2})$ or $(1, 0, 1^{d-2})$, which yields value $(-1)^{d-1}$ or $(-1)^d$, respectively. For $n_1 - i = 0$, it results in the state $(0, 1, 1^{d-2})$ or $(0, 0, 1^{d-2})$, giving value $(-1)^d$ or $(-1)^{d-1}$, respectively.

Now, suppose $n_1 - i > 1$, and let L remove i chips from the first heap with probability $q_j^{n_1-i}$, $j \in \{1, \ldots, n_1-i\}$. L inspects the second heap with probability $r_1^{n_1-i}$, and, without loss of generality, L empties the third heap with probability $r_2^{n_1-i}$. For $j \in \{1, \ldots, n_1-i-2\}$, R is given the state $(n_1-i-j, 1, 1^{d-2})$ or

$(n_1 - i - j, 0, 1^{d-2})$, with $n_1 - i - j \geq 2$. Hence, by inspecting the first heap and reducing it to height $\mathbb{1}_{\{d \text{ even}\}}$ or $\mathbb{1}_{\{d \text{ odd}\}}$, respectively, R can win. For $j = i-1$, R is led to $(1, 1, 1^{d-2})$ or $(1, 0, 1^{d-2})$, having value $(-1)^d$ or $(-1)^{d-1}$, respectively. Taking $j = i$ gives rise to $(0, 1, 1^{d-2})$ or $(0, 0, 1^{d-2})$, having values $(-1)^{d-1}$ and $(-1)^d$.

If L chooses to inspect the second heap, she must empty it if it contained a chip, leading to $(n_1 - i, 0, 1^{d-2})$ for R, who will win as $n_1 - i > 1$ by assumption. If it was empty, L must choose another move on $(n_1 - i, 0, 1^{d-2})$ and thus win. Emptying the third heap leads to either $(n_1 - i, 1, 0, 1^{d-3})$ or $(n_1 - i, 0, 0, 1^{d-3})$, which are both winning for R.

Hence, in summary, the value obtained by L playing any strategy against the fixed strategy of R is

$$
\begin{aligned}
&p_{n_1}\left(\tfrac{1}{2}(-1)^d + \tfrac{1}{2}(-1)^{d-1}\right) + p_{n_1-1}\left(\tfrac{1}{2}(-1)^{d-1} + \tfrac{1}{2}(-1)^d\right) \\
&+ \sum_{i=1}^{n_1-2} p_i \left(\frac{1}{2} \left(\sum_{j=1}^{n_1-i-2} -q_j^{n_1-i} + q_{n_1-i-1}^{n_1-i}(-1)^d + q_{n_1-i}^{n_1-i}(-1)^{d-1} - r_1^{n_1-i} - r_2^{n_1-i} \right) \right. \\
&\qquad \left. + \frac{1}{2} \left(\sum_{j=1}^{n_1-i-2} -q_j^{n_1-i} + q_{n_1-i-1}^{n_1-i}(-1)^{d-1} + q_{n_1-i}^{n_1-i}(-1)^d + r_1^{n_1-i} - r_2^{n_1-i} \right) \right) \\
&= - \sum_{i=1}^{n_1-2} p_i \left(\sum_{j=1}^{n_1-i-2} q_j^{n_1-i} + r_2^{n_1-i} \right)
\end{aligned}
$$

Again maximizing her payoff, L will make sure to obtain a value of 0. This concludes the second part of the proof.

For the final part, let $n \in \mathbb{N}^d$, $d \geq 3$ be such that $n_k > 1$ for more than one $k \in \{1, \ldots, d\}$. Again, we show that the value is 0. Suppose without loss of generality that $n_1, \ldots, n_c > 1$ and $n_{c+1} = \ldots = n_d = 1$ for some $c \in \{2, \ldots, d\}$. We use induction on c. The base case $c = 2$ is proven above. Hence, consider $c > 2$.

Suppose first that L plays on one of the first c heaps. As a fixed strategy for R in this case, we let him inspect this heap, and empty it if it was not yet so. This leads to a configuration with $c - 1$ either for L or R to move on, resulting in value 0 in any case.

Next, suppose that L plays on one of the last $d - c$ heaps. R can then force a value of 0 by emptying the first heap and moving to a situation with $c - 1$ in this case. $\square$

Note that a very similar result holds for the misère version of the game. Denoting $\text{FSN}^-(n)$ to be the misère version of Fuzzy Schrödinger Nim, we obtain the following.

Theorem 5.4.2. *Let $n \in \mathbb{N}^d$. Then*

$$v(FSN^-(n)) = \begin{cases} -1 & \text{if } \max_{k=1,\dots,d} n_k = 1 \text{ and } d \text{ is odd,} \\ 1 & \text{if } \max_{k=1,\dots,d} n_k = 1 \text{ and } d \text{ is even,} \\ & \text{or if } n_k > 1 \text{ for precisely one } k, \\ 0 & \text{otherwise.} \end{cases}$$

Indeed, in the proof above, the signs of the base cases are flipped. For the explicit strategies described for R in the proof, we can mirror the behaviour for d being odd or even, resulting in the same value of the game as before, being zero. For this misère variant of the game, we find that the values somewhat resemble those for the misère version of the regular game of Nim.

5.5 Kriegspiel Nim

Finally, we consider the Nim variant inspired by Kriegspiel, described in [10]. Recall that, for $n \in \mathbb{N}^d$, we denote this variant of the game with starting position n by $\text{KN}(n)$. We only consider the game with at most two heaps.

For $d = 1$, the game is trivially won by Left, removing all chips on her first turn. For $\text{KN}(n, 1)$, Left removes all but one chip from the first heap on her first turn, winning the game. For the other cases with $d = 2$, we have the following result.

Theorem 5.5.1. *Let $n_1, n_2 \in \mathbb{N}$, $n_1, n_2 \geq 2$. Then $v(KN(n_1, n_2)) = 0$.*

Proof. We give an explicit Nash equilibrium pair, depicted in Fig. 5.10. On the first turn, L reduces the first heap to size 1 or 0, both with probability $\frac{1}{2}$. Next, R tries to empty either heap with probability $\frac{1}{2}$. The attempt to empty the first heap will fail, after which R reduces the second heap to size 1.

Suppose R deviates. Because we assume the strategy of L to be fixed, we may assume R to know that he is either in the state $(1, n_2)$ or $(0, n_2)$, both with probability $\frac{1}{2}$. Hence, picking any move which tries to remove more than 1 chip from the first heap needs not be considered. Now, suppose R tries to pick a single chip from the first heap with probability p, and picks k chips from the second heap with probability q_k, $k = 1, \dots, n_2$.

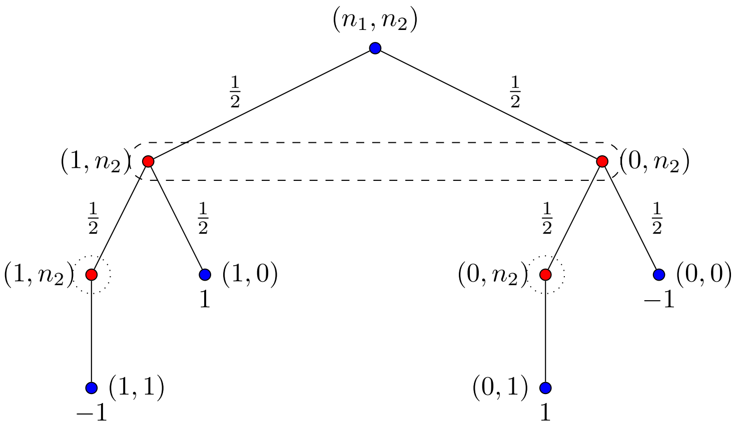


Figure 5.10: A Nash equilibrium for KN(n_1, n_2).

Moving from $(0, n_2)$, trying to move on the first heap will fail, telling R that the heap is empty. Hence, R may proceed to empty the second heap and win. Also, if R removes all n_2 chips from the second heap immediately, he wins. Otherwise, he leaves only chips on the second heap for L, who knows that the first heap is empty. Therefore, L can and will empty the second heap on her next turn and therewith win.

Now, consider the case where R moves from $(1, n_2)$. We define the strategy of L for her next turn(s) as follows: first, L will try to remove n_2 chips from the second heap. If this fails, she will try to remove 1 chip from the second heap. Finally, if this fails, she removes the single chip from the first heap. If and only if R has removed the single chip from the first heap, L will be able to remove n_2 chips from and therewith empty the second heap, winning the game. If R emptied the second heap, L will empty the first, again winning. If R removed all but one chip from the second heap, L is faced with $(1, 1)$ losing in any case. Otherwise, L moves from $(1, n_2 - k)$ to $(1, n_2 - k - 1)$. For $k = n_2 - 2$, this leads R to the state $(1, 1)$, where he loses. For any other $k \in \{1, \ldots, n_2 - 3\}$, R can proceed by reducing the second heap to size 1, giving the state $(1, 1)$ to L, and winning.

Hence, the payoff of the game now becomes

$$\tfrac{1}{2}\left(p - \sum_{k=1}^{n_2-3} q_k + q_{n_2-2} - q_{n_2-1} + q_{n_2}\right) + \tfrac{1}{2}\left(-p + \sum_{k=1}^{n_2-1} q_k - q_{n_2}\right) = q_{n_2-2},$$

so, minimizing, R will choose $q_{n_2-2} = 0$, yielding a value of 0. Hence, deviating does not lead to a better payoff for R.

Next, suppose L deviates. Let L remove k chips from the first heap with probability p_k, $k = 1, \dots, n_1$, or k chips from the second heap with probability q_k, $k = 1, \dots, n_2$. From $(n_1 - k, n_2)$, R will play to either $(n_1 - k, 1)$ or $(n_1 - k, 0)$, both with probability $\frac{1}{2}$. From $(n_1, n_2 - k)$, R will move to $(0, n_2 - k)$ or $(1, n_2 - k)$, again both with probability $\frac{1}{2}$. We will analyse the strategies going from $(n_1 - k, n_2)$; the strategies moving from $(n_1, n_2 - k)$ are symmetrical.

Similar to the discussion above, L knows that she is in either $(n_1 - k, 1)$ or $(n_1 - k, 0)$, both with probability $\frac{1}{2}$. Therefore, we may discard all moves trying to take more than 1 chip from the second heap. Now, let L remove i chips from the first heap with probability $r_i^{n_1-k}$, $i = 1, \dots, n_1 - k$, and let L try to remove the single chip in the second heap with probability s. If this fails, she proceeds to empty the first heap and win.

For R, we define the follow-up strategies as follows. If he emptied the second heap on his first move, he will empty the first heap and win the game on his next turn. If he left a chip in the second heap, he will try to remove it. If this fails, he empties the first heap and wins. If it succeeds, L can proceed by emptying the first heap and winning.

All in all, this leads to the following total payoff, considering only the strategies in which L moves on the first heap in the first turn:

$$\sum_{k=1}^{n_1-2} p_k \left(\frac{1}{2} \left(\sum_{i=1}^{n_1-k-1} r_i^{n_1-k} - r_{n_1-k}^{n_1-k} - s \right) + \frac{1}{2} \left(- \sum_{i=1}^{n_1-k-1} r_i^{n_1-k} + r_{n_1-k}^{n_1-k} + s \right) \right) = 0.$$

Note that we have excluded the cases $k = n_1 - 1$ and $k = n_1$, which both clearly lead to value 0, as well. Moreover, moving on the second heap in any way leads to the same payoff. Hence, deviating gives no larger payoff for L and we are done. □

To conclude, we consider one more variant of Kriegspiel Nim. In this variant, during a player's turn, a move still consists of trying to remove i chips from the j-th heap. If this is possible, it is done; otherwise, as many chips as possible are removed and the heap is left empty. In any case, the turn is passed, the other player not being informed of anything. It can be proven, analogous to the arguments above, that for this variant, the values for up to two heaps coincide with the values of Kriegspiel Nim.

5.6 Conclusions and future work

In this paper, we have considered three non-perfect information variants of the game Nim. For the first variant, Schrödinger Nim, we provided a partial characterization, and a set of structural results. For the second variant, Fuzzy Schrödinger Nim, we provided a complete characterization. For the third variant, Kriegspiel Nim, we gave some preliminary results.

A full solution to the first variant remains elusive, and is a natural direction for further research. One might try to replicate Lemma 5.3.9 with tighter bounds for higher heaps, for example. Moreover, it would be interesting to look at the misère version of this first variant.

The aim of introducing these variants is to generalize the theory of combinatorial games to the class of non-perfect information games. With this aim in mind, the third variant of Kriegspiel Nim is the most widely applicable. Therefore, not only is it interesting to further research this game, but using a similar setup, one might create non-perfect information variants of all combinatorial games. Consider, e.g., the game of Hackenbush. A move will now consist of asking the umpire whether it is possible to remove an edge. If so, it is done; otherwise, the player has to try and remove a different edge. Similarly, in Domineering, a move might be to try to place a stone until the placement is successful. By analysing these non-perfect information variants of different combinatorial games and comparing the results, we hope to distinguish some structure akin to that in the class of combinatorial games.

Chapter 6

Synchronized Cherries

In this chapter, we consider the synchronized version of a variant of the combinatorial game of Cherries, called *Synchronized Stack Cherries*. In this variant, the players may only remove tokens from the front of a segment, instead of from both sides as in regular Cherries. The material in this chapter is based on joint work with Thomas de Mol, started in [18].

We provide a linear-time algorithm to decompose any given Synchronized Stack Cherries position into irreducible stacks and show that these irreducible stacks are fully ordered, allowing for a quick assessment of the winner of a given position and the magnitude of the win. We conjecture that a similar decomposition method can be used to analyze positions in the synchronized version of the regular game of Cherries.

6.1 Introduction

Combinatorial game theory considers the class of two-person, deterministic games with perfect information. Most of the underlying theory can be found in [1], [2] and [3]. Games belonging to this class are, e.g., Nim, Hackenbush, Hex and Domineering. We will focus on variants of Cherries.

In combinatorial games, players move in a sequential fashion, alternating turns until the game's end. In recent years, research has been done on what remains of the theoretical framework if we instead allow the players to move

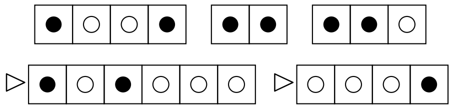


Figure 6.1: Examples of a Cherries position (top) and a Stack Cherries position (bottom).

simultaneously, leading to *synchronized* versions of combinatorial games [13,21]. Every turn, both players communicate their intended move to an impartial referee, who then executes both moves at once.

In this chapter, we study a variant of the combinatorial game of Cherries introduced in Example 2.1.22, which we name Stack Cherries. Recall that a Cherries position consists of ordered connected sequences of tokens, named *cherries*, each colored either black or white. See Figure 6.1 (top) for an example position. In the normal combinatorial game, the two players Left and Right alternate turns, Left removing one black cherry from exactly one sequence on her turn, and Right removing a white cherry on his; only cherries at the beginning or end of a sequence can be removed. A player unable to move on their turn immediately loses, the other player wins. In Stack Cherries, only cherries at the beginning of sequences can be removed, as indicated by a triangle in Figure 6.1 (bottom).

In the synchronized version of the two games, Left and Right pick their cherries simultaneously, after which both are removed at once. Again, if either player is unable to move, the other player immediately wins the game. If all sequences are cleared and therefore neither player can move, the game ends in a draw — which cannot happen for regular Cherries. We call the synchronized version Synchronized Cherries and Synchronized Stack Cherries (or briefly SSC), respectively. Sequences of cherries are usually referred to as *segments* or (in the context of SSC) sometimes as *stacks*; for example, the position in Figure 6.1 (bottom) consists of two stacks.

Though Synchronized Stack Cherries seems simple, it has surprising complexity and expressive power. Our main result, Theorem 6.4.5, is that any SSC segment can be decomposed into a sum of irreducible segments using a linear-time algorithm. Theorem 6.3.9 shows that these irreducible segments are ordered in a very strong sense, allowing us to readily draw conclusions about the winner of the given position, as well as the magnitude of the win.

First, in Section 6.2, we give a full characterization of the combinatorial game of Stack Cherries and provide the necessities concerning synchronized games. We continue by defining the irreducible elements for SSC in Section 6.3 and provide our main decomposition result in Section 6.4. Finally, in Section 6.5, we conclude with a few words on the relation between Synchronized Cherries and SSC, conjecturing that every position of the former is — in a very precise way — equivalent to a position of the latter.

6.2 Basics

We start by providing the full solution to the combinatorial game of Stack Cherries. In a stack of cherries, we say a *block* is a consecutive series of maximal length with cherries of the same color. A block is *non-trivial* if it consists of more than one cherry.

Theorem 6.2.1. *Let G be a Stack Cherries stack starting with $m \geq 1$ black cherries. If G contains a non-trivial block past the first black block, let $c = 1$ if this block is black, and $c = 0$ if it is white. Otherwise, let $c = 1$ if the last cherry of the stack is black, and $c = 0$ if it is white. Then $G = m - 1 + c$.*

The result is readily proven by induction. As an example, the two stacks in Figure 6.1 (bottom) have values 0 and -2, respectively. Notice that all Stack Cherries games are integers, just like regular Cherries games.

We continue by delving slightly into synchronized games. In Section 2.3.1, we defined eight outcome classes for synchronized games. For the remainder of this chapter, we gather all games in the classes $\mathcal{D}$, $\mathcal{LD}$, $\mathcal{RD}$, $\mathcal{LR}$ and $\mathcal{LRD}$ into a single class of games $\mathcal{U}$ for which the outcome of the game is a priori *uncertain*. We order the classes $\mathcal{R} < \mathcal{U} < \mathcal{L}$. In this framework, the following general lemma holds.

Lemma 6.2.2. *Let G be a synchronized game for which $\mathcal{G}^L \neq \emptyset$ and $\mathcal{G}^R = \emptyset$. Then $G > 0$.*

Proof. Let X be an arbitrary synchronized game. We will show that $o(G+X) \geq o(X)$ by induction on the birthday of X.

First, if X has no options at all, then $o(G+X) = o(G) = \mathcal{L} > \mathcal{U} = o(0) = o(X)$. If X has left options, but no right options, then $o(X) = \mathcal{L}$. Moreover, $o(G+X) = \mathcal{L}$, as G also has no right options, so indeed $o(G+X) \geq o(X)$. Conversely, if X only has right options, $o(X) = \mathcal{R}$, so certainly $o(G+X) \geq o(X)$.

Next, suppose X has both left and right options, and assume first that $o(G + X) = \mathcal{R}$. Then, G having no right options, there is some right option X_j^R of X such that $G + X_{ij}^S$ is a Right-player win for all i. By induction, then also $o(G + X_{ij}^S) \geq o(X_{ij}^S)$, so X_{ij}^S is a win for Right for all i, and thus $o(X) = \mathcal{R}$.

Second, suppose $o(X) = \mathcal{L}$. Left winning X, there must be some i such that $o(X_{ij}^S) = \mathcal{L}$ for all j. Then Left can also win $G + X$ by playing to $G + X_{ij}^S$ and continuing locally on X_{ij}^S until Right's options have run out. Hence, $o(G+X) = \mathcal{L} \geq o(X)$.

We conclude that $o(G + X) \geq o(X)$. Now, note that $o(G + 0) = o(G) = \mathcal{L} > \mathcal{U} = o(0)$, so $G \neq 0$. Hence $G > 0$. □

The technique demonstrated in this proof will be used more often throughout this chapter: to show that $G \geq H$ for any two synchronized games, it suffices to show that $o(G+X) = \mathcal{R} \Rightarrow o(H+X) = \mathcal{R}$ and $o(H+X) = \mathcal{L} \Rightarrow o(G+X) = \mathcal{L}$ for all synchronized games X. This type of proof is necessary as, unfortunately, even having introduced the class $\mathcal{U}$, for general synchronized games, $G \geq H$ need not imply $G - H \geq 0$.

Example 6.2.3. Define $G = \boxed{\bullet}$, and consider the somewhat unnatural game $X = \left\{ \boxed{\circ|\circ} \mid \boxed{\bullet|\bullet} \mid \boxed{\circ|\circ} \right\}$. It is clear that $o(X) = \mathcal{L}$. Now, consider $G - G + X$. If Left moves on G, Right can move on X to $\boxed{\circ} + \boxed{\circ|\circ}$, which is won by Right. If Left moves on X, Right can respond on $-G$, playing to $\boxed{\bullet} + \boxed{\circ|\circ}$, which is also won by Right. Hence, Left has no move guaranteeing a win, so $o(G - G + X) \neq \mathcal{L}$. Hence $o(G - G + X) \not\geq o(X)$, so $G - G \not\geq 0$, while obviously $G \geq G$. ◁

6.3 Irreducible segments

We now narrow our focus to the game of Synchronized Stack Cherries, setting out to prove Theorem 6.3.9, that orders the basic building blocks: the irreducible segments. We start by introducing some notation. For any Stack Cherries segment G of length $n \geq 1$, we denote by $G_{i:j}$ the part of the segment only consisting of the cherries $i, i+1, \ldots, j-1, j$; here $1 \leq i \leq j \leq n$. Moreover, we denote by $G_{i:}$ the suffix of the segment consisting of the cherries

$i, i+1, \ldots, n$; here $1 \leq i \leq n$. If $i > n$, $G_{i:}$ denotes the empty segment. Note that $G = G_{1:n} = G_{1:}$.

Previously, we saw that for two synchronized games G and H in general, it is not always easy to determine whether $G \geq H$. For SSC segments, this turns out to be more straightforward. We introduce a *lexicographical order* on the set of SSC segments, denoted by $\preceq$. We define a white cherry to be lexicographically smaller than an empty square, which is in turn smaller than a black cherry. To compare two segments consisting of more than one cherry, we align to the left, adding an empty square at the end of the shorter segment if necessary, and then compare.

Example 6.3.1. Consider

$$G_1 = \rhd\,\boxed{\bullet}\boxed{\circ}\boxed{\circ}\boxed{\bullet}\,, \quad G_2 = \rhd\,\boxed{\bullet}\boxed{\bullet} \quad \text{and} \quad G_3 = \rhd\,\boxed{\bullet}\boxed{\bullet}\boxed{\circ}\,.$$

We have $G_1 \preceq G_2$ and $G_1 \preceq G_3$, as the second cherry of G_1 is white, while the second cherries of G_2 and G_3 are black. Also $G_3 \preceq G_2$, as G_2 has no cherry in the third position, whereas the third cherry of G_3 is white. $\lhd$

It turns out the lexicographic order is in fact the synchronized order as we know it.

Theorem 6.3.2. *For any two SSC segments G and H, it holds that $G \leq H$ if and only if $G \preceq H$.*

Proof. First, if either segment is empty, the result immediately follows from Lemma 6.2.2. If, say, G starts with a white cherry and H with a black cherry, then also $G < 0 < H$ by Lemma 6.2.2. Hence, it remains to prove the statement for two segments starting with a cherry of the same color. Assume without loss of generality that both G and H start with a black cherry.

Let X be an arbitrary synchronized game and consider $G + X$ and $H + X$. We proceed by induction on the birthday of X. Note that, if X has no right options, then neither does $G + X$ nor $H + X$, so that $o(G+X) = o(H+X) = \mathcal{L}$. Hence, suppose that X does have at least one right option. Assume that $G \preceq H$.

First, suppose $o(H + X) = \mathcal{R}$. Right, having no moves on H, but winning $H + X$, must have some move to $H + X_j^R$ so that both $H_{2:} + X_j^R$ and $H + X_{ij}^S$ are Right-player wins for all i. By induction, also $o(G + X_{ij}^S) = \mathcal{R}$ for all i. Moreover, as $G_{2:} \preceq H_{2:}$, also $o(G_{2:} + X_j^R) \leq o(H_{2:} + X_j^R) = \mathcal{R}$. Hence, $G + X$ is won by Right playing to $G + X_j^R$.

Next, suppose $o(G + X) = \mathcal{L}$. If Left wins by playing to $G_{2:} + X$, then $o(G_{2:} + X_j^R) = \mathcal{L}$ for all j, so also $\mathcal{L} = o(G_{2:} + X_j^R) \leq o(H_{2:} + X_j^R)$ for all j, and thus Left wins $H + X$ moving on H. If Left wins $G + X$ by playing to $G + X_i^L$ for some i, then $G + X_{ij}^S$ is a win for Left for all j. Therefore, by induction, $H + X_{ij}^S$ is a Left win for all j, as well, so Left wins $H + X$ playing to $H + X_i^L$. So $G \leq H$.

Conversely, if $G \leq H$, we know in particular that $o(G + X) \leq o(H + X)$ where X is the longest common prefix of G and H with black and white cherries toggled, say of length $\ell \geq 1$. Playing in these two games leads to comparison of the $(\ell + 1)$-st cherries of G and H (perhaps even an empty square), showing that $G \preceq H$. □

Using the (lexicographic) order on SSC segments, determining the best move for either player is relatively straightforward.

Theorem 6.3.3. *Let G and H be SSC segments that start with a cherry of the same color. If $G \leq H$, then $G_{2:} + H \leq G + H_{2:}$.*

We omit the laborious proof, which can be found in [18]. This theorem gives us a deterministic optimal strategy for both players: Left repeatedly plays on the greatest segment (starting with a black cherry), whereas Right takes a cherry from the smallest segment (starting with a white cherry). Hence, the outcome of a game of Synchronized Stack Cherries is always deterministic under optimal play, and any position in $\mathcal{U}$ always ends in a draw.

However, though this observation does provide us with a way to determine the winner and therewith outcome class of a position, it does not give us an efficient way to compare positions, nor a way to measure the magnitude of a position. We continue by developing these concepts in more detail.

As illustrated previously, many beautiful results for combinatorial games do not readily carry over to synchronized games. However, by somewhat relaxing our definition of equality, we do pave the way for fundamental results more in line with combinatorial game theory.

Definition 6.3.4. For two SSC positions G and H, we say $G =_{\text{SC}} H$ if $o(G + X) = o(H + X)$ for all SSC positions X. Similarly, we use $\leq_{\text{SC}}$, $<_{\text{SC}}$, $\geq_{\text{SC}}$ and $>_{\text{SC}}$.

Lemma 6.3.5. *Let G be a SSC segment. Then $G - G =_{\text{SC}} 0$.*

Proof. Assume without loss of generality that G starts with a black cherry. Let X be an arbitrary SSC position, and proceed by induction on the birthdays of

G and X. First, suppose $o(G - G + X) = \mathcal{L}$. If Left can win by moving on G, then in particular $o(G_{2:} - G_{2:} + X) = \mathcal{L}$, so by induction also $o(X) = \mathcal{L}$. If Left wins by moving on X, then there is some i such that $o(G - G + X^S_{ij}) = \mathcal{L}$ for all j. Hence, by induction, also $o(X^S_{ij}) = \mathcal{L}$ for all j, so X is also won by Left.

Next, suppose $o(X) = \mathcal{L}$, and let X^L_i be a winning move for Left. Consider $G - G + X$. If the (lexicographically) smallest segment starting with a white cherry in this game is located in X, then moving on this segment to X^R_j is the dominating move for Right by Theorem 6.3.3. The result for Left also moving on X is $G - G + X^S_{ij}$, which is a win for Left by induction — hence, $o(G - G + X) = \mathcal{L}$. If the (lexicographically) smallest segment starting with a white cherry in $G - G + X$ is $-G$, then this is the dominating move for Right. Left can win by playing on G to $G_{2:} - G_{2:} + X$ by induction, so also in this case $G - G + X$ is a win for Left. Note that the latter case includes the case of G being part of X.

Hence, $o(G - G + X) = \mathcal{L}$ if and only if $o(X) = \mathcal{L}$. An analogous argument works for the case in which X is a Right-player win, from which the conclusion follows. □

Through the (lexicographical) order, all SSC segments are comparable. However, some segments turn out to be much smaller than others, similar to the infinitesimals in combinatorial game theory and the value ε described in [21]. To show this, we need the notion of irreducible elements.

First, note that, by Lemma 6.2.2, for all non-empty SSC segments G, we find either $G > 0$ or $G < 0$, depending on whether G starts with a black or white cherry. Hence, it makes sense to define

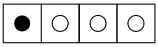
$$|G| = \begin{cases} G & \text{if } G \geq 0, \\ -G & \text{if } G < 0. \end{cases}$$

If $|G| > |H|$ for two SSC segments G and H, we say G is (*lexicographically*) *stronger* than H, and H is *weaker* than G.

Definition 6.3.6. Let G be a SSC segment of length $n \geq 1$. We call G an *irreducible element* if $|G| < |G_{i:}|$ for all $1 < i \leq n$.

In words, an irreducible G is weaker than all its suffixes, apart from itself.

Example 6.3.7. The five positive irreducible elements of length at most 4 are, in decreasing order, ▷[●], ▷[●|○], ▷[●|○|○|●], ▷[●|○|○] and

▷  . ◁

It turns out that all different irreducible elements are infinitesimal with respect to each other. We need the following lemma, the proof of which is based on a direct comparison of the segments in question and can be found in [18].

Lemma 6.3.8. *Let G be an irreducible element of length n. Then $|G_{i:}| \geq |G_{1:i-1}|$ for all $1 < i \leq n$.*

Theorem 6.3.9. *Let G and H be irreducible elements such that $G > H > 0$. Then $G >_{\mathrm{SC}} k \cdot H$ for all $k \in \mathbb{N}$.*

Proof. By Lemma 6.3.5, it suffices to prove $G - k \cdot H >_{\mathrm{SC}} 0$. We proceed by induction on k, the base case being true by assumption. Let X be an arbitrary SSC position and consider $G - k \cdot H + X$.

First, suppose that $-H$ is the lexicographically strongest segment starting with a white cherry, which Right will remove. Left plays on G, resulting in $G_{2:} - H_{2:} - (k-1) \cdot H + X$. Note that, as $G - H > 0$, at least one of $G_{2:}$ and $-H_{2:}$ starts with a black cherry. If the other starts with a white cherry, by the assumption of G and H being irreducible elements, this is the lexicographically strongest segment starting with a white cherry. Hence, optimal play for Right dictates the removal of this white cherry, and Left may respond on the other of the two segments. Play continues in this fashion until we arrive at $G_{i:} - H_{i:} - (k-1) \cdot H + X$ for some i, where both $G_{i:}$ and $-H_{i:}$ start with a black cherry, or one of them is empty.

If $G_{i:}$ starts with a black cherry, then $G_{i:} - H_{i:} \geq G_{i:} > G$, since G is an irreducible element. If not, then $G_{i:}$ is the empty segment, and, by the mirrored play so far, we find $G = H_{1:i-1}$. By Lemma 6.3.8, $|H_{i:}| \geq |H_{1:i-1}|$, so $G_{i:} - H_{i:} = 0 + |H_{i:}| \geq |H_{1:i-1}| = |G| = G$. Hence, $G_{i:} - H_{i:} \geq G$ in any case, so

$$G_{i:} - H_{i:} - (k-1) \cdot H + X \geq G - (k-1) \cdot H + X > X$$

by induction. Right having played optimally, we find $o(G - k \cdot H + X) \geq o(X)$.

Second, suppose that the lexicographically strongest segment starting with a white cherry is contained in X, and that removing this cherry leads to X_j^R. First, assume that $o(X) = \mathcal{L}$, Left winning by moving to X_i^L. By induction on the birthday of X, we conclude $o(G - k \cdot H + X_{ij}^S) \geq o(X_{ij}^S) = \mathcal{L}$. Hence, Right moving optimally, $G - k \cdot H + X$ is a Left-player win. Next, suppose $o(G - k \cdot H + X) = \mathcal{R}$, the winning move for Right being to $G - k \cdot H + X_j^R$. It follows that $\mathcal{R} = o(G - k \cdot H + X^S_{ij}) \geq o(X^S_{ij})$ for all i by induction, so $o(X) = \mathcal{R}$. Hence $o(G - k \cdot H + X) \geq o(X)$.

We conclude that $G - k \cdot H \geq_{\text{SC}} 0$. Now, noting that $o(G - k \cdot H) \geq o(G - (k-1) \cdot H)$ by the reasoning above, and that $o(G) = \mathcal{L}$, we conclude by induction that $o(G - k \cdot H) = \mathcal{L}$, whereas $o(0) = \mathcal{U}$. Hence, strict equality does not hold, and the result follows. □

Example 6.3.10. Consider the irreducible elements G_1 and G_2 defined by $G_1 =$ ▷[●] and $G_2 =$ ▷[●|○]. Then $G_1 > G_2$, and by Theorem 6.3.9, $G_1 >_{\text{SC}} k \cdot G_2$ for any $k \in \mathbb{N}$. ◁

Before continuing with the decomposition of SSC segments into irreducible elements, we make a small detour, providing a method to count the number of unique irreducible elements of a given length. To do so, we consider words over the alphabet $\{0, 1\}$. We use the lexicographical order $\preceq$ as defined before for Stack Cherries, writing $0 \prec \varepsilon \prec 1$, denoting the empty word by ε. We denote the canonical lexicographical order by $\trianglelefteq$, writing $\varepsilon \triangleleft 0 \triangleleft 1$. We define $\overline{0} = 1$ and $\overline{1} = 0$, and naturally extend this definition to all words in $\{0, 1\}^*$. For a word $w = w_1 w_2 \dots w_n$, we write $\rho_k(w) = w_{k+1} w_{k+2} \dots w_n w_1 \dots w_k$ for the counterclockwise rotation over k positions. We say that the *fundamental period* of w equals p if $\rho_p(w) = w$, and $\rho_q(w) \neq w$ for all $1 \leq q < p$. Finally, we write $|w| = w$ if $w_1 = 1$, and $|w| = \overline{w}$ otherwise.

Theorem 6.3.11. *Let G be a SSC segment starting with a black cherry and represent G by a word $w \in \{0, 1\}^*$, writing a 1 for a black cherry and 0 for a white cherry. Then G is irreducible if and only if w has fundamental period n and $\rho_1(w)$ is the lexicographically smallest element of $S(w) = \bigcup_{k=1}^{n} \{\rho_k(w), \rho_k(\overline{w})\}$ under the canonical lexicographical order $\trianglelefteq$.*

Proof. Suppose G is irreducible. First, suppose w has fundamental period $k < n$. Note that $k \mid n$; hence, $G = HH \dots H$ $(n/k$ times) for some suffix H of length k which starts with a black cherry. But then $|H| = H < G = |G|$, which contradicts the fact that G is irreducible. Hence, w indeed has fundamental period n.

Next, suppose $\rho_1(w)$ is not the lexicographically smallest element of $S(w)$. First, suppose that $w' = \rho_k(w)$ is the lexicographically smallest element of $S(w)$ for some $k \neq 1$. Naturally, $w'_1 = 0$, as the word consisting of only 1's is not irreducible. Moreover, $w'_n = 1$, as otherwise $\rho^{-1}(w')$ is lexicographically smaller than w'. Now, consider the suffix $z = w_k w_{k+1} \dots w_n$ of w, of which all but w_k is a prefix of w'. Note that $w_k = w'_n = 1$. By the assumption of $w' \trianglelefteq \rho_1(w)$, at the first element at which the words differ, the element of w' must be smaller than the element of $\rho_1(w)$. If this element is one of the first $n-k$ of w', then $z < w$, which is a contradiction. Otherwise, if the first $n-k$ elements match, we find the $n-k+1$-st element of w' to be $w_1 = 1$. For $w' \trianglelefteq \rho_1(w)$ to hold, then also $w_{n-k+2} = 1$. But then $z \prec w$. Hence, $w' \triangleleft \rho_1(w)$ cannot hold.

Next, suppose that $w' = \rho_k(\overline{w})$ is the lexicographically smallest element of $S(w)$ for some k. By the reasoning above, $w'_1 = 0$ and $w'_n = 1$. Now, first suppose that the first $n-k$ elements of w' are not all 0's. Let ℓ be the position of the first 1 in w' and consider the prefix $w'_1 \dots w'_{\ell-1}$ of w'. Then the same argument as above leads to a contradiction, taking as suffix $z = w_{n-\ell+1} \dots w_n$ of w. If the first $n-k$ elements of w' are 0's, then the last $n-k$ elements of $\rho_1(w)$ are 1's. Note that $n-k > \frac{n}{2}$ must hold; otherwise, a rotation of w would be smaller than this rotation of $\overline{w}$. But then for the suffix $z = w_{k+1} \dots w_n$ of w be find $|z| < |w| = w$. Hence, $\rho_1(w)$ must indeed be the lexicographically smallest element of $S(w)$.

Now, suppose w has fundamental period n and $\rho_1(w)$ is the lexicographically smallest element of $S(w)$. For the sake of contradiction, suppose that $w > |z|$ for some suffix z of w of length k (note that equality cannot occur for words of different length). First, suppose $|z| = z$. If the first ℓ elements of w and z are equal, and $w_{\ell+1} > z_{\ell+1}$, then we may rotate w such that it starts with $z_2 \dots z_{\ell+1}$, finding a smaller rotation than $\rho_1(w)$. Hence, all elements of z and w must be equal, forcing $w_{k+1} = 1$. But then we can rotate w such that it starts with w_{k+1}, which provides a smaller rotation, as the first $k-1$ elements remain the same, and the k-th is now a 0 instead of a 1.

Next, suppose $|z| = \overline{z}$. We repeat the above argument, now swapping the colors of w before rotating, again leading to smaller elements of $S(w)$. Hence, w is indeed irreducible. □

The words over $\{0, 1\}$ as described before are usually called *necklaces* in the context of equivalence under rotation [43].

Corollary 6.3.12. *For a fixed n, the number of irreducible elements of length n starting with a black cherry equals the number of binary necklaces of length n having fundamental period n, in which the colors may be swapped.*

The number of such necklaces, and therewith the number of positive irreducible elements for any fixed length, may be found as A000048 in the OEIS [44].

6.4 Decomposition

If a SSC position consists of only irreducible elements, it is straightforward to find the winner and its value. Hence, we develop a method to decompose any given SSC position into irreducible elements. Again, the omitted proof of Lemma 6.4.1 can be found in [18].

Lemma 6.4.1. *Let G be a SSC segment of length n that is not an irreducible element. Let $G_{i+1:}$ be the weakest suffix of G, and let $0 \leq j < i$ be such that $|G_{j+1:i}| \leq |G_{k+1:i}|$ for all $0 \leq k < i$. Then $|G_{i+1:}| \leq |G_{j+1:i}|$. If the $(i+1)$-st and $(j+1)$-st cherries of G have a different color, the inequality is strict.*

Theorem 6.4.2. *Let G be a SSC segment of length n and let $0 \leq i < n$ be such that $|G_{i+1:}| \leq |G_{j+1:}|$ for all $0 \leq j < n$. Then $G = G_{1:i} + G_{i+1:}$.*

Proof. Assume without loss of generality that the first cherry of G is black. For $i = 0$, there is nothing to show, so let $i > 0$, and consider $G_{i+1:}$.

First, suppose $G_{i+1:}$ also starts with a black cherry. By assumption, $G_{i+1:} = |G_{i+1:}| \leq |G_{0+1:}| = G$, so $G - G_{i+1:}$ is a Left-player win, as the segments differ in length. If G and $G_{i+1:}$ do not match for the first i cherries, then play on $G - G_{i+1:}$ will halt with less than i cherries having been taken from G, so $G_{1:i} - G_{i+1:} > 0$ must hold.

If G and $G_{i+1:}$ do match for the first i cherries, play continues to $G_{i+1:} - G_{2i+1} > 0$. Now, if the $i+1$-st cherry of G and $G_{i+1:}$ would match, being the first cherry of $G_{i+1:}$, these cherries should be black. However, it would follow that $|G_{i+1:}| = G_{i+1:} > G_{2i+1:} = |G_{2i+1:}|$, which is a contradiction. Hence, if G and $G_{i+1:}$ match for the first i cherries, they must mismatch at the $i+1$-st. This cherry in $G_{i+1:}$ being white or non-existent, we conclude that $G_{1:i} > G_{i+1:}$ in both cases.

Hence, in any context, we find that, in $G_{1:i} + G_{i+1:} + X$, Left's move on $G_{i+1:}$ is dominated. Therefore, by induction,

$$G_{1:i} + G_{i+1:} = \{G_{2:i} + G_{i+1:} \mid\mid\} = \{G_{2:} \mid\mid\} = G.$$

Second, suppose $G_{i+1:}$ starts with a white cherry. By induction, using the statement of Lemma 6.4.1, we can obtain a collection of irreducible elements $B_1, \ldots, B_k$ such that $B_1 > 0$, $B_1 \geq \ldots \geq B_k$ and $G_{1:i} = B_1 + \ldots + B_k$. Let X be an arbitrary SSC position, and consider $B_1 + \ldots + B_k + G_{i+1:} + X$. If the best move for Right is on some B_j, then $W \geq B_j$ for all other segments W starting with a white cherry, as well as $G_{i+1:} \geq B_j$ Furthermore, for all segments B starting with a black cherry, we have $B > 0 > B_j$. Hence, $B_1 + \ldots + B_k + G_{i+1:} + X \geq B_1 + \ell \cdot B_i$ for some ℓ. By Lemma 6.4.1, $|B_1| > |B_j|$, as the stacks start with different colors. We conclude that, both being irreducible elements,

$$B_1 + \ldots + B_k + G_{i+1:} + X \geq B_1 + \ell \cdot B_i > -\ell \cdot B_j + \ell \cdot B_j = 0.$$

Hence, if Right's best move on $B_1 + \ldots + B_k + G_{i+1:} + X$ is to play on some B_j, then the game is won by Left. Hence, leaving out the possible moves for Right on these stacks cannot worsen the outcome for Right, being $\mathcal{L}$. Furthermore, deleting possible moves for Right never worsens the outcome for Left. Hence, we may delete all moves on the B_j for Right without changing the outcome class in any context. We conclude that

$$\begin{aligned} G_{1:i} + G_{i+1:} &= B_1 + \ldots + B_k + G_{i+1:} \\ &= \{(B_1)_{2:} + \ldots + B_k + G_{i+1:} \mid\mid \} \\ &= \{G_{2:} \mid\mid \} \\ &= G. \end{aligned}$$

□

By repeatedly applying Theorem 6.4.2, one may decompose any given SSC segment into a sum of irreducible elements, every pair of different elements being infinitesimal with respect to each other by Theorem 6.3.9. Compare this to the uptimal notation for dicotic games as described in [2, Chapter 9].

Example 6.4.3. Consider the SSC segment

$$G = \triangleright\boxed{\bullet\,|\,\bullet\,|\,\circ\,|\,\bullet\,|\,\bullet\,|\,\circ\,|\,\circ}$$

Comparing all non-empty suffixes, we find that $G_{5:} = \triangleright\boxed{\bullet\,|\,\circ\,|\,\circ}$ is the weakest. Hence, in particular, $|G_{5:}| \leq |G_{6:}|$ and $|G_{5:}| \leq |G_{7:}|$, so

$$G = \triangleright\boxed{\bullet\,|\,\bullet\,|\,\circ\,|\,\bullet} + \triangleright\boxed{\bullet\,|\,\circ\,|\,\circ}$$

by Theorem 6.4.2. Now, comparing all suffixes of $\triangleright\boxed{\bullet\,|\,\bullet\,|\,\circ\,|\,\bullet}$, we find that $\triangleright\boxed{\circ\,|\,\bullet}$ is the weakest, leading to

$$G = \triangleright\boxed{\bullet\,|\,\bullet} + \triangleright\boxed{\circ\,|\,\bullet} + \triangleright\boxed{\bullet\,|\,\circ\,|\,\circ}$$

Again looking at the suffixes of $\triangleright\boxed{\bullet\,|\,\bullet}$, we conclude that this segment can be split once more, producing

$$G = \triangleright\boxed{\bullet} + \triangleright\boxed{\bullet} + \triangleright\boxed{\circ\,|\,\bullet} + \triangleright\boxed{\bullet\,|\,\circ\,|\,\circ}$$

Note that indeed every segment of the decomposition is an irreducible element. ◁

Using Theorem 6.4.2 to decompose a given SSC segment requires continuously comparing the lexicographic strength of suffixes, leading to an $O(n^2)$ algorithm to determine the decomposition of a segment of length n. This can be sped up. We once again omit the proof of Lemma 6.4.4 and refer to [18].

Lemma 6.4.4. *Let G be a SSC segment of length n and let $G_{i+1:}$ be its weakest suffix. Then, for all $1 \leq \ell, k \leq i$, it holds that $|G_{\ell:}| > |G_{k:}|$ if and only if $|G_{\ell:i}| > |G_{k:i}|$.*

Theorem 6.4.5. *Let G be a non-irreducible SSC segment of length n and let $1 \leq i < n$ be the smallest index for which $|G_{i+1:}| \leq |G|$. Then $G = G_{1:i} + G_{i+1:}$.*

Proof. Let $G_{k+1:}$ be the weakest suffix of G. Suppose first that $G_{1:k}$ is an irreducible element, i.e., $|G_{j:k}| > |G_{1:k}|$ for all $1 \leq j \leq k$. By Lemma 6.4.4, then also $|G_{j:}| > |G_{1:}| = |G|$ for all $1 \leq j \leq k$, so $i \geq k+1$. As $G_{k+1:}$ is the weakest suffix of G, also $i \leq k+1$, so we conclude that $i = k+1$. Hence, by Theorem 6.4.2, $G = G_{1:i} + G_{i+1:}$.

Next, suppose $G_{1:k}$ is not an irreducible element. It still holds that $i \leq k+1$, as $G_{k+1:}$ is the weakest suffix of G. If $i = k+1$, we are done by Theorem 6.4.2, so suppose that $i < k+1$. By Lemma 6.4.4, i is the smallest index for which $|G_{i+1:k}| \leq |G_{1:k}|$. By induction on the birthday, it follows that $G_{1:k} = G_{1:i} + G_{i+1:k}$. As $G_{k+1:}$ is the weakest suffix of G, it is also the weakest suffix of $G_{i+1:}$. Hence, by Theorem 6.4.2, $G_{i+1:} = G_{i+1:k} + G_{k+1:}$. We conclude that

$$G = G_{1:k} + G_{k+1:} = G_{1:i} + G_{i+1:k} + G_{k+1:} = G_{1:i} + G_{i+1:}.$$

□

Theorem 6.4.5 provides us with a faster way of decomposing a given SSC segment into a sum of irreducible elements. By using *suffix arrays* to store information about the suffixes of a given SSC segment, we can develop a linear-time algorithm to decompose the given segment into irreducible elements [45–47].

6.5 Summary and future work

In this chapter, we have considered a variant of the combinatorial game of Cherries, called Synchronized Stack Cherries. In this variant, players are only allowed to take a cherry from the front of a given segment or stack. In the combinatorial version of the game, all encountered positions are integers, readily characterized by an inductive argument.

Subsequently, we analysed the synchronized version of the game, in which the two players make moves simultaneously. For this game of Synchronized Stack Cherries, we showed that any segment can be decomposed into a sum of irreducible elements in linear time using Theorem 6.4.5. These irreducible elements are all infinitesimal with respect to each other by Theorem 6.3.9. Analogous to the uptimal notation for dicotic combinatorial games, just looking at the sign of the largest irreducible element in the decomposition immediately tells us the outcome class of a game.

We conjecture that it is possible to decompose a given segment of Synchronized Cherries into a sum of Synchronized Stack Cherries irreducible elements in a similar way. We repeatedly start at the lexicographically strongest end of the segment, taking cherries until the other side becomes stronger. Every group of cherries taken in this way forms a new Stack Cherries segment. More details can be found in [18]. Using this decomposition, it would be possible to determine the outcome class and value of a Synchronized Cherries position in quadratic or perhaps even linear time.

Chapter 7

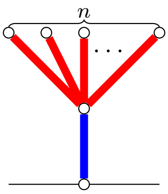
Synchronized Hackenbush

In this chapter, we study the Nash synchronized version of Hackenbush. Most of the chapter will be devoted to Red-Blue Hackenbush, which is a separable ruleset and therefore allows for an intuitive synchronized version. In Section 7.1, we start out by defining some of the positions of interest, as well as the value function which we will use. In Sections 7.2 and 7.3, we continue by computing the Nash values for a series of positions, proving Conjecture 2.3.34 for these positions. In Section 7.4, we continue by looking at one of these positions in more detail. Finally, in Section 7.5, we consider the non-separable game of Red-Blue-Green Hackenbush, and showcase some of the problems encountered in finding and analyzing a synchronized version of this game.

7.1 Introduction

Throughout this chapter, much of the discussion will be on positions defined as Hackenbush *flowers* in [1,2]. It is convenient to introduce a shorthand for these games.

Definition 7.1.1. For $n \in \mathbb{N}$, we define H_n to be the Hackenbush position consisting of one blue edge with n red edges on top, that is,

$H_n =$ 

The game H_n is also called a *flower* with n leaves. We write $H = H_1$ for short.

As mentioned in Section 2.3.3, there is only one value function that satisfies Definition 2.3.24, assigning a value of 0 to the empty game, and n or $-n$ to a position consisting only of n blue or red edges, respectively. In the sequel, we write $v(G)$ for the Nash value of a Hackenbush game stemming from this value function. Moreover, for decided games, we tend to write the value of the game also when speaking of the game itself. For example, we write $1 = \{0 \mid \mid \}$.

Example 7.1.2. For a single copy of H as defined above, we find $v(H) = 0$, as both players take their only edge in the first turn, resulting in the empty game, being a draw. For two copies of H, denoted $H + H$ or $2H$, the resulting zero-sum game is described by the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Here, the first row and column correspond to the players playing on the one copy of H, while the second row and column correspond to the other copy. If the players play on the same copy, the result is H, having value 0 as argued before. If the players play on a different copy, the result is a single blue stalk, having value 1. Hence, the value of $2H$ is that of the Nash equilibrium of this game, being $\frac{1}{2}$. ◁

The following corollary of Theorem 2.3.32 is useful in analyzing synchronized Hackenbush positions.

Lemma 7.1.3. *In synchronized Hackenbush, a move that removes at least two edges of the own color is dominated.*

Proof. If all edges in G have the same color, the result is trivial. Hence, let G be a synchronized Hackenbush game, let G^L be a move which removes at least two blue edges, and let G^R be an arbitrary move for Right. Let $G^{L'}$ be a move which removes a strict subset of the blue edges removed by G^L; this must be possible, as at least one of the edges removed in G^L must lie on a path from the ground to this edge containing at least one more blue edge. Then either $G^{L+R} = G^{L'+R}$ if all blue edges removed in G^L are also removed by G^R; or G^{L+R} is attainable from $G^{L'+R}$ through one move of Left. Hence, either $v(G^{L+R}) = v(G^{L'+R})$, or $v(G^{L+R}) \leq v(G^{L'+R})$ by Theorem 2.3.32. □

We prove Conjecture 2.3.33 for a special case, which we will encounter repeatedly in the rest of this chapter.

Theorem 7.1.4. *Let G be a synchronized Red-Blue Hackenbush tree, and let T be a terminal Red-Blue Hackenbush position. Then $v(G+T) = v(G) + v(T)$.*

Proof. If G is also terminal, the result is trivial. Without loss of generality, suppose that $v(T) > 0$, in which case we may represent T by a collection of blue edges. Consider a move G^L that, by Lemma 7.1.3, we may assume to remove no more than one blue edge. Let G^R be arbitrary, and consider $G_1 = G^{L+R} + T$ and $G_2 = G^R + H^L = G^R + T - 1$.

We couple G_1 and G_2. On G_2, let Left play her Nash equilibrium strategy, which she copies on G_1. If she would remove the one blue edge which is missing in G_1, she plays on T instead. Conversely, Right plays his Nash equilibrium strategy on G_1, and copies his moves on G_2, which is always possible. We denote the values thus obtained by $v^L(G_1)$ and $v^R(G_2)$, respectively.

Playing like this, the games unfold, reaching G_1' and G_2' for which $G_1' = G_2'$. Hence, we find $v^L(G_1) = v^R(G_2)$. As $v(G_1) \geq v^L(G_1)$ and $v^R(G_2) \geq v(G_2)$, the statement follows. □

7.2 Flowers

We start by considering n copies of the simplest flower H, which we will denote by nH.

Theorem 7.2.1. *Let $v_n = v(nH)$. Then v_n satisfies the following recurrence relation:*

$$\begin{cases} v_n = \frac{1}{n}\left((n-1)(1+v_{n-2}) + v_{n-1}\right), & n \geq 3, \\ v_1 = 0, v_2 = \frac{1}{2}. \end{cases}$$

Proof. As computed in Example 7.1.2, we find $v_1 = v(H) = 0$ and $v_2 = v(2H) = \frac{1}{2}$. Now, consider the game nH. Each player chooses one of the n stalks to play on. If both players choose the same stalk, we end up with the game $(n-1)H$, having value v_{n-1}. If the players choose a different stalk, we continue play in $1 + (n-2)H$, having value $1 + v_{n-2}$. Hence, gathering these values in an outcome matrix, we find v_{n-1} on the diagonal, and $1 + v_{n-2}$ elsewhere. The resulting linear program to be solved is thus symmetric in all variables, so by Theorem 2.2.20, picking any stalk with equal probability for both players is a Nash equilibrium. The resulting value is $\frac{1}{n}v_{n-1} + \frac{n-1}{n}(1 + v_{n-2})$, as required. □

This recurrence relation is not easy to solve. However, the difference $v_n - v_{n-1}$ turns out to be well-behaved.

Theorem 7.2.2. *Define $d_n = v_n - v_{n-1}$. Then d_n satisfies the following recurrence relation:*

$$\begin{cases} d_n = \frac{n-1}{n}(1 - d_{n-1}), & n \geq 3, \\ d_2 = \frac{1}{2}. \end{cases}$$

The solution of this equation is given by

$$d_n = \frac{2n + (-1)^n - 1}{4n}.$$

Proof. By Theorem 7.2.1, we have $d_2 = v_2 - v_1 = \frac{1}{2}$. For the recurrence, we rewrite using the recurrence for v_n:

$$\begin{aligned} d_n &= v_n - v_{n-1} \\ &= \tfrac{n-1}{n}(1 + v_{n-2}) + \tfrac{1}{n}v_{n-1} - v_{n-1} \\ &= \tfrac{n-1}{n} + \tfrac{n-1}{n}v_{n-2} + \tfrac{1-n}{n}v_{n-1} \\ &= \tfrac{n-1}{n} - \tfrac{n-1}{n}d_{n-1} \\ &= \tfrac{n-1}{n}(1 - d_{n-1}). \end{aligned}$$

The solution may be verified by substituting it into the proven recurrence. □

Note that for $n = 2k$ even, we have that

$$d_n = \frac{4k + (-1)^{2k} - 1}{8k} = \frac{4k + 1 - 1}{8k} = \frac{1}{2}.$$

For $n = 2k+1$ odd, we find

$$d_n = \frac{2(2k+1)+(-1)^{2k+1}-1}{4(2k+1)} = \frac{2(2k+1)-2}{4(2k+1)} = \frac{2n-2}{4n} = \frac{n-1}{2n} = \frac{1}{2} - \frac{1}{2n}.$$

We thus have the following result.

Corollary 7.2.3. *We have*

$$\lim_{n\to\infty} d_n = \frac{1}{2}.$$

Hence, for large values of n, the Nash equilibrium value $v(nH)$ of the synchronized Hackenbush position nH converges to the combinatorial value of the regular Hackenbush position nH, as conjectured in Conjecture 2.3.34.

Using the above solutions for d_n also provides us with a solution for v_n itself.

Theorem 7.2.4. *We have*

$$v_n = \left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{1}{2} + \sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{k}{2k+1}.$$

Writing $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ for the digamma function, we conclude that

$$v_n = \frac{1}{4}\left(2n - 2 - \psi\left(\frac{n+1+\mathbb{1}_{\{n \text{ odd}\}}}{2}\right) + \psi\left(\frac{3}{2}\right)\right).$$

Proof. We rewrite

$$\begin{aligned} v_n &= v_1 + \sum_{k=2}^{n}(v_k - v_{k-1}) \\ &= 0 + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor}(v_{2k} - v_{2k-1}) + \sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor}(v_{2k+1} - v_{2k}) \\ &= \left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{1}{2} + \sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{2k}{2(2k+1)} \\ &- \left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{1}{2} + \sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{k}{2k+1}. \end{aligned}$$

The conclusion in terms of the digamma function may be computed using computational tools. □

We continue by considering games consisting of flowers, i.e., copies of H_n. Let us start by examining kH_2 for some small k.

Example 7.2.5. It is obvious that H_2 ends in a draw: Left's first move removes all edges. Hence, $v(H_2) = 0$. The game $2H_2$ proceeds to either H_2 if both players choose to remove an edge from the same flower, or to H if the players choose different flowers. In any case, the result is a draw, so also $v(2H_2) = 0$.

Finally, consider $3H_2$. If the players pick the same flower to move on, we arrive at $2H_2$, ending in a draw. If the players pick different flowers, we continue to $H + H_2$. From here, if both players play on H, we arrive at H_2. If both play on H_2, we continue to H. If Left plays on H and Right on H_2, we end up in H, as well. Finally, if Left plays on H_2 and Right on H, we arrive at 1. Hence, Right will always play on H_2, forcing a draw also in this case. So $v(3H_2) = 0$, as well. $\triangleleft$

In this example, we see that the optimal strategy for Right is to always play on a flower with the largest number of petals. For Left, keeping to this strategy is also optimal. Both players pick uniformly if there is a choice between multiple identical copies of a flower. In general, this is always a Nash equilibrium pair of strategies for the players, regardless of the number and size of the flowers.

Theorem 7.2.6. *Let $n \in \mathbb{N}$, $a_1, \ldots, a_{n-1} \in \mathbb{N}_0$, $a_n \in \mathbb{N}$ and consider the game $G = \sum_{k=1}^{n} a_k H_k$. Left cutting her edge in any copy of H_n uniformly at random and Right cutting any edge in any copy of H_n uniformly at random is a Nash equilibrium for G.*

We split the proof into proving that the strategies described are indeed best responses to each other.

Definition 7.2.7. Let $G = \sum_{k=1}^{n} a_k H_k$ as in Theorem 7.2.6. We define

$$v(a_1, \ldots, a_n) = v(G).$$

Lemma 7.2.8. *Let $G = \sum_{k=1}^{n} a_k H_k$ as before. Suppose Right employs the strategy of cutting any edge in any copy of H_n uniformly at random. Then a best response for Left is to cut an edge in any copy of H_n uniformly at random.*

Proof. Suppose Left also plays on a copy of H_n uniformly at random. If both players pick the same copy of H_n, we end up in the game with one H_n removed, having value $v(a_1, \ldots, a_{n-1}, a_n - 1)$. If the players pick a different copy, the game has two copies of H_n less, and one copy of H_{n-1} more, giving value $v(a_1, \ldots, a_{n-1}+1, a_n-2)$.

Now, if Left plays on any copy of H_k, if $k < n$, then this will result in a value of $v(a_1, \ldots, a_k - 1, \ldots, a_{n-1}+1, a_n-1)$; or, otherwise, if $k = n-1$, a value of $v(a_1, \ldots, a_{n-1}, a_n - 1)$. We will show that these results are not preferable to the results obtained when playing on the H_n uniformly at random.

First, note that $v(a_1, \ldots, a_k - 1, \ldots, a_{n-1}+1, a_n-1) \leq v(a_1, \ldots, a_{n-1}, a_n-1)$, as in the game with the latter value, we simply have $n-1-k$ fewer red edges on one of the flowers, which does not change the game in Right's favor. Similarly, we have that $v(a_1, \ldots, a_k - 1, \ldots, a_{n-1}+1, a_n-1) \leq v(a_1, \ldots, a_{n-1}+1, a_n - 2)$. Hence, picking a copy of H_k with $k < n-1$ is not better for Left than any outcome which can be obtained by picking a copy of H_n. Finally, also $v(a_1, \ldots, a_{n-1}, a_n-1) \leq v(a_1, \ldots, a_{n-1}+1, a_n-2)$ for the same reason, so also picking a copy of H_{n-1} is not preferable for Left.

Hence, the best response for Left is to play only on copies of H_n. By the last line of the previous argument, we see that picking the same copy of H_n as Right is preferable over picking a different copy. To maximize the probability of picking the same copy, picking a copy uniformly at random suffices for Left. ☐

Lemma 7.2.9. *Let $G = \sum_{k=1}^{n} a_k H_k$. Suppose Left employs the strategy of cutting any edge in any copy of H_n uniformly at random. Then a best response for Right is to cut an edge in any copy of H_n uniformly at random.*

Proof. The resulting game if Right employs this strategy is either

$$G_1 = \sum_{k=1}^{n-1} a_k H_k + (a_n - 1)H_n$$

if Left and Right pick the same copy of H_n, or

$$G_2 = \sum_{k=1}^{n-2} a_k H_k + (a_{n-1}+1)H_{n-1} + (a_n - 2)H_n$$

if they pick different copies. Now, suppose that Right deviates, playing on a copy of H_j, to

$$G_3 = \sum_{k \neq j-1, j, n} a_k H_k + (a_{j-1}+1)H_{j-1} + (a_j - 1)H_j + (a_n-1)H_n.$$

Note that G_3 has strictly one less red edge than G_1, so $v(G_3) \geq v(G_1)$. Remains to prove that $v(G_3) \geq v(G_2)$. We proceed by induction on a_n. The case $a_n = 1$ is trivial: if Right does not play on the only copy of H_n, the resulting game will have strictly one red edge less.

Hence, suppose $a_n > 1$. The game G_3 then has at least one copy of H_n. By induction, it is optimal for Right to play on such a copy uniformly at random. In the best case (for Right), Left chooses to play on the same copy, resulting in

$$G_4 = \sum_{k \neq j-1,j,n} a_k H_k + (a_{j-1}+1)H_{j-1} + (a_j - 1)H_j + (a_n - 2)H_n.$$

Now, we couple G_2 and G_4: by induction, both players will only play on copies of H_n uniformly at random until $a_n = 0$. We let both players pick the same copy of H_n in G_2 if and only if they pick the same copy in G_4. Hence, G_2 is played to

$$G_2' = \sum_{k=1}^{n-2} a_k H_k + (a_{n-1} + 1 + \ell)H_{n-1}$$

and G_4 to

$$G_4' = \sum_{k \neq j-1,j,n-1,n} a_k H_k + (a_{j-1}+1)H_{j-1} + (a_j - 1)H_j + (a_{n-1} + \ell)H_{n-1}$$

for some $\ell \in \{0, 1, \ldots, a_n - 2\}$. Now, if $j \neq n-1$, by deviating from the optimal strategy and picking a petal from a copy of H_j, Right can move from G_2' to G_4'. If $j = n-1$, this move is optimal, but the worst case outcome for Right. Hence, in both cases, $v(G_2') \leq v(G_4')$. As this holds for all couples, we also find that $v(G_2) \leq v(G_4)$. Noting that the move from G_3 to G_4 was the best case possible for Right, we conclude that $v(G_4) \leq v(G_3)$, completing the proof. □

By the reasoning in the proof of Lemma 7.2.8, we immediately see that the value $v(a_1, \ldots, a_n)$ satisfies the following recurrence.

$$v(a_1, \ldots, a_n) = \frac{1}{a_n} v(a_1, \ldots, a_{n-1}, a_n - 1) + \frac{a_n - 1}{a_n} v(a_1, \ldots, a_{n-1} + 1, a_n - 2).$$

Indeed, if both players pick a copy of H_n uniformly at random, the probability of the players picking the same copy is $\frac{1}{a_n}$ and the probability of them picking a different copy is $\frac{a_n - 1}{a_n}$.

For large numbers of copies of flowers, we find the following.

Theorem 7.2.10. *Let $m \in \mathbb{N}$. Then* $\lim_{n\to\infty}(v(nH_m) - v((n-1)H_m)) = 2^{-m}$.

Proof. For any $n \in \mathbb{N}$, we define the stochastic process $X(n)$ by $X(0) = X(1) = 0$ with probability 1 and

$$X(n) = \begin{cases} X(n-1) & \text{with probability } \frac{1}{n}, \\ 1 + X(n-2) & \text{with probability } 1 - \frac{1}{n}. \end{cases}$$

Note that $X(n)$ models the amount of single blue edges produced when starting play from nH. Hence, $\mathbb{E}[X(n)] = v(nH)$, and, by Corollary 7.2.3, $\mathbb{E}[X(n) - \tilde{X}(n-1)] \to \frac{1}{2}$ for $n \to \infty$ for two copies $X, \tilde{X}$ of the process. For $m \in \mathbb{N}$, we define $X^m(n)$ inductively by $X^m(n) = X^{m-1}(X(n))$, i.e., $X^m(n)$ models the number of leftover blue edges when starting play from nH_m. Hence, $\mathbb{E}[X^m(n)] = v(nH_m)$ and we are interested in $\lim_{n\to\infty} \mathbb{E}[X^m(n) - \tilde{X}^m(n-1)]$.

We proceed by induction to m, the base case $m = 1$ being Corollary 7.2.3. For $k = 2, \dots, n$, we define the random variables Z_k by

$$Z_k = \begin{cases} 1 & \text{with probability } \frac{1}{k} \\ 0 & \text{with probability } 1 - \frac{1}{k}. \end{cases}$$

We couple $X(k)$ and $\tilde{X}(k)$ for all $k = 2, \dots, n$ by setting

$$X(k) = Z_k X(k-1) + (1 - Z_k)(1 + X(k-2))$$

and

$$\tilde{X}(k) = Z_k \tilde{X}(k-1) + (1 - Z_k)(1 + \tilde{X}(k-2)).$$

Note that, by this definition, $X(k) = \tilde{X}(k)$ for all $k = 0, 1, \dots, n-1$. Let $\ell = \max\{k \mid Z_k = 1\}$ and suppose first that $n - \ell$ is even. In this case,

$$X(n) = \tfrac{n-\ell}{2} + X(\ell) = \tfrac{n-\ell}{2} + X(\ell - 1)$$

and

$$\tilde{X}(n-1) = \tfrac{n-\ell}{2} + \tilde{X}(\ell - 1).$$

Hence, $X(n) = \tilde{X}(n-1)$. If $n - \ell$ is odd, we have $X(n) = \frac{n-\ell+1}{2} + X(\ell - 1)$ and

$$\tilde{X}(n-1) = \tfrac{n-\ell-1}{2} + \tilde{X}(\ell) = \tfrac{n-k-1}{2} + \tilde{X}(k-1),$$

so $X(n) = \tilde{X}(n-1) + 1$. Thus, in any case, $X(n) - \tilde{X}(n-1) \in \{0, 1\}$. By our base case, we conclude that

$$\mathbb{P}(X(n) - \tilde{X}(n-1) = 1) = \mathbb{E}[X(n) - \tilde{X}(n-1)] \to \tfrac{1}{2}$$

for $n \to \infty$. Now, we compute

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{E}[X^m(n) - \tilde{X}^m(n-1)] = \lim_{n\to\infty} \mathbb{E}[X^{m-1}(X(n)) - \tilde{X}^{m-1}(\tilde{X}(n-1))] \\
&= \lim_{n\to\infty} \Big(\mathbb{E}[X^{m-1}(X(n)) - \tilde{X}^{m-1}(\tilde{X}(n-1)) \mid X(n) - \tilde{X}(n-1) = 1] \\
&\qquad\qquad\qquad\qquad \cdot \mathbb{P}(X(n) - \tilde{X}(n-1) = 1) \\
&\quad + \mathbb{E}[X^{m-1}(X(n)) - \tilde{X}^{m-1}(\tilde{X}(n-1)) \mid X(n) - \tilde{X}(n-1) = 0] \\
&\qquad\qquad\qquad\qquad \cdot \mathbb{P}(X(n) - \tilde{X}(n-1) = 0) \Big) \\
&= (\tfrac{1}{2})^{m-1} \cdot \tfrac{1}{2} = \tfrac{1}{2^m},
\end{aligned}$$

where the last line only follows if $X(n) \to \infty$ almost surely for $n \to \infty$. Hence, remains to prove this. We compute

$$\begin{aligned}
\mathbb{P}\left(\lim_{n\to\infty} X(n) < \infty\right) &= \mathbb{P}\left(\bigcup_{k=1}^{\infty} \{X(k) = X(k+1) = \ldots\}\right) \\
&= \mathbb{P}\left(\bigcup_{k=1}^{\infty} \{Z_k = Z_{k+1} = \ldots = 1\}\right) \\
&\leq \sum_{k=1}^{\infty} \prod_{i=1}^{\infty} \mathbb{P}(Z_i = 1) \\
&= 0.
\end{aligned}$$

This concludes the proof. □

Note that for regular combinatorial Hackenbush, the value of H_m is precisely 2^{-m}. Hence, the result again supports Conjecture 2.3.34.

Finally, analyzing the sequences inspires the following result.

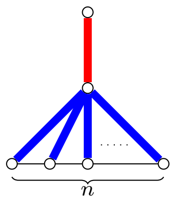
Theorem 7.2.11. *Let $m \in \mathbb{N}$. Then $v(nH_m) = 0$ for $n = 0, 1, \ldots, 2^m - 1$.*

Proof. Induction on m. The base case is $m = 1$; indeed $v(0) = v(H_1) = 0$. By Theorem 7.2.6, both players will only play on copies of H_m, uniformly at random, until none are left. At this point, we have a game consisting of $k \leq \frac{2^m}{2} = 2^{m-1}$ copies of H_{m-1}; a copy of H_{m-1} is only created when both players select a different copy of H_m, which removes two copies of H_m from the game. By induction, $v(kH_{m-1}) = 0$. □

7.3 Inverse flowers

We continue with the analysis of sums of what we will call *inverse flowers*.

Definition 7.3.1. For $n \in \mathbb{N}$, the Hackenbush position given by

$H_n^- =$ 

is called an *inverse flower* with n stems and a single leaf. We write $H^- = H_2^-$ for short.

In regular Hackenbush, the value of H^- is 1. We will show that this also holds for multiples of H^- and $-H^-$ in the synchronized version of the game.

Theorem 7.3.2. *Let* $m, n \in \mathbb{N}$. *Then* $v(mH^- - nH^-) = m - n$.

We will prove Theorem 7.3.2 by induction. The base cases are handled by the following lemma.

Lemma 7.3.3.

(1) $v(H^-) = 1$.
(2) $v(2H^-) = 2$.
(3) $v(2H^- - H^-) = 1$.

Proof.

(1) The players play to 1 on the first turn.
(2) If the players choose the same copy to play on, the result is $1 + H^-$, having value 2. If the players choose a different copy, the result is $H + 2$ with value 2. Hence, the value is 2.
(3) If both players play on the same copy of H^-, we obtain $1 + H^- - H^-$ with value 1. If the players play on different copies of H^-, this leads to $H + 2 - H^-$. From here, both players playing on H leads to $2 - H^-$ with value 1, both players playing on $-H^-$ gives $H + 2 - 1$ with value 1, Left playing on H and Right on $-H^-$ yields $2 + H$ with value 2 and Left

playing on $-H^-$ and Right on H gives $1+2-2=1$. Hence, Right will play on H and force a value of 1.

If Left plays on a copy of H^- and Right on $-H^-$, this yields $H+H^- - H$. From here, both players have three options, giving the following matrix game, where the first row and column correspond to playing on H, the second to playing on H^- and the third to playing on $-H$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & 1 & \frac{1}{2} \\ 1 & 1 & 1 \end{pmatrix}.$$

The first row dominates the other rows, so Left will play on H. Right will then play on either H or $-H$, forcing a value of 1.

If Left plays on $-H^-$ and Right on a copy of H^-, we end up in $H^- + 2 - 2$ having value 1. Finally, if both players play on $-H^-$, this results in $2H^- - 1$, also having value 1. We conclude that $2H^- - H^-$ as a whole has value 1.

□

Proof of Theorem 7.3.2. We proceed by induction on $m+n$. The base cases for $m+n<2$ are given by Lemma 7.3.3. Therefore, we assume that $m+n \geq 2$ and that the statement is true for smaller $m+n$. Throughout, if a move is unavailable (because, e.g., $m<2$ or $n<2$), we may ignore it.

If both players play on the same copy of H^-, the resulting game is $1+(m-1)H^- - nH^-$, having value $1+m-1-n=m-n$ by induction. Likewise, if both players pick the same copy of $-H^-$ to play on, the result is $mH^- - (n-1)H^- - 1$, also with value $m-n$. If Right plays on a copy of H^- and Left on a copy of $-H^-$, we obtain $(m-1)H^- + 2 - (n-1)H^- - 2$, again with value $m-n$.

If Left and Right play on different copies of H^-, we arrive at $2+H+(m-2)H^- - nH^-$. We will show by induction that $v(H+(m-2)H^- - nH^-) = m-n-2$. The base case $m=n=0$ is handled by Example 7.1.2.

First, suppose Left plays on H. If Right also plays on H, the result is $(m-2)H^- - nH^-$, having value $m-n-2$ by induction. If Right plays on a copy of H^-, we arrive at $2+(m-3)H^- - nH^-$, having value $2+m-3-n=m-n-1$. If Right plays on a copy of $-H^-$, we obtain $(m-2)H^- - (n-1)H^- - H$. Note that, by induction,

$$\begin{aligned} v((m-2)H^- - (n-1)H^- - H) &= -v(-(m-2)H^- + (n-1)H^- + H) \\ &= -(n-1-(m-2)) \\ &= m-n-1. \end{aligned}$$

Next, suppose Left plays on a copy of H^-. If Right plays on H, or on the same copy of H^-, the result is $1 + H + (m-3)H^- - nH^-$, with value $m-n-2$ by induction. If Right plays on a different copy of H^-, if possible, we arrive at $2H + (m-4)H^- + 2 - nH^-$. Note that

$$v(2H + (m-4)H^- + 2 - nH^-) \leq v(2 + H + (m-3)H^- - nH^-) = m-n-1$$

by induction. If Right plays on a copy of $-H^-$, the result is $2H + (m-3)H^- - H - (n-1)H^-$. First, note that

$$v(2H + (m-3)H^- - H - (n-1)H^-) \leq v(2H + (m-3)H^- - (n-1)H^-);$$

indeed, adding a copy of $-H$ will not decrease the value of the game. Right can stick to his best response strategy in the new game, not touching $-H$ until no more contested red edges exist in the original game. Left's best response to this strategy is to also not touch $-H$ until all contested blue edges are gone, which results precisely in the value of the old game added to $-H$, which yields value 0. Now, note that, by induction,

$$\begin{aligned} (2H + (m-3)H^- - (n-1)H^-) &\leq v(H + (m-2)H^- - (n-1)H^-) \\ &= m-2-(n-1) \\ &= m-n-1. \end{aligned}$$

Finally, suppose Left plays on a copy of $-H^-$. If Right plays on H, this yields $1 + (m-2)H^- - (n-1)H^- - 2$ with value $m-n-2$. If Right plays on a copy of H^-, this gives $H + 2 + (m-3)H^- - (n-1)H^- - 2$, also with value $m-n-2$. If Right plays on the same copy of $-H^-$, this gives $H + (m-2)H^- - (n-1)H^- - 1$, having value $m-n-2$. If Right plays on a different copy of $-H^-$, the result is $H + (m-2)H^- - (n-2)H^- - 2 - H$. We compute

$$\begin{aligned} &v(H + (m-2)H^- - (n-2)H^- - 2 - H) \\ &\quad \leq -2 + v((m-1)H^- - (n-2)H^- - H) \\ &\quad = -2 - v(-(m-1)H^- + (n-2)H^- + H) \\ &\quad = -2 - (-(m-1) + n - 2) \\ &\quad = m-n-1. \end{aligned}$$

Collecting all results, we find that playing on H is dominating for Left, after which it follows that playing on H is also dominating for Right. The result is a value of $m-n-2$ as conjectured.

Back to $mH^- - nH^-$. If the players play on different copies of $-H^-$, we arrive at $mH^- - (n-2)H^- - 2 - H$. Note that

$$v(mH^- - (n-2)H^- - H) = -v(-mH^- + (n-2)H^- + H) = m-n+2.$$

Finally, if Left plays on a copy of H^- and Right on a copy of $-H^-$, we arrive at $H + (m-1)H^- - (n-1)H^- - H$. Now, note that, by the same reasoning as before,

$$v(H+(m-1)H^- - (n-1)H^- - H) \leq v(H+(m-1)H^- - (n-1)H^-) = m-n.$$

Furthermore,

$$\begin{aligned} v(H+(m-1)H^- - (n-1)H^- - H) &\geq v((m-1)H^- - (n-1)H^- - H) \\ &= -v(-(m-1)H^- + (n-1)H^- + H) \\ &= m-n. \end{aligned}$$

Hence, we conclude that $v(H + (m-1)H^- - (n-1)H^- - H) = m-n$. Collecting all results, we find that, no matter the strategies of both players, our game $mH^- - nH^-$ will result in a value of $m-n$. □

For copies of the game H_n^- for different n, we obtain the following result analogous to Theorem 7.2.6.

Theorem 7.3.4. *Let $n \in \mathbb{N}$, $a_1, \ldots, a_n \in \mathbb{N}$ and consider $G = \sum_{k=1}^{n} a_k H_k^-$. Then*

$$v(G) = v(nH) + \sum_{k=1}^{n} a_k(k-1).$$

The following partial result is easy to prove, and forms the basis for the proof of Theorem 7.3.4.

Lemma 7.3.5. *Let $n \in \mathbb{N}$, $a_2, \ldots, a_n \in \mathbb{N}$ and consider $G = \sum_{k=2}^{n} a_k H_k^-$. Then*

$$v(G) = \sum_{k=2}^{n} a_k(k-1).$$

Proof. As there are $\sum_{k=2}^{n} a_k$ red edges in total, the game cannot last more than $\sum_{k=2}^{n} a_k$ turns. Moreover, as removing a red edge never removes a blue edge, and removing a blue edge never removes another blue edge, every turn, exactly one blue edge will be removed. Hence, to minimize the outcome, Right needs to make the game last $\sum_{k=2}^{n} a_k$ turns.

To do so, on the first turn, let Right remove a red edge at random. On the next turns, if there is a copy of H, Right plays on this copy; otherwise, he plays randomly. Note that at most one copy of H can be created by Left every turn, so there will never be more than one copy of H. Moreover, by using this strategy, Right prevents Left from ever removing a red edge. Hence, the game lasts $\sum_{k=2}^{n} a_k$ turns, so $\sum_{k=2}^{n} a_k$ blue edges are removed, leaving a value of $\sum_{k=2}^{n} a_k k - \sum_{k=2}^{n} a_k = \sum_{k=2}^{n} a_k(k-1)$. □

Proof of Theorem 7.3.4. We claim that the optimal strategy for both players is to play on a copy of H uniformly at random until none are left. Proof will go by induction on a_1. For $a_1 = 1$, if both players adhere to this strategy, the resulting game is $\sum_{k=2}^{n} a_k H_k^-$ with value $\sum_{k=2}^{n} a_k(k-1)$ by Lemma 7.3.5. If Right deviates and plays on any copy of H_j^-, say, the resulting game has value $j + \sum_{k=2}^{n} a_k(k-1)$, which is clearly not profitable. If Left deviates by playing on H_j^-, the resulting game has value $1 + \sum_{k=2}^{n} a_k(k-1) - (j-2) + j - 3$, which is also not profitable. This concludes the base case.

Now, suppose $a_1 > 1$. If both players employ the described strategy, the resulting game is either

$$G_1 = (a_1 - 1)H + \sum_{k=2}^{n} a_k H_k^-$$

if both players pick the same copy of H to play on, or

$$G_2 = 1 + (a_1 - 2)H + \sum_{k=2}^{n} a_k H_k^-$$

if they pick different copies. First, suppose Right deviates, playing on a copy of H_j, $j > 1$, to

$$G_3 = j + \sum_{k \neq 1, j} a_k H_k^- + (a_1 - 1)H + (a_j - 1)H_j^- .$$

Note that G_3 has strictly one less red edge than G_1, so $v(G_3) \geq v(G_1)$. Remains to prove that $v(G_3) \geq v(G_2)$. The game G_3 has at least one copy of H. By induction, it is optimal for Right to play on such a copy uniformly at random. In the best case (for Right), left chooses to play on the same copy, resulting in

$$G_4 = j + \sum_{k \neq 1,j} a_k H_k^- + (a_1 - 2)H + (a_j - 1)H_j^-.$$

Now, we couple G_2 and G_4: by induction, both players will only play on copies of H uniformly at random until $a_1 = 0$. We let both players pick the same copy of H in G_2 if and only if they pick the same copy in G_4. Hence, G_2 is played to

$$G_2' = 1 + \ell + \sum_{k=2}^{n} a_k H_k^-$$

and G_4 to

$$G_4' = j + \ell + \sum_{k \neq 1,j} a_k H_k^- + (a_j - 1)H_j^-$$

for some $\ell \in \{0, 1, \ldots, a_1 - 2\}$. By Lemma 7.3.5, we have

$$v(G_2') = 1 + \ell + \sum_{k=2}^{n} a_k(k-1)$$

and

$$v(G_4') = j + \ell + \sum_{k=2}^{n} a_k(k-1) - (j-1) = v(G_2').$$

Hence, also $v(G_2) = v(G_4)$. Noting that the move from G_3 to G_4 was the best possible for Right, we conclude that $v(G_4) \leq v(G_3)$. Right thus cannot gain by deviating.

Next, suppose Left deviates, playing on a copy of H_j, $j > 1$, to

$$G_5 = 1 + \sum_{k \neq 1, j-1, j} a_k H_k^- + (a_1 - 1)H + (a_{j-1} + 1)H_{j-1}^- + (a_j - 1)H_j^-.$$

Note that G_2 has strictly one less red edge than G_1, so $v(G_1) \leq v(G_2)$. Remains to prove that $v(G_5) \leq v(G_1)$. First, suppose $j = 2$. In this case, Left can move from G_1 to G_5 by a non-optimal move (by induction), so indeed $v(G_5) \leq v(G_1)$ holds. Thus, assume that $j > 2$. We couple G_1 and G_5 like before: by induction, in both games, both players play on a copy of H uniformly at random until none are left — let the players play on the same copies in both games. The resulting games are

$$G_1' = \ell + \sum_{k=2}^{n} a_k H_k^-$$

and

$$G_5' = 1 + \ell + \sum_{k \neq 1, j-1, j} a_k H_k^- + (a_{j-1} + 1) H_{j-1}^- + (a_j - 1) H_j^-$$

for some $\ell \in \{0, 1, \ldots, a_1 - 1\}$, respectively. By Lemma 7.3.5, it follows that

$$v(G_1') = \ell + \sum_{k=2}^{n} a_k (k-1)$$

and

$$v(G_5') = 1 + \ell + \sum_{k=2}^{n} a_k (k-1) + j - 2 - (j-1) = v(G_1').$$

We conclude that $v(G_1) = v(G_5)$, which completes the proof. $\square$

7.4 Halves and their negatives

We continue by analyzing how sums of H and $-H$ behave, that is, finding out what the value of $mH - nH$ is for any m and n natural numbers. In this section, we denote $v_{m,n} = v(mH - nH)$.

First, note that $v_{n,m} = v(nH - mH) = v(-(mH - nH)) = -v(mH - nH) = -v_{m,n}$, so also $v_{n,n} = -v_{n,n}$, from which we conclude that $v_{n,n} = 0$ for all n, in accordance with Proposition 2.3.31. Hence, any game consisting of an equal number of copies of H and copies of $-H$ will result in a draw.

For $m \neq n$, a more detailed analysis is required. Both players have the option of playing on one of the copies of H, or on one of the copies of $-H$. Moving from $mH - nH$, if both players play on the same copy of H, the game will continue as $(m-1)H - nH$, yielding value $v_{m-1,n}$. Similarly, if both players play on the same copy of $-H$, the outcome will be $v_{m,n-1}$. If both players play on different copies of H, the result is $1 + v_{m-2,n}$. Playing on different copies of $-H$ yields $v_{m,n-2} - 1$. If one player plays on a copy of H and the other on a copy of $-H$, the result always has value $v_{m-1,n-1}$. We thus arrive at the following conclusion.

Theorem 7.4.1. *Let $v_{m,n} = v(mH - nH)$. If $m = n$, then $v_{m,n} = 0$. Otherwise, it is the Nash value of the following zero-sum matrix game:*

$$\begin{pmatrix}
v_{m-1,n} & 1+v_{m-2,n} & \cdots & 1+v_{m-2,n} & v_{m-1,n-1} & \cdots & \cdots & v_{m-1,n-1} \\
1+v_{m-2,n} & \ddots & \ddots & \vdots & \vdots & \ddots & & \vdots \\
\vdots & \ddots & \ddots & 1+v_{m-2,n} & \vdots & & \ddots & \vdots \\
1+v_{m-2,n} & \cdots & 1+v_{m-2,n} & v_{m-1,n} & v_{m-1,n-1} & \cdots & \cdots & v_{m-1,n-1} \\
v_{m-1,n-1} & \cdots & \cdots & v_{m-1,n-1} & v_{m,n-1} & v_{m,n-2}-1 & \cdots & v_{m,n-2}-1 \\
\vdots & \ddots & & \vdots & v_{m,n-2}-1 & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \vdots & \vdots & \ddots & \ddots & v_{m,n-2}-1 \\
v_{m-1,n-1} & \cdots & \cdots & v_{m-1,n-1} & v_{m,n-2}-1 & \cdots & v_{m,n-2}-1 & v_{m,n-1}
\end{pmatrix}$$

Theorem 7.4.1 allows us to recursively compute the values $v_{m,n}$ using linear programming. The results for m and n up to 8 are shown in Table 7.1.

$m \setminus n$	0	1	2	3	4	5	6	7	8
0	0.000								
1	0.000	0.000							
2	0.500	0.250	0.000						
3	0.833	0.643	0.250	0.000					
4	1.333	1.006	0.666	0.250	0.000				
5	1.733	1.459	1.059	0.666	0.250	0.000			
6	2.233	1.867	1.508	1.076	0.666	0.250	0.000		
7	2.662	2.339	1.928	1.528	1.078	0.666	0.250	0.000	
8	3.162	2.771	2.394	1.959	1.534	1.078	0.666	0.250	0.000

Table 7.1: The Nash value $v(mH - nH)$ rounded to three decimal places, for $n, m \leq 8$. Note that $v(mH - nH) = -v(nH - mH)$.

Looking at these values, we arrive at the following conjectures.

Conjecture 7.4.2. *Let $v_{m,n} = v(mH - nH)$. Then*

(1) *For all $n \geq 2$, we have $v_{n,n-1} = \frac{1}{4}$.*
(2) *For all n, we have $\lim_{m\to\infty}(v_{m,n} - v_{m-1,n}) = \frac{1}{2}$.*
(3) *For all m, we have $\lim_{n\to\infty}(v_{m,n} - v_{m,n-1}) = -\frac{1}{2}$.*

(4) *For all m, the limit $\lim_{n\to\infty} v_{m+n,n}$ exists.*

The values $v_{m,n}$ can also be computed using a recurrence relation derived from the linear program formulation for solving the matrix game in Theorem 7.4.1. Let y_i, $i = 1,\ldots,m$ be the probability of Left playing on the i-th copy of H, corresponding to the first m columns of the matrix, and let z_i, $i = 1,\ldots,n$ be the probability of Left playing on the i-th copy of $-H$, corresponding with the last n columns of the matrix. Note that, given any constraint in a row of the matrix, by permuting the y_i's and z_i's in any way, we arrive at another row in the matrix. Hence, by Theorem 2.2.20, we may replace all y_i's by a new variable x_1 and all z_i's by one new variable x_2, where x_1 is the probability of Left playing on some copy of H, and x_2 is the probability of Left playing on some copy of $-H$. The LP with these new variables then becomes

$$\max\left\{ x_0 \;\middle|\; \begin{array}{rcl} x_0 & \leq & v_{m-1,n}x_1 + (m-1)(1+v_{m-2,n})x_1 + nv_{m-1,n-1}x_2 \\ x_0 & \leq & mv_{m-1,n-1}x_1 + v_{m,n-1}x_2 + (n-1)(v_{m,n-2}-1)x_2 \\ & & \hfill mx_1 + nx_2 = 1 \\ & & \hfill x_1, x_2 \geq 0 \end{array} \right\}$$

Rewriting gives

$$\max\left\{ x_0 \;\middle|\; \begin{array}{rcl} x_0 & \leq & (v_{m-1,n} + (m-1)(1+v_{m-2,n}))x_1 + nv_{m-1,n-1}x_2 \\ x_0 & \leq & mv_{m-1,n-1}x_1 + (v_{m,n-1} + (n-1)(v_{m,n-2}-1))x_2 \\ & & \hfill mx_1 + nx_2 = 1 \\ & & \hfill x_1, x_2 \geq 0 \end{array} \right\}$$

Note that $x_2 = \frac{1-mx_1}{n}$. Renaming $x = x_1$, we may thus rewrite the program as

$$\max\left\{ x_0 \;\middle|\; \begin{array}{rcl} x_0 & \leq & (v_{m-1,n} + (m-1)(1+v_{m-2,n}))x + v_{m-1,n-1}(1-mx) \\ x_0 & \leq & mv_{m-1,n-1}x + (v_{m,n-1} + (n-1)(v_{m,n-2}-1))\frac{1-mx}{n} \\ & & \hfill 0 \leq x \leq \frac{1}{m} \end{array} \right\}$$

For this one-dimensional problem, the optimal solution is found either at the border of the interval ($x = 0$ or $x = \frac{1}{m}$), or at the intersection point of the two constraints. We compute this intersection point by simply solving the linear equation, yielding the following result.

Theorem 7.4.3. *Consider $G = mH - nH$, $m, n \geq 2$, and let $v_{m,n} = v(G)$. Define*

$$\delta_m = \tfrac{1}{m}v_{m-1,n} + \tfrac{m-1}{m}(1+v_{m-2,n}) - v_{m-1,n-1}$$

and

$$\delta_n = \tfrac{1}{n}v_{m,n-1} + \tfrac{n-1}{n}(v_{m,n-2}-1) - v_{m-1,n-1}.$$

Then, in the Nash equilibrium, the probability $p_{m,n}$ for either player to play on any one copy of H is

$$p_{m,n} = \frac{1}{m} \cdot \frac{\delta_n}{\delta_m + \delta_n}.$$

if this value lies in $[0, \frac{1}{m}]$. Otherwise, Left plays on a copy of H with probability $\frac{1}{m}$, and Right plays on a copy of $-H$ with probability $\frac{1}{n}$.

Filling in the obtained value in the constraints, we thus arrive at the following conclusion.

Theorem 7.4.4. *Let $v_{m,n} = v(mH - nH)$, $m, n \geq 2$, and let δ_m, δ_n and $p_{m,n}$ be as in Theorem 7.4.3. Then*

$$v_{m,n} = v_{m-1,n-1} + \frac{\delta_m \delta_n}{\delta_m + \delta_n}$$

if $p_{m,n} \in [0, \frac{1}{m}]$ and $v_{m,n} = v_{m-1,n-1}$ otherwise.

Some remarks are in order. First, note that the solution displayed in the above theorems is in fact simply the solution to the non-cooperative game defined by the following 2×2-matrix:

$$\begin{pmatrix} \frac{1}{m} v_{m-1,n} + \frac{m-1}{m}(v_{m-2,n} + 1) & v_{m-1,n-1} \\ v_{m-1,n-1} & \frac{1}{n} v_{m,n-1} + \frac{n-1}{n}(v_{m,n-2} - 1) \end{pmatrix}.$$

Furthermore, note that we may interpret δ_n as the increase in payoff established for Left when playing on $-H$ uniformly at random instead of on H, assuming that Right plays on $-H$ uniformly at random.

We expand upon this a little more. Define $a = \frac{1}{m} v_{m-1,n} + \frac{m-1}{m}(v_{m-2,n} + 1)$, $b = v_{m-1,n-1}$ and $c = \frac{1}{n} v_{m,n-1} + \frac{n-1}{n}(v_{m,n-2} - 1)$. The matrix game then becomes

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

We distinguish four cases:

(i) $a > b > c$. Now the top-right entry b is a saddle point of the game, so the optimal strategy for Left is to play uniformly at random on a copy of H and for Right to play uniformly at random on a copy of $-H$, the value being $b = v_{m-1,n-1}$. We have $\delta_m > 0 > \delta_n$.

(ii) $a > b < c$. The game has no saddle point, and the value is given by Theorem 7.4.4. We have $\delta_L, \delta_R > 0$, which guarantees $p_{m,n}$ is well-defined and $v_{m,n} \geq v_{m-1,n-1}$.

(iii) $a < b > c$. Again, there is no saddle point, and the Nash value and strategies are the same as in (ii). We now have $\delta_m, \delta_n < 0$, from which it follows that $p_{m,n}$ is well-defined and $v_{m,n} \leq v_{m-1,n-1}$.

(iv) $a < b < c$. Now, the bottom-left entry b is a saddle point. The optimal strategy for Left is to play only on $-H$, and for Right to play on H, giving a value of $b = v_{m-1,n-1}$. We have $\delta_m < 0 < \delta_n$.

We conjecture that for $m \geq n$, we are always in case (i) or (ii), for $m \leq n$ we are always in case (i) or (iii), and case (iv) never occurs. To prove Conjecture 7.4.2, it would then be sufficient to show that $v_{m+n,n}$ is bounded from above for $m > 0$ and bounded from below for $m < 0$, and/or that, for n large enough, we always end up in case (i). Hence, the relations between a, b and c, and therewith the sign of δ_m and δ_n, seem to be crucial.

Though a proof of Conjecture 7.4.2 remains elusive, we are able to prove the following, somewhat similar, result.

Theorem 7.4.5. *For all m and n, we have*

$$\lim_{k \to \infty} \frac{v_{km,kn}}{k} = \frac{m-n}{2}.$$

The following lemma turns out to be helpful.

Lemma 7.4.6. *For all $m, n \geq 0$, we have $v_{m,n+1} + 1 \geq v_{m+1,n}$.*

Proof. Consider the games $G_1 = 1 - H + mH - nH$ and $G_2 = H + mH - nH$, of which $v_{m,n+1} + 1 = v(G_1)$ and $v_{m+1,n} = v(G_2)$ are the Nash values. We couple G_1 and G_2. Let Right play his Nash equilibrium strategy on G_1. As for his strategy on G_2, if he plays on $mH - nH$ in G_1, the move is also executed in G_2. If he plays on $-H$ in G_1, he plays on the H in G_2. Likewise, let Left play her Nash strategy on G_2, playing on $-H$ in G_1 if she plays on H in G_2. Denote the values of the games under these strategy pairs by $v^L(G_1)$ and $v^R(G_2)$, respectively.

By the coupling, the $mH - nH$ part of both games will play out the same. We analyze the possible outcome of the other parts. If Left and Right simultaneously decide not to play on $mH - nH$, the result is 1 in G_1 and 0 in G_2. If Left does not play on $mH - nH$, but Right does, the result is $1 - 1 = 0$ in G_1 and 0 in G_2. Vice versa, if Right does not play on $mH - nH$, the result is 1 in both G_1 and G_2. Hence, we find that $v^L(G_1) \geq v^R(G_2)$.

By the definition of a Nash equilibrium, we also find that $v(G_1) \geq v^L(G_1)$ and $v^R(G_2) \geq v(G_2)$, so that indeed $v(G_1) \geq v(G_2)$. □

Proof of Theorem 7.4.5. Using Lemma 7.4.6 and Theorem 7.2.4, we compute, for fixed k,

$$\begin{aligned}
v_{km,kn} &\le km + v_{0,km+kn}\\
&= km - \frac{1}{4}\left(2(km+kn) - 2 - \psi\left(\frac{km+kn+1+\mathbb{1}_{\{kn+km \text{ odd}\}}}{2}\right) + \psi\left(\frac{3}{2}\right)\right)\\
&\le \frac{km}{2} - \frac{kn}{2} + \psi\left(\frac{km+kn+2}{2}\right) + O(1)\\
&\le \frac{km}{2} - \frac{kn}{2} - \frac{1}{km+kn+2} + \log(km+kn) + O(1)\\
&= \frac{km}{2} - \frac{kn}{2} + \log(km+kn) + O(1).
\end{aligned}$$

By a symmetrical argument, we find

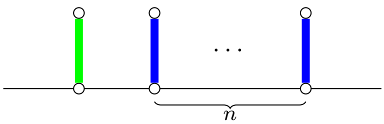
$$\frac{km}{2} - \frac{kn}{2} - \log(km+kn) + O(1) \le v_{km,kn} \le \frac{km}{2} - \frac{kn}{2} + \log(km+kn) + O(1).$$

Dividing by k and taking $k \to \infty$ yields the result. □

7.5 Red-Blue-Green Hackenbush

We conclude the chapter with a brief overview of synchronized Red-Blue-Green Hackenbush, mainly with the aim of providing an interesting topic for future research. Incorporating green edges, which may be cut by either player, results in the ruleset no longer being separable. Hence, the games at hand can no longer be modelled as standard zero-sum matrix games. Instead, we obtain games in which repetition is possible: if both players decide to cut the same green edge simultaneously, this results in no change, forcing the players to try again. If the equilibrium strategy — if this exists — prescribes both players to take this move with probability 1, we declare the game a draw, having value 0.

Example 7.5.1. For $n \in \mathbb{N}$, consider the game

$B_n =$ 

For $n = 0$, both players play on the only available green edge with probability 1, ending in a draw by definition. For $n = 1$, whichever strategy the players employ, the result is again a draw with value 0. For $n = 2$, the dominating strategy for Left is to play on any blue edge, with Right removing the only green edge, giving a value of $n - 1$. $\triangleleft$

This example shows that the statement of Conjecture 2.3.33 fails to hold for this non-separable game: for Left, it is more beneficial to play on one of her own blue edges, saving the positive value of the remaining ones, than to play on the contested green edge, which would result in a value of 0.

Furthermore, as already touched upon in Section 2.3.4, the existence of Nash equilibria is a priori no longer guaranteed, and, moreover, if they exist, they are much harder to find. Indeed, recursively defining the value of a synchronized move to be equal to the value of the game itself results in a quadratic problem instead of a linear one, which is computationally much more complicated.

Example 7.5.2. Let $a, b, c \in \mathbb{R}$, and consider the game described by the matrix

$$\begin{pmatrix} * & a \\ b & c \end{pmatrix},$$

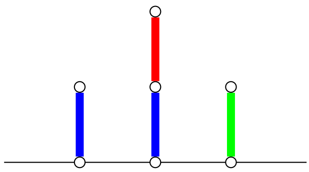
where the $*$ denotes a required repetition of the game. The program to find the value of the game now reads

$$\max \left\{ x_0 \;\middle|\; \begin{array}{rcrcl} x_0 & \leq & x_0 x_1 & + & a x_2 \\ x_0 & \leq & b x_1 & + & c x_2 \\ \multicolumn{5}{c}{x_1 + x_2 = 1, x_1, x_2 \geq 0} \end{array} \right\}.$$

This is indeed no longer linear. $\triangleleft$

Instead, one could resolve to using partial derivatives and Lagrange multipliers to find the value of the game. In any case, the result is computationally harder than regular zero-sum games.

Example 7.5.3. Consider the game

$G =$ 

Let p_1 be the probability of Left playing on the leftmost blue edge, p_2 on the rightmost blue edge, and p_3 the probability that Left plays on the green edge.

Similarly, let q_1 be the probability that Right cuts the red edge, and q_2 the probability that he cuts the green edge.

The value of the game is given by

$$\begin{aligned} v &= p_1q_1 \cdot 0 + p_1q_2 \cdot 0 + p_2q_1 \cdot 0 + p_2q_2 \cdot 1 + p_3q_1 \cdot 2 + p_3q_2 \cdot v \\ &= p_2q_2 + 2p_3q_1 + p_3q_2v, \end{aligned}$$

i.e., $v = \frac{p_2q_2+2p_3q_1}{1-p_3q_2}$. Noting that $q_1 + q_2 = 0$ must hold, we may rewrite $v = \frac{p_2-p_2q_1+2p_3q_1}{1-p_3+p_3q_1}$. Furthermore, note that Left can always force a value of at least 0 by picking $p_2 = 1$, so that we may set $p_1 = 0$. Hence, $p_2 + p_3 = 1$ and we obtain $v = \frac{p_2-3p_2q_1+2q_1}{p_2+q_1-p_2q_1}$.

Solving $\frac{\partial v}{\partial p_2} = 0$ yields $q_1 = 0$ as only feasible solution. Solving $\frac{\partial v}{\partial q_1} = 0$ yields $p_2 = 0$ or $p_2 = \frac{1}{2}$. Note that the combination $p_2 = q_1 = 0$ does not yield a Nash equilibrium: Left can improve by playing on the rightmost blue edge instead, obtaining value 1. Hence, the only Nash equilibrium is for Right to always play on the green edge, and for Left to either play on the rightmost blue edge or the green edge with probability $\frac{1}{2}$, yielding a value of 1. $\triangleleft$

Chapter 8

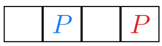
Synchronized Push

In this chapter, based on joint work with Ronald Takken [19], we discuss some results on the synchronized versions of Push and Shove. Both games being separable, the results largely mirror those in Chapter 7 and support Conjecture 2.3.34. In this chapter, we consider both the combinatorial and synchronized version of Push in detail, showing that it behaves similarly but not identically to Hackenbush. In Section 8.4, we briefly consider the game of Shove, again showing some similar behavior.

8.1 Basics

The ruleset of Push, described in Example 2.1.20, provides us with separable games. If the players pick two pieces of which the moves do not influence each other, it is clear that the moves can be executed simultaneously in any order. If the move of one of the pieces would cause the second piece to be pushed, we perform the move on the second piece first.

Example 8.1.1. Consider the synchronized game

$G =$ 

Here, moving either piece does not affect the position of the other piece, so $G^{LR} = G^{RL} = G^S$ is the only synchronized move of the game.

Next, consider

$$G = \boxed{}\boxed{\color{blue}P}\boxed{\color{red}P}$$

We now find that the blue piece is pushed by moving the red piece, so the only synchronized move amounts to first moving the blue piece out of the way, followed by moving the red piece: $G^S = G^R = G^{LR}$. $\triangleleft$

Hence, by Corollary 2.3.11, all Push positions are numbers in combinatorial sense. The numerical values of some simple combinatorial Push positions are shown in [2, Problem 5.15], proven in [19].

Theorem 8.1.2. *Writing* $\boxed{}^n$ *for n blank squares, we have*

(i) $\boxed{}^n\boxed{\color{blue}P} = n+1$;
(ii) $\boxed{}^n\boxed{\color{red}P}\boxed{\color{blue}P} = 2 - \frac{1}{2^{n+1}}$;
(iii) $\boxed{}^n\boxed{\color{red}P}\boxed{}^m\boxed{\color{blue}P} = m+1$ *for* $m > 0$.

For Push and Shove, we are able to prove Conjecture 2.3.33.

Theorem 8.1.3. *Let G and H be games of synchronized Push or Shove, and suppose T is terminal. Then $v(G+T) = v(G) + v(T)$.*

Proof. We prove the statement for Push; for Shove, the proof is analogous. If G is also terminal, the result is trivial. Hence, suppose G is non-terminal. Furthermore, without loss of generality, suppose $T \in \mathcal{L}$, in which case we may represent T by $T = \boxed{}^n\boxed{\color{blue}P}$.

Let G^R be any move of Left which moves the leftmost piece on any one strip of G, let G^R be any arbitrary move on G, and consider $G_1 = G^{L+R} + T$ and $G_2 = G^R + T^L = G^R + T - 1$. First, if $G^{L+R} = G^{LR} = G^R$, i.e., if Left's chosen piece is pushed by Right's move, then

$$v(G_1) = v(G^{L+R}) + v(T) = v(G^R) + v(T) > v(G^R) + v(T) - 1 = v(G_2).$$

Next, suppose that Left's chosen piece is not pushed by the move of Right. Noting that the chosen piece is the leftmost blue one by assumption, the result is

$$G_1 = \cdots\boxed{\color{red}P}\cdots\boxed{\color{red}P}\boxed{\color{blue}P}\boxed{}\cdots + \boxed{}^n\boxed{\color{blue}P}$$
$$G_2 = \cdots\boxed{}\boxed{\color{red}P}\cdots\boxed{\color{red}P}\boxed{\color{blue}P}\cdots + \boxed{}^{n-1}\boxed{\color{blue}P}$$

Note that the amount of red pieces to the left of the moved blue piece may be zero, and that the position of the empty square in G_2 may vary depending on Right's move. Now, couple G_1 and G_2. Let Left play her Nash strategy on G_2, and move the same piece on G_1. Conversely, let Right play his Nash strategy on G_1, moving the corresponding piece in G_2. Denote the values under these strategies by $v^L(G_1)$ and $v^R(G_2)$, respectively. Let play continue until the leftmost blue piece in G_1 is moved off the board, ending in

$$\begin{aligned} G'_1 &= \boxed{}\cdots + \boxed{}^{\,n}\boxed{\color{blue}P} \\ G'_2 &= \boxed{\color{blue}P}\cdots + \boxed{}^{\,n-1}\boxed{\color{blue}P} \end{aligned}$$

if this piece has not been pushed in G_2, or in

$$\begin{aligned} G'_1 &= \cdots + \boxed{}^{\,n}\boxed{\color{blue}P} \\ G'_2 &= \cdots + \boxed{}^{\,n-1}\boxed{\color{blue}P} \end{aligned}$$

if it has been pushed. In the second case, it is clear that $v(G'_1) > v(G'_2)$. In the first case, continue the coupling, connecting a move on the leftmost piece in G'_2 to a move on H in G'_1. We then arrive in either

$$\begin{aligned} G''_1 &= \cdots + \boxed{}^{\,n-1}\boxed{\color{blue}P} \\ G''_2 &= \cdots + \boxed{}^{\,n-1}\boxed{\color{blue}P} \end{aligned}$$

or

$$\begin{aligned} G''_1 &= \cdots + \boxed{}^{\,n}\boxed{\color{blue}P} \\ G''_2 &= \cdots + \boxed{}^{\,n-1}\boxed{\color{blue}P} \end{aligned}$$

depending on whether this piece is pushed at some time. In both cases, it is clear that $v(G''_1) \geq v(G''_2)$. Hence, we find that $v(G_1) \geq v^L(G_1) \geq v^R(G_2) \geq v(G_2)$.

We thus conclude that moving on G dominates moving on T for Left. Hence, by domination and induction, we conclude that $v(G+T) = v(G) + v(T)$. □

In the sequel, we may write n and $-n$ for integer Push and Shove games, and consider these separately from the other summands in sums of games.

8.2 Halves

We continue by considering Conjecture 2.3.34 in the light of Push games. To this end, we start by analyzing copies of the games

$$H = 2 + \boxed{\color{blue}P}\boxed{\color{red}P} \quad \text{and} \quad H' = -1 + \boxed{\color{red}P}\boxed{\color{blue}P}$$

which are both equal to $\frac{1}{2}$ in combinatorial sense by Theorem 8.1.2. Let $v_n = v(nH)$ and $v'_n = v(nH')$. We then find the following recurrence relations, akin to Theorem 7.2.1.

Theorem 8.2.1. *We have*

$$v_n = \begin{cases} 1 + \frac{1}{n}v_{n-1} + \frac{n-1}{n}v_{n-2} & n \geq 3, \\ v_1 = 1, v_2 = \frac{3}{2}, \end{cases}$$

and

$$v'_n = \begin{cases} \frac{1}{n}v'_{n-1} + \frac{n-1}{n}(v'_{n-2} + 1) & n \geq 3, \\ v'_1 = 0, v'_2 = \frac{1}{2}. \end{cases}$$

Proof. Computing v_1 is trivial; for v_2, note that playing on either copy of H uniformly at random is a Nash equilibrium by Theorem 2.2.20, resulting in $v_2 = \frac{1}{2}(2-1+2-1) + \frac{1}{2}(2-2+2-1) = \frac{3}{2}$. For the recurrence step, again by Theorem 2.2.20, both players play on the same copy of H with probability $\frac{1}{n}$, to $(n-1)H + (2-1)$ with value $v_{n-1}+1$; or on different copies with probability $\frac{n-1}{n}$, to $(n-2)H + (2-2) + (2-1)$ with value $v_{n-2}+1$.

Again, computing v'_1 and v'_2 is straightforward. For the recurrence step, once more by Theorem 2.2.20, the players play on the same copy of H' with probability $\frac{1}{n}$, to $(n-1)H' + (-1+1)$ with value v'_{n-1}; or on different copies with probability $\frac{n-1}{n}$, to $(n-2)H' + (-1+1) + (-1+2)$ with value $v'_{n-2}+1$. □

Just like for Hackenbush, the recurrence relations do not have a closed-form solution, but the differences $d_n = v_n - v_{n-1}$ and $d'_n = v'_n - v'_{n-1}$ are well-behaved.

Theorem 8.2.2. *We have*

$$d_n = \begin{cases} 1 - \frac{n-1}{n}d_{n-1} & n \geq 2, \\ d_1 = 1, \end{cases}$$

and

$$d'_n = \begin{cases} \frac{n-1}{n}(1 - d'_{n-1}) & n \geq 2, \\ d'_1 = 0. \end{cases}$$

The solutions are

$$d_n = \frac{2n + (-1)^{n-1} + 1}{4n} \quad \textit{and} \quad d'_n = \frac{2n + (-1)^n - 1}{4n}.$$

Proof. We expand:

$$\begin{aligned} d_n &= v_n - v_{n-1} \\ &= 1 + \tfrac{1}{n} v_{n-1} + \tfrac{n-1}{n} v_{n-2} - v_{n-1} \\ &= 1 - \tfrac{n-1}{n}(v_{n-1} - v_{n-2}) \\ &= 1 - \tfrac{n-1}{n} d_{n-1}. \end{aligned}$$

and

$$\begin{aligned} d'_n &= v'_n - v'_{n-1} \\ &= \tfrac{1}{n} v'_{n-1} + \tfrac{n-1}{n}(v'_{n-2} + 1) - v'_{n-1} \\ &= \tfrac{n-1}{n} + \tfrac{n-1}{n} v'_{n-2} - \tfrac{n-1}{n} v'_{n-1} \\ &= \tfrac{n-1}{n}(1 - d'_{n-1}). \end{aligned}$$

The solutions may be verified by substituting them into the proven recurrences. □

Corollary 8.2.3. *We have*

$$\lim_{n\to\infty} \frac{v(nH)}{n} = \lim_{n\to\infty} \frac{v(nH')}{n} = \frac{1}{2}.$$

Hence, we see that for these Push positions, Conjecture 2.3.34 holds. Note that, just as for d_n in Section 7.2, the parity of d_n is crucial to its value. Here, $d_{2k} = d'_{2k+1} = \frac{1}{2}$ for $k \in \mathbb{N}$, d_{2k+1} converges to $\frac{1}{2}$ from above and d'_{2k} from below, as shown in Figure 8.1. We will encounter a similar pattern again later on.

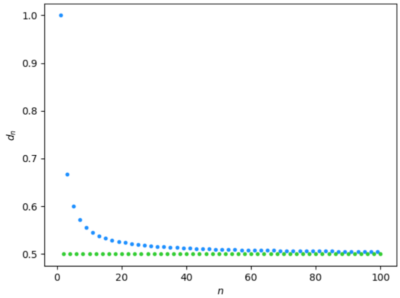
Next, we consider games of the form $mH - nH$. Some computed values can be found in Table 8.1.

We again follow a path similar to that in Chapter 7, mirroring Conjecture 7.4.2, Lemma 7.4.6 and Theorem 7.4.5.

Conjecture 8.2.4. *Let* $v_{m,n} = v(mH - nH)$. *Then*

(1) *For all* $n \geq 2$, *we have* $v_{n,n-1} = \frac{3}{4}$.
(2) *For all* n, *we have* $\lim_{m\to\infty}(v_{m,n} - v_{m-1,n}) = \frac{1}{2}$.
(3) *For all* m, *we have* $\lim_{n\to\infty}(v_{m,n} - v_{m,n-1}) = -\frac{1}{2}$.
(4) *For all* m, *the limit* $\lim_{n\to\infty} v_{m+n,n}$ *exists.*

Lemma 8.2.5. *For all* $m, n \geq 0$, *we have* $v_{m,n+1} + 1 \geq v_{m+1,n}$.



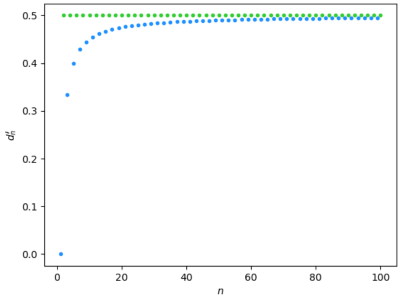


Figure 8.1: The convergence of d_n (left) and d'_n (right). The values for d_n for n even and d'_n for n odd are shown in green; the other values in blue.

$m \setminus n$	0	1	2	3	4	5	6	7
0	0.0000							
1	1.0000	0.0000						
2	1.5000	0.7500	0.0000					
3	2.1667	1.3571	0.7500	0.0000				
4	2.6667	1.9935	1.3338	0.7500	0.0000			
5	3.2667	2.5415	1.9411	1.3338	0.7500	0.0000		
6	3.7667	3.1333	2.4923	1.9236	1.3338	0.7500	0.0000	
7	4.3381	3.6608	3.0719	2.4721	1.9220	1.3338	0.7500	0.0000

Table 8.1: The Nash value $v(mH - nH)$ rounded to four decimal places. Note that $v(mH - nH) = -v(nH - mH)$.

Proof. We couple $G_1 = 1 - H + mH - nH$ and $G_2 = H + mH - nH$ as in Lemma 7.4.6, again denoting the values under the coupled strategies by $v^L(G_1)$ and $v^R(G_2)$, respectively. If Left and Right both not play on $mH - nH$, the result is 0 in both G_1 and G_2. If only Left does not play on $mH - nH$, the result is the same. If only Right does not play on $mH - nH$, then the result is 1 in both G_1 and G_2. In any case, we conclude that $v^L(G_1) \geq v^R(G_2)$, so that also $v(G_1) \geq v^L(G_1) \geq v^R(G_2) \geq v(G_2)$. □

Theorem 8.2.6. *For all $m, n \in \mathbb{N}$ we have*

$$\lim_{k\to\infty} \frac{v_{km,kn}}{k} = \frac{m-n}{2}.$$

Proof. Using Theorem 8.2.2 and computational software, we find that

$$v_n = \frac{n}{2} + \frac{1}{4}\left(\psi\left(\left\lceil\frac{n}{2}\right\rceil + \frac{5}{4}\right) - \psi\left(\frac{5}{4}\right)\right).$$

The result then follows from Lemma 8.2.5 and a computation along the lines of the proof of Theorem 7.4.5. □

8.3 Quarters

In this section, we further explore the Nash values of synchronized Push positions, continuing with copies of the position

$$H_2 = 2 + \fbox{}\fbox{P}\fbox{P}$$

which has value $\frac{1}{4}$ in combinatorial sense. From H_2, Left plays to

$$H_2^L = 2 + \fbox{P}\fbox{}\fbox{P}$$

while Right plays to

$$H_2^R = 2 + \fbox{P}\fbox{P}\fbox{} = H$$

which is also the result of both players playing on H_2 simultaneously. In the previous section, we analyzed the result of playing on multiple copies of H, and showed that in the limit the Nash value converges to the combinatorial value. To ultimately prove the same behaviour of multiple copies of H_2, we start by analyzing H_2^L in more detail.

Lemma 8.3.1. *For all $n_1, n_2 \in \mathbb{N}$, we find $v(n_1H + n_2H_2^L) = v(n_1H)$.*

Proof. By Theorem 2.2.20, the probability of a player playing on one given copy of either of the games is equal across all these copies. This allows us to rewrite the game as a two-dimensional zero-sum game. We continue by induction on the birthday, the base cases for $n_1, n_2 \leq 2$ being straightforward computation.

If both players play on the same copy of H, the result is $(2-1) + (n_1 - 1)H + n_2H_2^L = 1 + (n_1 - 1)H + n_2H_2^L$. If the players play on different copies of H, the result is $(2-1) + (2-2) + (n_1 - 2)H + n_2H_2^L = 1 + (n_1 - 2)H + n_2H_2^L$. Hence, by induction, writing $v_n = v(nH)$ as before, the expected result of both parties playing on copies of H is

$$\tfrac{1}{n_1}(1 + v_{n_1-1}) + \tfrac{n_1-1}{n_1}(1 + v_{n_1-2}) = 1 + \tfrac{1}{n_1}v_{n_1-1} + \tfrac{n_1-1}{n_1}v_{n_1-2} = v_{n_1},$$

where the last equality follows from Theorem 8.2.1. If Left plays on a copy of H, while Right plays on a copy of H_2^L, the result is $(2-2) + (n_1 - 1)H + H + (n_2 - 1)H_2^L = n_1H + (n_2 - 1)H_2^L$, with value v_{n_1} by induction. If, conversely, Left plays on a copy of H_2^L and Right on a copy of H, the result is $(2-1) + (n_1 - 1)H + (2-3) + (n_2 - 1)H_2^L = (n_1 - 1)H + (n_2 - 1)H_2^L$ with value v_{n_1-1}.

Finally, if both players play on the same copy of H_2^L, the result is $n_1H + (2 - 2) + (n_2 - 1)H_2^L$, with value v_{n_1}. If they manage to play on different copies of H_2^L, the result is $n_1H + (2-3) + H + (n_2 - 2)H_2^L$, having value $v_{n_1+1} - 1$ by induction, Hence, both parties playing on H_2^L yields an average result of $\frac{1}{n_2}v_{n_1} + \frac{n_2-1}{n_2}(v_{n_1+1} - 1)$. We thus find that $n_1H + n_2H_2^L$ boils down to

$$\begin{pmatrix} v_{n_1} & v_{n_1} \\ v_{n_1-1} & \frac{1}{n_2}v_{n_1} + \frac{n_2-1}{n_2}(v_{n_1+1} - 1) \end{pmatrix}.$$

By Theorem 8.2.2, $v_{n_1} \geq v_{n_1-1}$ and $v_{n_1+1} - 1 \leq v_{n_1}$. Hence, the first row dominates, yielding a value of $v_{n_1} = v(n_1H)$. □

Lemma 8.3.2. *For $m, n \in \mathbb{N}$, let $v_{m,n} = v(mH_2 + nH)$. Then*

(i) $v_{m+1,n} \leq v_{m,n} + 1$;
(ii) $v_{m,n} \leq v_{m,n+1} \leq v_{m,n} + 1$;
(iii) $v_{m+1,n} \leq v_{m,n+1}$.

Proof. We prove all statements through Theorem 2.3.32.

(i) Consider $G = (m+1)H_2 + nH$, from which Right can play to $G^{RR} = mH_2 + (n-1)H + 1$ by moving on the same copy of H_2^L twice.

(ii) For the first inequality, consider $G = mH_2 + (n+1)H$, from which Left can move to $G^L = mH_2 + nH$. For the second inequality, consider $G = mH_2 + (n+1)H$, from which Right plays to $G^R = mH_2 + nH + 1$.
(iii) Consider $G = (m+1)H_2 + nH$ and note that Right can move to $G^R = mH_2 + (n+1)H$.

□

Lemma 8.3.3. *For all $n_1, n_2, n_3 \in \mathbb{N}$, we find $v(n_1H_2 + n_2H + n_3H_2^L) = v(n_1H_2 + n_2H)$.*

Proof. Let $G = n_1H_2 + n_2H + n_3H_2^L$. By induction on the birthday, using the previous lemmas, we compute G being equal to the zero-sum game given by the matrix

$$\begin{pmatrix} \begin{array}{c}\frac{1}{n_1}v_{n_1-1,n_2+1}\\ +\frac{n_1-1}{n_1}v_{n_1-2,n_2+1}\end{array} & v_{n_1-1,n_2-1}+1 & v_{n_1-1,n_2+1} \\ v_{n_1-1,n_2} & \begin{array}{c}\frac{1}{n_2}(v_{n_1,n_2-1}+1)\\ +\frac{n_2-1}{n_2}(v_{n_1,n_2-2}+1)\end{array} & v_{n_1,n_2} \\ v_{n_1-1,n_2+1}-1 & v_{n_1,n_2-1} & \begin{array}{c}\frac{1}{n_3}v_{n_1,n_2}\\ +\frac{n_3-1}{n_3}(v_{n_1,n_2+1}-1)\end{array} \end{pmatrix}$$

We first show that the first column dominates the third. Note that, from the game $(n_1-1)H_2+(n_2+1)H+n_3H_2^L$, Left can move to $(n_1-2)H_2+(n_2+1)H+(n_3+1)H_2^L$, which has value v_{n_1-2,n_2+1} by induction. Hence, $v_{n_1-2,n_2+1} \leq v_{n_1-1,n_2+1}$, so also $\frac{1}{n_1}v_{n_1-1,n_2+1} + \frac{n_1-1}{n_1}v_{n_1-2,n_2+1} \leq v_{n_1-1,n_2+1}$. By similar reasoning, also $v_{n_1-1,n_2} \leq v_{n_1,n_2}$ and $v_{n_1-1,n_2+1} \leq v_{n_1,n_2+1}$. By Lemma 8.3.2, $v_{n_1-1,n_2+1}-1 \leq v_{n_1-1,n_2}$, so that $v_{n_1-1,n_2+1}-1 \leq \frac{1}{n_3}v_{n_1,n_2}+\frac{n_3-1}{n_3}(v_{n_1,n_2+1}-1)$. Hence, the first column indeed dominates the third one.

Next, we show that the first row dominates the third. By Lemma 8.3.2, we find that $v_{n_1-2,n_2+1} \geq v_{n_1-1,n_2+1} - 1$, so also $\frac{1}{n_1}v_{n_1-1,n_2+1} + \frac{n_1-1}{n_1}v_{n_1-2,n_2+1} \geq v_{n_1-1,n_2+1} - 1$. Moreover, $v_{n_1-1,n_2-1} + 1 \geq v_{n_1,n_2-1}$. Hence, the first row indeed dominates the third.

By induction, play continues by making moves only on copies of H_2 and H until only terminal games and copies of H_2^L are left, which do not contribute to the value of the game by Lemma 8.3.1. We conclude that $v(n_1H_2+n_2H+n_3H_2^L) = v(n_1H_2 + n_2H)$. □

Note that, by the reasoning in the proof, we may conclude that $v_{m,n} \leq v_{m+1,n}$ for all $m, n \in \mathbb{N}$, extending Lemma 8.3.2. Moreover, recall that, for Blue-Red-

Hackenbush, a Nash equilibrium exists in which both players first play on copies of H_2, only turning to copies of H after the quarters have been exhausted. For Push, this is not the case, which makes the argument much less neat.

Lemma 8.3.4. *Let $G = H_2 + nH$ for $n \geq 4$. Then Left moving on H_2 and Right moving on any copy of H uniformly at random is a Nash equilibrium, with value $v(G) = v_{n-1} + 1$.*

Proof. We proceed by induction on n. The base cases $n = 4, 5$ can be checked using computational software to find the Nash equilibria of the games. Hence, suppose $n \geq 6$, in which case the game boils down to the zero-sum game given by

$$\begin{pmatrix} v_{n+1} & v_{n-1} + 1 \\ v_n & \frac{1}{n}(v_{1,n-1} + 1) + \frac{n-1}{n}(v_{2,n-2} + 1) \end{pmatrix},$$

where the last entry evaluates to $1 + \frac{1}{n}v_{1,n-1} + \frac{n-1}{n}v_{2,n-2}$. First, note that $v_{n+1} \geq v_n + \frac{1}{2} \geq v_{n-1} + 1$ by Theorem 8.2.2. Next, note that, by induction, $v_{1,n-1} = v_{n-2} + 1$ and $v_{2,n-1} = v_{n-3} + 1$. Hence,

$$\begin{aligned} 1 + \tfrac{1}{n}v_{1,n-1} + \tfrac{n-1}{n}v_{2,n-2} &= 1 + \tfrac{1}{n}(v_{n-2} + 1) + \tfrac{n-1}{n}(v_{n-3} + 1) \\ &= 2 + \tfrac{1}{n}v_{n-2} + \tfrac{n-1}{n}v_{n-3} \\ &\leq 2 + \tfrac{1}{n-1}v_{n-2} + \tfrac{n-2}{n-1}v_{n-3} \\ &= 1 + v_{n-1}, \end{aligned}$$

where the inequality follows from Theorem 8.2.2 and the last equality from Theorem 8.2.1. Hence, $v_{n-1} + 1$ is a saddle point. $\square$

Theorem 8.3.5. *Let $G = mH_2 + nH$ for $m \geq 11$ and $n \in \mathbb{N}$. Then Left and Right both moving on a copy of H_2 uniformly at random is a Nash equilibrium.*

Proof. Consider the game $G = mH_2 + nH$. We prove that the statement holds for all pairs (m, n) for which $n \geq \max\{1, 5 - \lfloor \frac{m-1}{2} \rfloor\}$ by induction on m. For the base cases, we prove the validity by using computational software to compute the Nash equilibria for the pairs $(m, n) \in \{(1, 5), (2, 5), (2, 4), (3, 4), (4, 4), (4, 3), (5, 3), (6, 3), (6, 2), (7, 2), (8, 2), (8, 1)\}$.

Now, let (m, n) be no base case, and such that $n \geq \max\{1, 5 - \lfloor \frac{m-1}{2} \rfloor\}$. Then $G = mH_2 + nH$ equals the zero-sum game

$$\begin{pmatrix} \frac{1}{m}v_{m-1,n+1} + \frac{m-1}{m}v_{m-2,n+1} & v_{m-1,n-1} + 1 \\ v_{m-1,n} & \frac{1}{n}(v_{m,n-1} + 1) + \frac{n-1}{n}(v_{m,n-2} + 1) \end{pmatrix}$$

by induction, where the last entry evaluates to $1+\frac{1}{n}v_{m,n-1}+\frac{n-1}{n}v_{m,n-2}$. We set out to show that the first entry is a saddle point. By the extended Lemma 8.3.2, we have $v_{m-1,n} \leq v_{m-2,n+1} \leq v_{m-1,n+1}$, so that $v_{m-1,n} \leq \frac{1}{m}v_{m-1,n+1} + \frac{m-1}{m}v_{m-2,n+1}$. Remains to show that $\frac{1}{m}v_{m-1,n+1}+\frac{m-1}{m}v_{m-2,n+1} \leq v_{m-1,n-1}+1$.

Consider

$$\begin{aligned} G_1 &= (m-1)H_2 + (n-1)H + 1, \\ G_2 &= (m-2)H_2 + (n+1)H, \\ G_3 &= (m-1)H_2 + (n+1)H. \end{aligned}$$

By induction, the Nash equilibrium strategy for both players is to play on copies of H_2 uniformly at random until none are left. We provide a coupling of the three games. For $2 \leq k \leq m$, define

$$Z_k = \begin{cases} 1 & \text{with probability } \frac{1}{k}, \\ 0 & \text{with probability } \frac{k-1}{k}, \end{cases}$$

and

$$X(k) = \begin{cases} 1 + X(k-1) & \text{with probability } \frac{1}{k}, \\ 1 + X(k-2) & \text{with probability } \frac{k-1}{k}. \end{cases}$$

Set $X(0) = 0$ and $X(1) = -1$. Then $X(k)$ models the number of copies of H obtained when starting play from kH_2 for k sufficiently large; note that if we end up with only one copy of H_2, this indeed costs a copy of H by Lemma 8.3.4. Now, for $i = 1, 2, 3$, set

$$X_i(k) = 1 + Z_k X_i(k-1) + (1 - Z_k) X_i(k-2).$$

We consider $X_1(m-1)$, $X_2(m-2)$ and $X_3(m-1)$ and let A_i be the event that X_i lands on 1. Then G_1, G_2 and G_3 will result in

$$\begin{aligned} G_1' &= (n-1+X_1(m-1))H + 1 + \mathbb{1}(A_i), \\ G_2' &= (n+1+X_2(m-2))H + \mathbb{1}(A_i), \\ G_3' &= (n+1+X_3(m-1))H + \mathbb{1}(A_i), \end{aligned}$$

and we have $v(G_i) = \mathbb{E}_{Z_2,\ldots,Z_{m-1}}[v(G_i')]$ for $i = 1, 2, 3$. Note that we may indeed omit copies of H_2^L if so created by Lemma 8.3.3. Define

$$\ell = \begin{cases} \max\{k \mid 2 \leq k \leq m-1, Z_k = 1\} & \text{if } \prod_{k=2}^{m-1}(1 - Z_k) = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Then ℓ tells us the first time that both players play on the same copy of H_2, with $\ell = 0$ signifying that this never happens. We distinguish between four cases.

First, suppose $\ell = 0$ and m is even. Then $X_1(m-1) = X_3(m-1) = \frac{m-2}{2} + X(1) = \frac{m-2}{2} - 1$ and $X_2(m-2) = \frac{m-2}{2} + X(0) = \frac{m-2}{2}$. Setting $n' = n + \frac{m-2}{2} - 1$, we thus find that, in this case, $v(G_1') = v_{n'-1} + 2$, $v(G_2') = v_{n'+2}$ and $v(G_3') = v_{n'+1} + 1$. By Lemma 8.2.2, we find

$$\begin{aligned}\tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= \tfrac{1}{m}(v_{n'+1} + 1) + \tfrac{m-1}{m}v_{n'+2}\\ &\le \tfrac{1}{2}(v_{n'+1} + 1) + \tfrac{1}{2}v_{n'+2}\\ &= \tfrac{1}{2}(v_{n'-1} + d_{n'} + d_{n'+1} + 1) + \tfrac{1}{2}(v_{n'-1} + d_{n'} + d_{n'+1} + d_{n'+2})\\ &= v_{n'-1} + d_{n'} + d_{n'+1} + \tfrac{1}{2}d_{n'+2} + \tfrac{1}{2}.\end{aligned}$$

For $n' = 2$, we have

$$d_2 + d_3 + \tfrac{1}{2}d_4 + \tfrac{1}{2} = \tfrac{1}{2} + \tfrac{2}{3} + \tfrac{1}{2}\cdot\tfrac{1}{2} + \tfrac{1}{2} = \tfrac{23}{12} < 2,$$

and $n' = 3$ yields

$$d_2 + d_3 + \tfrac{1}{2}d_4 + \tfrac{1}{2} = \tfrac{2}{3} + \tfrac{1}{2} + \tfrac{1}{2}\cdot\tfrac{3}{5} + \tfrac{1}{2} = \tfrac{59}{30} < 2.$$

Noting that $d_{2j} = \frac{1}{2}$ and d_{2j+1} is decreasing in j, we conclude that $d_{n'} + d_{n'+1} + \frac{1}{2}d_{n'+2} + \frac{1}{2} \le 2$ for all n', so that

$$\tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') \le v_{n'-1} + 2 = v(G_1').$$

Second, suppose $\ell = 0$ and m is odd. Then $X_1(m-1) = X_3(m-1) = \frac{m-1}{2} + X(0) = \frac{m-1}{2}$ and $X_2(m-2) = \frac{m-1}{2} + X(1) - 1 = \frac{m-1}{2} - 2$. Hence, now setting $n' = n + \frac{m-1}{2}$, we find $v(G_1') = v_{n'-1} + 1 = v(G_2')$ and $v(G_3') = v_{n'+1}$. Again by Theorem 8.2.2,

$$\begin{aligned}\tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= \tfrac{1}{m}v_{n'+1} + \tfrac{m-1}{m}(v_{n'-1} + 1)\\ &\le \tfrac{1}{3}v_{n'+1} + \tfrac{2}{3}(v_{n'-1} + 1)\\ &= \tfrac{1}{3}(v_{n'-1} + d_{n'} + d_{n'+1} + \tfrac{2}{3}(v_{n'-1} + 1)\\ &= v_{n'-1} + \tfrac{1}{3}d_{n'} + \tfrac{1}{3}d_{n'+1} + \tfrac{2}{3}.\end{aligned}$$

By similar reasoning as above, for $n' \ge 2$, we find

$$\tfrac{1}{3}d_{n'} + \tfrac{1}{3}d_{n'+1} + \tfrac{2}{3} \le \tfrac{1}{3}\cdot\tfrac{1}{2} + \tfrac{1}{3}\cdot\tfrac{2}{3} + \tfrac{2}{3} = \tfrac{19}{18} < \tfrac{11}{10},$$

so that

$$\tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') \le v_{n'-1} + \tfrac{11}{10} = v(G_1') + \tfrac{1}{10}.$$

Third, suppose $\ell \geq 2$ and $\ell \equiv m-1 \bmod 2$. Then $X_1(m-1) = X_3(m-1) = X_2(m-2)+1$, as the processes meet. Moreover, $A_1 = A_2 = A_3$. Hence, setting $A = A_1$ and $N = n + X(m-1)$, we find

$$\begin{aligned} v(G_1') &= v_{N-1} + 1 + \mathbb{1}(A), \\ v(G_2') &= v_N + \mathbb{1}(A), \\ v(G_3') &= v_{N+1} + \mathbb{1}(A). \end{aligned}$$

We compute

$$\begin{aligned} \tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= \tfrac{1}{m}(v_{N+1} + \mathbb{1}(A)) + \tfrac{m-1}{m}(v_N + \mathbb{1}(A)) \\ &\leq \tfrac{1}{2}(v_{N+1} + \mathbb{1}(A)) + \tfrac{1}{2}(v_N + \mathbb{1}(A)) \\ &= \tfrac{1}{2}(v_{N-1} + d_N + d_{N+1} + \mathbb{1}(A)) + \tfrac{1}{2}(v_{N-1} + d_N + \mathbb{1}(A)) \\ &= v_{N-1} + d_N + \tfrac{1}{2}d_{N+1} + \mathbb{1}(A). \end{aligned}$$

We compute

$$d_4 + \tfrac{1}{2}d_5 = \tfrac{4}{5} \quad \text{and} \quad d_5 + \tfrac{1}{2}d_6 = \tfrac{17}{20},$$

so that, noting that $d_{2J} + \frac{1}{2}d_{2J+1}$ and $d_{2J+1} + \frac{1}{2}d_{2J+2}$ are decreasing in J,

$$\begin{aligned} \tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= v_{N-1} + d_N + \tfrac{1}{2}d_{N+1} + \mathbb{1}(A) \\ &\leq v_{N-1} + \tfrac{9}{10} + \mathbb{1}(A) \\ &= v(G_1') - \tfrac{1}{10} \end{aligned}$$

for $N \geq 4$.

Fourth, and finally, suppose $\ell \geq 2$ and $\ell \equiv m \bmod 2$. Defining A and N as above, we find $v(G_1')$ and $v(G_3')$ arrive at the same values, the only difference being that in this case $v(G_2') = v_{N+1} + \mathbb{1}(A)$. We thus compute

$$\begin{aligned} \tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= \tfrac{1}{m}(v_{N+1} + \mathbb{1}(A)) + \tfrac{m-1}{m}(v_{N+1} + \mathbb{1}(A)) \\ &= v_{N+1} + \mathbb{1}(A) \\ &= v_{N-1} + d_N + d_{N+1} + \mathbb{1}(A). \end{aligned}$$

We compute $d_4 + d_5 = \frac{11}{10}$, so that

$$\begin{aligned} \tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2') &= v_{N-1} + d_N + d_{N+1} + \mathbb{1}(A) \\ &\leq v_{N-1} + \tfrac{11}{10} + \mathbb{1}(A) \\ &= v(G_1') + \tfrac{1}{10}. \end{aligned}$$

In the cases $\ell \geq 2$, the statements only hold for $N = n + X_1(m-1) \geq 4$, i.e., $n \geq 4 - X_1(m-1)$. Noting that $X_1(m-1) \geq \lfloor \frac{m-1}{2} \rfloor - 1$, we find that the statements hold for $n \geq 5 - \max\{1, 5 - \lfloor \frac{m-1}{2} \rfloor\}$, which was the assumption.

Next, we compute the relevant probabilities. As the Z_k are independent, for $k \geq 2$ we compute

$$\mathbb{P}(\ell = k) = \mathbb{P}(Z_k = 1) \prod_{j=k+1}^{m-1} \mathbb{P}(Z_j = 0) = \frac{1}{k} \prod_{j=k+1}^{m-1} \frac{j}{j+1} = \frac{1}{m-1}.$$

For $\ell = 0$, we compute

$$\mathbb{P}(\ell = 0) = 1 - \mathbb{P}(\ell \geq 2) = 1 - \sum_{k=2}^{m-1} \mathbb{P}(\ell = k) = 1 - \frac{m-2}{m-1} = \frac{1}{m-1}.$$

Moreover, using these probabilities, we compute

$$\mathbb{P}(\ell \geq 2, \ell \equiv m-1 \bmod 2) = \begin{cases} \frac{m-2}{2} \cdot \frac{1}{m-1}, & \text{if } m \text{ is even,} \\ \frac{m-1}{2} \cdot \frac{1}{m-1}, & \text{if } m \text{ is odd,} \end{cases}$$

and

$$\mathbb{P}(\ell \geq 2, \ell \equiv m \bmod 2) = \begin{cases} \frac{m-2}{2} \cdot \frac{1}{m-1}, & \text{if } m \text{ is even,} \\ (\frac{m-1}{2} - 1) \cdot \frac{1}{m-1}, & \text{if } m \text{ is odd,} \end{cases}$$

Hence, for m even, we compute

$$\begin{aligned}
&\mathbb{E}_{Z_2,\ldots,Z_{m-1}}[\tfrac{1}{m} v(G_3') + \tfrac{m-1}{m} v(G_2')] \\
&\quad \leq \mathbb{P}(\ell = 0) v(G_1') + \mathbb{P}(\ell \geq 2, \ell \equiv m-1 \bmod 2)(v(G_1') - \tfrac{1}{10}) \\
&\qquad + \mathbb{P}(\ell \geq 2, \ell \equiv m \bmod 2)(v(G_1') + \tfrac{1}{10}) \\
&\quad = \tfrac{1}{m-1} v(G_1') + \tfrac{m-2}{2} \cdot \tfrac{1}{m-1}(v(G_1') - \tfrac{1}{10}) \\
&\qquad + \tfrac{m-2}{2} \cdot \tfrac{1}{m-1}(v(G_1') + \tfrac{1}{10}) \\
&\quad = (\tfrac{1}{m-1} + \tfrac{m-2}{m-1}) v(G_1') = v(G_1').
\end{aligned}$$

For m odd, we compute

$$\begin{aligned}
&\mathbb{E}_{Z_2,\ldots,Z_{m-1}}[\tfrac{1}{m}v(G_3') + \tfrac{m-1}{m}v(G_2')] \\
&\quad \le \mathbb{P}(\ell = 0)(v(G_1') + \tfrac{1}{10}) + \mathbb{P}(\ell \ge 2, \ell \equiv m-1 \bmod 2)(v(G_1') - \tfrac{1}{10}) \\
&\qquad + \mathbb{P}(\ell \ge 2, \ell \equiv m \bmod 2)(v(G_1') + \tfrac{1}{10}) \\
&\quad = \tfrac{1}{m-1}(v(G_1') + \tfrac{1}{10} + \tfrac{m-1}{2} \cdot \tfrac{1}{m-1}(v(G_1') - \tfrac{1}{10}) \\
&\qquad + (\tfrac{m-1}{2} - 1) \cdot \tfrac{1}{m-1}(v(G_1') + \tfrac{1}{10}) \\
&\quad = v(G_1').
\end{aligned}$$

Hence, indeed $\frac{1}{m}v_{m-1,n+1} + \frac{m-1}{m}v_{m-2,n+1} \le v_{m-1,n-1} + 1$, so the top-left entry of the matrix is a saddle point. □

From the proof of Theorem 8.3.5, it follows that, for m and n large enough,

$$v_{m,n} = \tfrac{1}{m}v_{m-1,n+1} + \tfrac{m-1}{m}v_{m-2,n+1}.$$

Defining $d_{m,n} = v_{m,n} - v_{m-1,n}$, and using the base cases written in the aforementioned proof, one may readily compute values for $v_{m,0} = v(mH_2)$ and $d_{m,0} = v(mH_2) - v((m-1)H_2)$. Some results are shown in Figure 8.2.

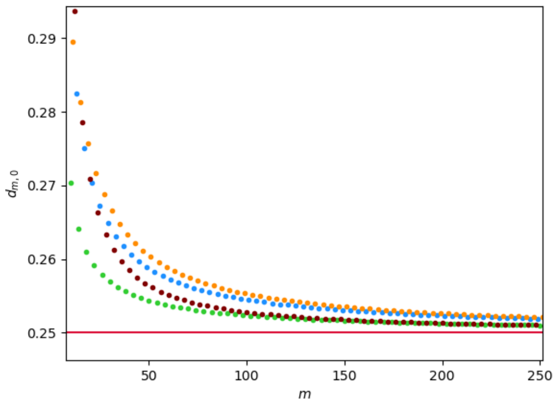


Figure 8.2: Values of $d_{m,0}$ for $10 \le m \le 250$. The line $y = \frac{1}{4}$ is drawn in red, and points are given a color based on their index modulo 4.

We see that, just like the clearly distinguishable subsequences d_{2k} and d_{2k+1} when looking at copies of H, we now find four decreasing subsequences, converging from above to $\frac{1}{4}$.

Theorem 8.3.6. *We have*

$$\lim_{m\to\infty}(v(mH_2) - v((m-1)H_2)) = \tfrac{1}{4}.$$

Proof. For $n \in \mathbb{N}$, define the stochastic process $Y(n)$ by $Y(0) = 0$, $Y(1) = 1$, and, for $n \geq 2$,

$$Y(n) = \begin{cases} 1 + Y(n-1) & \text{with probability } \frac{1}{n}, \\ 1 + Y(n-2) & \text{with probability } \frac{n-1}{n}. \end{cases}$$

The process $Y(n)$ models the amount of copies of 1 obtained when starting play from nH. Hence, $\mathbb{E}[Y_n] = v(nH)$ and, by Theorem 8.2.2, $\mathbb{E}[Y(n) - \tilde{Y}(n)] \to \frac{1}{2}$ for any two copies $Y, \tilde{Y}$ of the process.

Next, define $X(m)$, A, Z_k for $k = 2, \ldots, m$, and ℓ as in the proof of Theorem 8.3.5. Define two copies of each process, X and $\tilde{X}$, and Y and $\tilde{Y}$, coupling them through

$$Y(k) = 1 + Z_k Y(k-1) + (1 - Z_k)Y(k-2)$$

and

$$\tilde{Y}(k) = 1 + Z_k\tilde{Y}(k-1) + (1 - Z_k)\tilde{Y}(k-2),$$

and analogous definitions for X and $\tilde{X}$. We will prove that

$$\lim_{m\to\infty}\mathbb{E}[Y(X(m)) + \mathbb{1}(A) - \tilde{Y}(\tilde{X}(m)) - 1(\tilde{A})] = \tfrac{1}{4}.$$

First, note that, by the proof of Theorem 8.3.5, $\mathbb{P}(\ell = 0) = \frac{1}{m} \to 0$, so that also $\mathbb{E}[\mathbb{1}(A) - \mathbb{1}(\tilde{A})] = 0$. Next, we consider the difference $Y(k) - \tilde{Y}(k-1)$. If $k \equiv \ell \bmod 2$, then $Y(k) = \frac{k-\ell}{2} + 1 + Y(\ell - 1)$ and $\tilde{Y}(k-1) = \frac{k-\ell-1}{2} + Y(\ell-1)$. If $k \equiv \ell - 1 \bmod 2$, then $Y(k) = \frac{k-\ell-1}{2} + 1 + Y(\ell-1)$ and $\tilde{Y}(k-1) = \frac{k-\ell-1}{2} + 1 + Y(\ell-1)$. Hence, we find that $Y(k) - \tilde{Y}(k-1) \in \{0, 1\}$ for both cases, so that

$$\mathbb{P}(Y(n) - \tilde{Y}(n-1) = 1) = \mathbb{E}[Y(n) - \tilde{Y}(n-1)] \to \tfrac{1}{2}.$$

By similar reasoning, the same holds for X and $\tilde{X}$. Hence, we compute

$$\begin{aligned}
&\lim_{m\to\infty} \mathbb{E}[Y(X(m)) + \mathbb{1}(A) - \tilde{Y}(\tilde{X}(m-1)) - \mathbb{1}(\tilde{A})] \\
&\quad = \lim_{m\to\infty} \mathbb{E}[Y(X(m)) - \tilde{Y}(\tilde{X}(m-1))] \\
&\quad = \lim_{m\to\infty} \Big(\mathbb{E}[Y(X(m)) - \tilde{Y}(\tilde{X}(m-1)) \mid X(m) - \tilde{X}(m-1) = 1] \\
&\qquad\qquad\qquad\qquad \cdot \mathbb{P}(X(m) - \tilde{X}(m-1) = 1) \\
&\qquad\quad + \mathbb{E}[Y(X(m)) - \tilde{Y}(\tilde{X}(m-1)) \mid X(m) - \tilde{X}(m-1) = 0] \\
&\qquad\qquad\qquad\qquad \cdot \mathbb{P}(X(m) - \tilde{X}(m-1) = 0)\Big) \\
&\quad = \tfrac{1}{2} \cdot \tfrac{1}{2} + 0 \cdot \tfrac{1}{2} \\
&\quad = \tfrac{1}{4}.
\end{aligned}$$

The reasoning only holds if $X(m), Y(m) \to \infty$ for $m \to \infty$, which is indeed the case. □

For higher powers of $\frac{1}{2}$, the difficulties encountered in the proof of Theorem 8.3.5 become critical. While for Red-Blue Hackenbush, both players always play on the highest power of $\frac{1}{2}$, and this is also almost always true if only copies of H and H_2 are available for Push, such pure Nash equilibria fail to exist for higher powers of $\frac{1}{2}$ in Push. This makes theoretical analysis difficult. However, experimental results do support the conjecture that a statement along the lines of Theorem 8.3.6 holds in a more general sense.

8.4 Shove

We conclude with some preliminary results for the game of Shove. The combinatorial version is fully solved; the following result is from [2].

Theorem 8.4.1. *Consider a single strip of Shove G containing n pieces. Let $p(i)$ be the position of the i-th piece from the left of the strip, and let $c(i) = 1$ if the i-th piece is blue and -1 otherwise. Let $r(i)$ be the number of pieces to the right of the i-th piece up to and including both pieces of the last color alternation, setting $r(i) = 0$ if the i-th piece and all pieces to the right are of the same color. Then*

$$G = \sum_{i=1}^{n} c(i) \frac{p(i)}{2^{r(i)}}.$$

For the synchronized version, we consider copies of the game

$$K_n = \boxed{}\boxed{S} + \underbrace{\boxed{S}\boxed{S}\cdots\boxed{S}}_{n}\boxed{S}$$

which has combinatorial value 2^{-n} by Theorem 8.4.1. For $n = 1$, the resulting game is isomorphic to H, and therefore the same results hold. For $n = 2$, some computational results are shown in Figure 8.3.

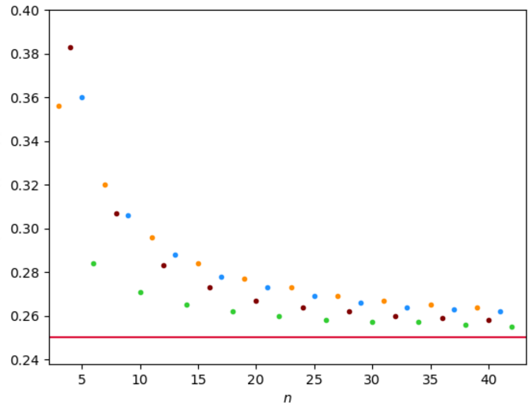


Figure 8.3: The difference $v(nK_2 - (n-1)K_2)$ for $k = 3, \ldots, 42$. The line $y = \frac{1}{4}$ is drawn in red, and points are given a color based on their index modulo 4.

Just like for higher powers of $\frac{1}{2}$ in Push, copies of K_2 (and also K_n for $n > 2$) display non-deterministic Nash equilibria, making the analysis difficult. However, Figure 8.3 suggest that also for this game, a result similar to Theorem 8.3.6 holds.

Bibliography

[1] E. Berlekamp, J. Conway, and R. Guy, *Winning Ways for Your Mathematical Plays, second edition*. Academic Press, 2014.

[2] R. Nowakowski, D. Wolfe, and M. Albert, *Lessons in Play, second edition*. CRC Press, 2019.

[3] A. Siegel, *Combinatorial Game Theory*. American Mathematical Society, 2013.

[4] J. Conway, *On Numbers and Games*. CRC Press, 1976.

[5] N. Elkies, "On numbers and endgames: Combinatorial game theory in chess endgames," *Games of No Chance*, vol. 29, pp. 135–150, 1996.

[6] E. Berlekamp and Y. Kim, "Where is the "thousand-dollar ko"?" *Games of No Chance*, vol. 29, pp. 203–226, 1996.

[7] A. Cincotti, "N-player partizan games," *Theoretical Computer Science*, vol. 411, no. 34, pp. 3224–3234, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397510003105

[8] K. Suetsugu, "Multiplayer games as extension of misère games," *International Journal of Game Theory*, vol. 48, no. 3, pp. 781–796, 2019.

[9] J. M. Ettinger, "Topics in combinatorial games," PhD Thesis, University of Wisconsin–Madison, 1996.

[10] P. Ciancarini and G. Favini, "Monte carlo tree search in kriegspiel," in *Artificial Intelligence*, vol. 174, 2010, pp. 670–684.

[11] P. Ciancarini and G. P. Favini, "Representing kriegspiel states with metapositions," in *Proceedings of the 20th International Joint Conference on Artifical*

Intelligence, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, p. 2450–2455.

[12] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Canbridge University Press, 2007.

[13] M. Huggan, R. Nowakowski, and P. Ottaway, "Simultaneous combinatorial game theory," *arXiv: Combinatorics*, 2018.

[14] N. Burgers, "Childish hackenbush," Bachelor's Thesis, Leiden University, 2020.

[15] M. J. H. van den Bergh, S. T. Castelein, and D. M. H. van Gent, "Order versus chaos," in *2020 IEEE Conference on Games (CoG)*, 2020, pp. 391–398.

[16] S. Sniderman, "Games magazine," 9 1981.

[17] M. van den Bergh, W. Kosters, and F. Spieksma, "Nim variants," *ICGA Journal*, vol. 44, no. 1, pp. 2—-17, 2022.

[18] T. de Mol, "Synchronized cherries," Bachelor's Thesis, Leiden University, 2021.

[19] R. Takken, "Synchronized push and shove games," Master's Thesis, Leiden University, 2021.

[20] T. S. Ferguson, "Game theory," http://www.inf.ufsc.br/~joao.dovicchi/pos-ed/pos/games/comb.pdf, accessed 7/4/21.

[21] A. Cincotti and H. Iida, "The game of synchronized domineering," in *Conference on Computers and Games 2008*, 2008, pp. 241–251.

[22] ——, "The game of synchronized cutcake," in *2007 IEEE Symposium on Computational Intelligence and Games*, 05 2007, pp. 374 – 379.

[23] A. Cincotti, S. Komori, and H. Iida, "The game of synchronized triomineering and synchronized tridomineering," *International Journal of Computational and Mathematical Sciences*, vol. 2, 01 2008.

[24] M. Piccione and A. Rubinstein, "On the interpretation of decision problems with imperfect recall," in *Games and Economic Behavior*, vol. 20, 1997, pp. 3–24.

[25] H. Kuhn, "Extensive games and the problem of information," in *Annals of Mathematical Studies*, vol. 28, 1953, pp. 193–216.

[26] J. Nash, "Non-cooperative games," in *Annals of Mathematics*, vol. 54, 1951, pp. 286–295.

[27] C. Lemke and J. Howson, "Equilibrium points of bimatrix games," in *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, 1964, pp. 413–423.

[28] R. Bödi, K. Herr, and M. Joswig, "Algorithms for highly symmetric linear and integer programs," in *Mathematical Programming*, vol. 137, no. 1–2, 2013, pp. 65–90.

[29] T. van Roode, "Partizan forms of hackenbush combinatorial games," Master's Thesis, University of Calgary, 2002.

[30] S. T. Castelein, "Order versus chaos," Bachelor's Thesis, Leiden University, 2019.

[31] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, *Positional games*. Springer, 2014.

[32] B. Turner, "solved: Order wins," https://boardgamegeek.com/thread/1579397/solved-order-wins, 2016, accessed 13/3/20.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016. [Online]. Available: http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

[34] O. Ohrimenko, P. J. Stuckey, and M. Codish, "Propagation = lazy clause generation," in *Principles and Practice of Constraint Programming – CP 2007*, C. Bessière, Ed. Springer, 2007, pp. 544–558.

[35] S. Sniderman, "order and chaos solved?" https://boardgamegeek.com/thread/1744598/order-and-chaos-solved, 2017, accessed 27/3/20.

[36] A. Biere, "PicoSAT essentials," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 4, no. 2-4, pp. 75–97, 2008.

[37] P. Erdös and J. Selfridge, "On a combinatorial game," *Journal of Combinatorial Theory, Series A*, vol. 14, no. 3, pp. 298 – 301, 1973. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0097316573900058

[38] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck, "Monte-Carlo tree search: A new framework for game AI." in *Artificial Intelligence for Interactive Digital Entertainment*, 2008, pp. 216–217.

[39] M. H. M. Winands, "Monte-carlo tree search," in *Encyclopedia of Computer Graphics and Games*, N. Lee, Ed. Springer, 2015, pp. 1–6. [Online]. Available: https://doi.org/10.1007/978-3-319-08234-9_12-1

[40] M. Bowling, N. Burch, M. Johanson, and O. Tammelin, "Heads-up limit hold'em poker is solved," *Science*, vol. 60, no. 11, pp. 81—-88, oct 2017. [Online]. Available: https://doi.org/10.1145/3131284

[41] M. Moravčík, M. Schmid, N. Burch, V. Lisý, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling, "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aam6960

[42] B. von Stengel, "Computing equilibria for two-person games," 1998.

[43] F. Ruskey. (2006) Information on necklaces, unlabelled necklaces, lyndon words, de bruijn sequencesn. [Online]. Available: https://web.archive.org/web/20061002130346/http://www.theory.csc.uvic.ca/~cos/inf/neck/NecklaceInfo.html

[44] N. J. A. Sloane and T. O. F. Inc., "The on-line encyclopedia of integer sequences," 2020. [Online]. Available: http://oeis.org/?language=english

[45] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," *SIAM Journal on Computing*, vol. 22, no. 5, pp. 935–948, 1993. [Online]. Available: https://doi.org/10.1137/0222058

[46] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch, "Replacing suffix trees with enhanced suffix arrays," *Journal of Discrete Algorithms*, vol. 2, no. 1, pp. 53–86, 2004, the 9th International Symposium on String Processing and Information Retrieval. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570866703000650

[47] Z. Li, J. Li, and H. Huo, "Optimal in-place suffix sorting," *Inf. Comput.*, vol. 285, no. PB, may 2022. [Online]. Available: https://doi.org/10.1016/j.ic.2021.104818

Summary

Combinatorial games are games for two competing players, moving in a turn-by-turn fashion, in which there is no chance nor hidden information. Chess, checkers and the simpler tic tac toe are well-known examples of this class of games, as well as game of go. Though these games are by no means simple, there does exist a beautiful mathematical framework for their analysis. Using this theory, it is possible to efficiently determine the outcome of a given position of a game without having to explicitly compute the results for every possible move. Moreover, the theory provides a measure of how profitable a given position is to either player, often denoted by the 'value' of a position. An example application of the theory is research on endgames in go.

However, not all games are combinatorial games. The game of poker, for example, introduces hidden information. In practice, impressive results have been obtained for these non-combinatorial games using artificial intelligence, but theory and understanding are perhaps lacking. In this thesis, the main question we address is whether the existing theory for combinatorial games can be adapted or extended to non-combinatorial ones.

We will look at two types of variants of combinatorial games in particular. For the first type, we introduce hidden information in a given combinatorial game by no longer communicating the details of the current position to the opponent after a move has been made. A variant of this type for the game of chess is called Kriegspiel in the literature. Besides the two players, we now also need a referee to whom the players can communicate their prospective moves. If the move is legal, it is executed, and the other player may or may not be informed about any or all aspects of the move that was made. If not, the player has to try another move. By attempting different moves until a legal one is executed, a

player can obtain information about the current state of the game.

The second way of altering an existing combinatorial game is by making the players move simultaneously instead of turn by turn. We call this type of variant the synchronized version of a game. Again, a referee is appointed, to whom the players may communicate their prospective moves. After they have both done so, the moves are executed simultaneously. In practice, this might cause problems, as not all combinations of moves will allow for a natural way of them being executed at the same time. What should happen, for example, if two players want to move their chess pieces to the same unoccupied square?

We start with a concise introduction in Chapter 1. Next, in Chapter 2, we provide some of the fundamental theory for combinatorial games, as well as for the two types of variants as introduced above. We do not only provide existing results for analyzing combinatorial games and games with hidden information, but also introduce a new framework for analyzing synchronized games. We state some properties which are sufficient for a combinatorial game to allow for a synchronized version, and redefine the concept of value for synchronized games.

In Chapters 3 and 4, we analyze a handful of combinatorial games. In Chapter 3, we consider two variants of the game of Hackenbush. We identify which positions are easy and which are hard to analyze, and conclude that the results more or less coincide with those for the original game. In Chapter 4, we consider a variant of tic tac toe called Order versus Chaos. In this asymmetrical game, one player, Order, attempts to construct a line of identical symbols, while the other, Chaos, tries to prevent this. We show that some positions are always won by Order, and that many positions are always won by Chaos. For some positions, we do not find the winner. For these positions, we formulate a conjecture for who the winner should be by using artificial intelligence.

In Chapter 5, we consider three variants of the combinatorial game of Nim with hidden information. A position in this game consists of heaps of coins. A turn consists of picking one of the heaps and removing any number of coins from this heap. If a player finds no more coins to remove on their turn, they lose the game. We introduce two variants in which the opponent only receives partial information on the amount of coins that has been removed by a player during a turn. One variant turns out to have an elegant solution; the other seems to be complicated. Finally, we consider a Kriegspiel type variant, in which the players have even less information on the current state of the game.

We compute the value for some small positions.

In the final three chapters, we turn to synchronized variants of combinatorial games. In Chapter 6, we treat the game Cherries and a variant of this game called Stack Cherries. We show that every Stack Cherries position can be efficiently decomposed into basic building blocks. By looking at this decomposition, it is easy to find out who the winning player is in any given position. We conclude with the conjecture that every Cherries position can be decomposed in a similar way.

In Chapter 7, we consider the synchronized version of Hackenbush. We show that, for a specific type of position, many copies of this position set next to each other behave almost identically in the synchronized and combinatorial versions of the game. We conjecture that this statement holds more generally for a certain class of synchronized combinatorial games. In Chapter 8, we further explore this conjecture by considering synchronized versions of the games of Push and Shove.

Samenvatting

Combinatorische spellen zijn spellen waarin twee spelers om de beurt een zet doen, waarbij kans en verborgen informatie geen rol spelen. Welbekende voorbeelden hiervan zijn schaken, dammen en het simpelere boter, kaas en eieren. Ook het van oorsprong Chinese spel go valt in deze categorie. Hoewel deze spellen zeker niet eenvoudig zijn, bestaat er wel een mooie wiskundige machinerie om ze te analyseren. Met deze theorie valt voor een gegeven positie van een spel snel te bepalen wie het spel zal winnen als er optimaal wordt gespeeld, zonder daadwerkelijk alle mogelijke zetten uit te rekenen. Ook bevat de theorie een maat voor in hoeverre een positie een van beide spelers bevoordeelt, vaak de 'waarde' van een positie genoemd. Met behulp van deze theorie is bijvoorbeeld onderzoek gedaan naar eindspellen in go.

Niet alle spellen die we kennen zijn echter combinatorische spellen. Het spel poker bijvoorbeeld voldoet niet aan de eis van het niet hebben van onvolledige informatie. In de praktijk zijn er met behulp van kunstmatige intelligentie indrukwekkende resultaten geboekt voor dergelijke spellen, maar het ontbreekt wellicht aan theoretisch begrip. In dit proefschrift stellen we de vraag centraal in hoeverre het wiskundige raamwerk voor combinatorische spellen in stand gehouden kan worden voor spellen die niet combinatorisch zijn.

In het bijzonder zullen we kijken naar twee soorten varianten van bestaande combinatorische spellen. Voor de eerste soort introduceren we onvolledige informatie in een combinatorisch spel door na een zet niet meer alle details van de bereikte positie aan de tegenstander door te geven. Een dergelijke variant van schaken wordt in de literatuur Kriegspiel genoemd. Naast de twee spelers is er nu ook een onafhankelijke scheidsrechter bij het spel betrokken aan wie een speler diens beoogde zet kan doorgeven. Als deze mogelijk is, wordt de zet uitgevoerd en krijgt de andere speler hier al dan niet summiere informatie over; anders moet er een nieuwe poging worden gewaagd. Door zetten uit te proberen kunnen de spelers zo informatie verkrijgen over de huidige toestand van het spel.

De tweede manier waarop we combinatorische spellen zullen aanpassen is door de spelers tegelijk te laten spelen in plaats van om de beurt. Deze variant noemen we de gesynchroniseerde versie van een spel. Wederom wordt er een scheidsrechter ingeschakeld, aan wie beide spelers tegelijk hun boogde zet doorgeven. Deze zetten worden vervolgens tegelijk uitgevoerd. Een probleem dat hierbij kan ontstaan is dat niet alle mogelijke combinaties van zetten zonder meer tegelijk uitgevoerd kunnen worden. Wat moet er bijvoorbeeld gebeuren als beide spelers een schaakstuk naar hetzelfde lege veld willen verplaatsen?

Na een beknopte introductie in Hoofdstuk 1 beginnen we in Hoofdstuk 2 met een uiteenzetting van de fundamentele theorie voor combinatorische spellen en voor de twee soorten varianten die hierboven zijn genoemd. We bekijken hierbij niet alleen de bestaande oplossingsmethoden voor combinatorische spellen en spellen met onvolledige informatie, maar introduceren ook een nieuw wiskundig raamwerk voor het analyseren van gesynchroniseerde spellen. Hierbij stellen we eisen op waaraan een combinatorisch spel moet voldoen om zonder meer gesynchroniseerd te kunnen worden en herdefiniëren we het begrip van waarde van een positie voor dit soort spellen.

In Hoofdstukken 3 en 4 beschouwen we een aantal combinatorische spellen. In Hoofdstuk 3 kijken we naar twee varianten van het spel Hackenbush. We identificeren welke posities eenvoudig zijn om te analyseren en welke ingewikkeld, en concluderen dat de resultaten in grote lijnen overeenkomen met de bestaande resultaten voor het originele spel. In Hoofdstuk 4 bekijken we een variant van boter, kaas en eieren, genaamd Orde versus Chaos. Bij dit asymmetrische spel probeert de ene speler Orde een rij van dezelfde symbolen te maken, terwijl de andere speler Chaos dit juist probeert te verhinderen. We laten zien dat sommige posities altijd gewonnen worden door Orde, en dat veel posities altijd gewonnen worden door Chaos. Voor sommige posities vinden we geen uitsluitsel, en gebruiken we kunstmatige intelligentie om een vermoeden op te stellen voor wie deze posities zou moeten winnen.

In Hoofdstuk 5 bekijken we drie varianten van het combinatorische spel Nim met onvolledige informatie. Een positie van dit spel bestaat uit een aantal stapels muntjes. Als een speler aan de beurt is, kiest die één stapel, waarvan een aantal munten naar keuze wordt verwijderd. Kan een speler geen munten meer verwijderen, dan verliest die het spel. We introduceren twee varianten waarbij de andere speler niet volledig te horen krijgt hoeveel munten de ander verwijderd heeft. De ene variant blijkt een elegante oplossing te hebben, terwijl de analyse van de tweede variant lastig blijkt. Tenslotte bekijken we een variant in de geest van Kriegspiel, waarbij de spelers nog minder informatie hebben over de huidige toestand van het spel. We rekenen een aantal kleine posities door.

In de laatste drie hoofdstukken richten we ons tenslotte op gesynchroniseerde varianten van combinatorische spellen. In Hoofdstuk 6 gaat het over het spel Cherries en een variant hiervan, Stack Cherries. We laten zien dat iedere positie van gesynchroniseerd Stack Cherries op een efficiënte manier ontleed kan worden in zekere basisblokken. Door vervolgens naar de configuratie van deze blokken te kijken, is het eenvoudig te achterhalen wie de gegeven positie wint bij optimaal spel. We sluiten af met het vermoeden dat een dergelijke ontleding ook voor gesynchroniseerd Cherries bestaat.

In Hoofdstuk 7 kijken we naar de gesynchroniseerde versie van Hackenbush. We laten zien dat voor een bepaald type positie geldt dat veel kopieën van deze posities naast elkaar zich bijna net zo gedragen als bij het combinatorische spel. We formuleren het vermoeden dat dit algemener geldt voor bepaalde gesynchroniseerde combinatorische spellen. Dit vermoeden toetsen we verder in Hoofdstuk 8, waarbij we kijken naar gesynchroniseerde versies van Push en Shove.

Acknowledgements

Allereerst wil ik mijn begeleiders bedanken voor dat ze me al sinds het schrijven van mijn bachelorscriptie onder hun hoede hebben genomen. We hebben veel gezellige bijeenkomsten gehad waarbij we gemiddeld genomen vaker over de algehele staat van de wereld hebben gediscussieerd dan over de inhoud van het onderzoek. Floske, bedankt voor alle mogelijkheden die je me hebt geboden, niet alleen in het onderzoek, maar ook in het onderwijs. Ik heb enorm veel plezier beleefd aan het opzetten van het vak Combinatorial Game Theory en het begeleiden van studenten met het schrijven van hun scripties. Walter, op eigen verzoek zal ik je niet bedanken voor de begeleiding tijdens de promotie. Wel enorm bedankt voor alle gezellige gesprekken en de ondersteuning buiten het noodzakelijke om, vooral in de laatste maanden van het traject.

Bij de totstandkoming van dit proefschrift heb ik nauw samengewerkt met een aantal studenten wiens bachelor- en masterscripties ik heb begeleid in de afgelopen jaren. Ik wil Ronald, Sipke, Thomas en Nienke bedanken voor de productieve en ook gezellige gesprekken, zowel online als offline.

Naturally, I would also like to thank my colleagues for the pleasurable years I have spent at the department. We always had a good laugh during lunch and coffee breaks. You are too many to list, but a few I would like to call out to in particular. Daan en Pim, jullie zijn waardige tegenstanders tijdens onze spelletjesmiddagen, waarvan ik hoop dat we ze regelmatig zullen blijven organiseren. Laura en Michelle, jullie kantoor was altijd een heerlijke plek om de dag te beginnen (of juist te onderbreken) met een kop thee. Jullie hebben me altijd van goed advies voorzien, en ik heb genoten van het schilderen tijdens de MI gets crafty woensdagmiddagen. En tenslotte natuurlijk Rosa, mijn steun en toeverlaat in de behoorlijk roerige afgelopen jaren. Om alles te beschrijven wat onze vriendschap inhoudt zou ik nog een boekje nodig hebben, maar het belangrijkste is dat ik op de moeilijkste momenten altijd op je terug heb kunnen vallen, waar ik je voor altijd dankbaar voor zal zijn.

Ook mijn vrienden van Het Duivelsei en daarbuiten wil ik bedanken voor alle afleiding en ondersteuning die ze me hebben gegeven in de afgelopen jaren. Avondjes spelletjes spelen, lekker koken en eten, escape rooms doen, picknicken, zwemmen, muziek maken… en af en toe flink klagen over de wereld, wederom met een goede kop thee. Het was tijdens de coronaperiode niet altijd makkelijk om contact te houden, maar we hebben flink ons best gedaan. Ook jullie zijn met te veel om op te noemen, maar twee van jullie wil ik graag uitlichten. Becky, ik heb genoten van onze wandelingen, reizen en uitstapjes, en hoop dat we nog vele leuke plekken gaan ontdekken. En Anna, dankjewel voor het maken van de prachtige omslag van dit proefschrift.

Pap en mam, dankjulliewel voor de oprechte interesse in alles wat ik doe en het meedenken af en toe. Ik heb veel geluk met zo’n warm nest. En tenslotte Niels, dankjewel voor alle liefde waarmee je me het laatste jaar hebt overladen. Ik ben blij dat we het altijd over alles kunnen hebben, zelfs als het in een wiskunde geek-out van mij resulteert.

Curriculum Vitae

Mark van den Bergh was born in The Hague on 14 August 1994. From 2006 to 2012, he attended the Dalton Den Haag, following the bilingual VWO program. He continued his education at Universiteit Leiden, obtaining bachelor degrees in mathematics and in computer science in 2015, both summa cum laude. His bachelor thesis titled *Hanabi, a cooperative game of fireworks*, supervised by Floske Spieksma and Walter Kosters, led to a publication at the BNAIC 2016. In 2017, Mark obtained a summa cum laude master's degree in mathematics in Leiden. During his studies, he was also active in the board and several committees of the student association Het Duivelsei.

After graduating, Mark continued his career as PhD candidate at Universiteit Leiden from 2017 to 2022. His research interests include, but are not limited to game theory, combinatorics and discrete optimization. During this research, Mark worked closely together with some of his bachelor and master students. Furthermore, he delivered scientific talks at several international conferences, as well as popular scientific talks to varying audiences. Also being active in education, he was responsible for one first-year course and one third-year course in the mathematics bachelor program, and was chair of the program committee for two years. Currently, Mark is appointed as junior lecturer at the Vrije Universiteit Amsterdam.