Imperfect information variants of combinatorial games

Bergh, M.J.H. van den

Citation

Bergh, M. J. H. van den. (2023, June 1). *Imperfect information variants of combinatorial games*. Retrieved from https://hdl.handle.net/1887/3619307

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 $\mathbb{G}$.

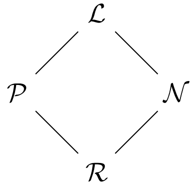


Figure 2.1: The partial order on the outcome classes.

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* $m, n \in \mathbb{N}$*,*

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

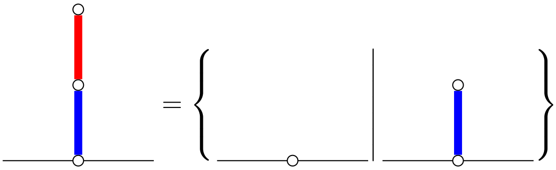
in canonical form.

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* $\min\{G^R\}$*.*

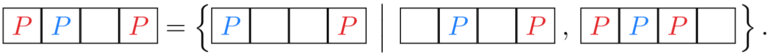
Example 2.1.19. Recall the game of *Red-Blue-Hackenbush* described in Example 1.1.1. By the simplest number theorem, we find, for example,



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

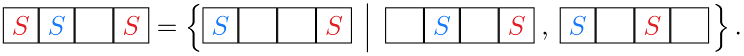
◁

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



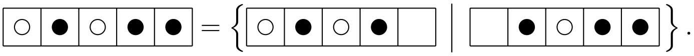
◁

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.

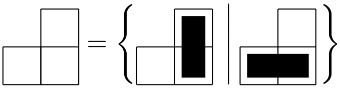
2.1.3 Infinitesimals

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 $x > 0$.

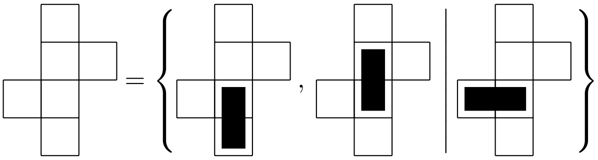
Example 2.1.24. Trivially, the game 0 itself is infinitesimal, being the only infinitesimal number. Somewhat less trivial is the game $* = \{0 \mid 0\}$, which is the smallest example of a next-player win. The game $\uparrow = \{0 \mid *\}$, pronounced *up*, is an example of an infinitesimal win for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

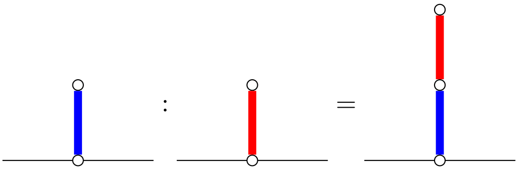
2.1.5 Ordinal sums

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.

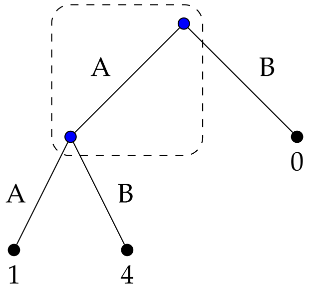


Figure 2.2: The absent-minded driver problem.

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.

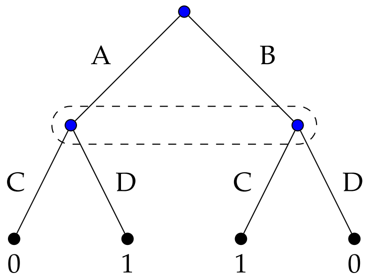


Figure 2.3: Dependency of moves.

◁

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

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

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 $G = \{\ |\ |\ \}$, 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 $1 = \{0\ |\ |\ \}$ and $-1 = \{\ |\ |\ 0\}$, 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 $\mathcal{D}$. $\triangleleft$

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

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

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

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

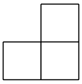
where $(\mathcal{G}^S)^\top$ denotes the transpose of $\mathcal{G}^S$.

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

$G =$



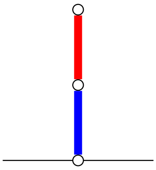
which as a combinatorial game would read $* = \{0 \mid 0\}$. For a synchronized version of this game, it would make sense to define $\mathcal{G}^L = \mathcal{G}^R = (0)$. However, it is unclear what choice would be suitable for $\mathcal{G}^S$, as the players cannot legally execute their moves simultaneously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

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 $0 = \{\,|\,|\,\}$. 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.

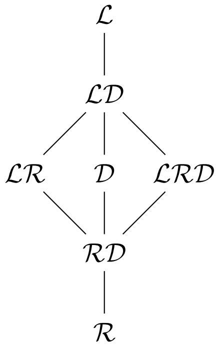


Figure 2.4: The partial order on the synchronized outcome classes.

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*

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

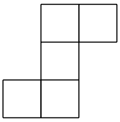
Hence, it seems that, when looking at many copies, some synchronized games behave very much like their combinatorial counterparts. The intuition behind this could be that the probability of the two players playing on the same component will be small, for a large number of components. Hence, the game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

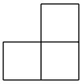
Starting from a ruleset, it is often possible to give a natural interpretation to allowing both moves to be executed anyway, even if this does not lead to a legal position in the underlying combinatorial game. For example, in Domineering, we can allow the placement of two overlapping dominoes if placed 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 $0 \not\cong H + H$ 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. $\triangleleft$

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. ◁

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.