How are Combinatorial Games Solved?

January 2024
Sylvan Kenbeek

Table of contents

Introduction

This paper concerns *game theory*, the mathematical study of games. As the name suggests, a *game* encompasses board and tabletop games. But the reach of game theory is much wider than that, as it also used in economics, computer science, biology, and many more fields. Thus, besides being a beautiful field of study on its own, game theory also proves useful in applied settings, though more on this will come later. Game theory is subdivided into multiple sub-fields. One of those is *combinatorial game theory*, which is the focus of this paper. Among combinatorial games fall, for example, *Nim*, *domineering*, and *Hackenbush*. Besides that, though, combinatorial game theory can also be applied to games such as *chess*, *go*, and *checkers*, even if they themselves are not combinatorial games.

I have chosen to write and research this topic, because, quite simply, it interests me. For a while now, mathematics has been a passion of mine. As such, I have explored a variety of mathematical fields in my own time and learned about others at school. However, game theory was a topic that I knew very little about and had not yet taken the time to research. So, I decided that this paper was a good opportunity for me to learn more about it. Then, looking for a more specific topic that the whole of game theory, I found the sub-field of combinatorial game theory, which especially intrigued me, because it is used to analyse games such as *chess*, which I tend to enjoy playing because of the strategic aspect.

An important aspect of (combinatorial) game theory is finding ways to get the best move for a player given a position. This paper aims to shed light on exactly this aspect of combinatorial games. Hence, the research question: *How are combinatorial games solved?*
The goal is not, however, to give one way to solve any combinatorial game – however nice that would be – as combinatorial games are too varied for such a thing to exist. Instead, the objective is to give multiple methods to determine who will win a game given a set of rules and a starting position and to present multiple strategies and tactics that may be useful in the determining thereof and otherwise for the reader that may want to apply them when playing games in the future.
To get there, we must first start with the basics: *What is Game Theory?*
In this chapter, game theory as a mathematical field of study will be explored and important terminology required for the rest of the paper will be explained.
The second chapter is all about *Combinatorial Games*: how to describe them and how to analyse them.
The third chapter explains various *Strategies* that can be applied to combinatorial games.
Finally, the fourth chapter addresses *Solving Methods*. First though, will be given a definition of what a *solved game* even is and what *solving a game* means. And then a variety of methods that can and have been used to solve combinatorial games are presented.

This paper is a literary study of combinatorial game theory. A variety of books and papers on the topic have been collected and were used to gather the information and knowledge required to write this text.

The reading of this paper requires minimal mathematical knowledge, – a high school level understanding should be plenty, – though a good intuition or reasoning when it comes to mathematics is certainly useful, but again, likely not required.

Two indices and an appendix are included at the end of the paper. *Index A* lists the important terms in alphabetical order and also notes on what page(s) they are explained. *Index B* lists (non-trivial) symbols, sectioned by type and then by appearance. For each symbol, as short description as well as the page number where it is first used/explained are noted. As this paper is about game theory, some games are used as examples for certain topics. *Appendix: Rulesets* is a collection of rulesets for all games that are used[1] in the paper. Keep in mind that most, if not all, of the games in the appendix have variants. Unless such a variant is used in the paper, and is not explained when it is used, it is unlikely that the rules of the variant can be found in the appendix.

[1] Games that are merely mentioned are not included in the appendix.

1. What is Game Theory?

Game theory is the mathematical study of interactive situations, called *games*. That is to say, situations involving multiple parties called *players*, that compete or cooperate in a way that affects one another. (Bonanno, 2018; Osborne, 2000; Peters, 2015)

As the name suggests, games such as *chess*, *tic tac toe*, and *poker*, belong in this field, but the study of games has much wider applications. So is game theory highly applicable in economics (see, for example, Samuelson, 2016), but also finds use in many other fields, such as politics (see, for example, Brams, 2003), computer science (see, for example, Apt et al., 2011), biology (see, for example, McNamara, Leimar, 2020), and linguistics (see, for example, Jaeger, 2008).

What could be considered a game is so broad, even, that game theoretic situations can be recognized in works as early as the *Bible*[2], the *Talmud*[3], and *The Art of War* by Sun Tzu, over 2000 years ago. Among the first formal works on game theory, however, were Zermelo on Zermelo's theorem (Zermelo, 1913) and Von Neumann on the Minimax theorem for zero-sum games (Von Neumann, 1928), the latter work being the basis for the book *Theory of Games and Economic Behavior* (Von Neumann, Morgenstern, 1944), which is considered the breakthrough or start of game theory as a mathematical discipline. In the following decades, game theory was mostly developed in the domain of mathematics, by mathematicians counted among whom are Nash, Shapley, and Bondareva. By then game theory had also become a major influence on economic branches and later also many other fields. In 1994, mathematicians Nash, Harsanyi, and Selten won the Nobel Prize in Economic Sciences for their work in game theory. This same prize has since been awarded for achievements in or related to game theory a number more times: in 1996 to Mirrlees and Vickrey; in 1998 to Sen; in 2001 to Akerlof, Spence, and Stiglitz; in 2005 to Aumann and Schelling; in 2007 to Hurwicz, Maskin, and Myerson; and most recently in 2012 to Roth and Shapley. (Bonanno, 2018; Osborne, 2000; Peters, 2015)

1.1. A dip into Set Theory

A *set* is a collection of items called *elements*. These elements can be anything from vehicles, to animals, to fruits, and even 'nothingness'. In the study of sets, though, sets usually contain numbers or other sets.

The notation for a set is as follows:

$$A = \{element\ 1, element\ 2, element\ 3, \dots, element\ n\}$$

This means that A is a set that contains elements 1 through n (everything that is within the curly brackets).

In a set, the order of elements does not matter, so

$$\{1,2,3\} = \{3,1,2\},$$

and neither does how often an element appears matter, meaning that

$$\{1,1,1,1,1,1,1,1,1,2,2,2,2\} = \{1,2\}.$$

This notation for sets is useful when you don't have too many elements, but some sets contain a very large amounts or even an infinite number of elements. For such cases, the following notation might be easier to work with:

$$B = \{x \mid y\}$$

Here, B is the set of all elements x, for which condition y is true. Left of the vertical bar, one might also denote what kind of element x is.

For example:

$$C = \{integers\ x \mid x\ is\ even\}$$ [4]

is read as "*C is the set of all integers x, such that x is even*." "*integers x*" tells us that we are only looking at integer numbers and the condition that must be met is that x is even. So, if an integer x is even, then it is an element of the set C.

Other examples of sets with this notation are:

$P = \{numbers\ p \mid p\ is\ prime\}$, is the set of all prime numbers,

$D = \{sets\ X \mid X\ contains\ 3\ elements\}$, is the set of all sets that contain three elements, and

[2] 1 Kings 3:26

[3] Ketubod 93

[4] Integers, also "whole numbers", are all numbers without a decimal part. E.g. -314, 0, and 271.

$E = \{numbers\ x \mid x\ is\ divisible\ by\ 1931\}$, is the set of all integers that are divisible by 1931.
This can also be done with more mathematically rigorous notation, which looks thusly:

$E = \{x \in Z \mid \frac{x}{1931} \in Z\}$,

read as: E is the set of all integers x, such that x divided by 1931 is an integer. This is the same as the set of all integers that are divisible by 1931.

Two important symbols in set theory are the symbol for set membership operation (∈) and the symbol for the union of sets (∪).

The set membership operation is used in the following ways:

$x \in \{1,2,3\}$ or $y \in A$.

The first example means that x is an element in the set of the numbers 1, 2, and 3 (so it is one of them) and the second example means that y is an element of the set A. On the other hand,

$z \notin B$

means that z is not an element of the set B.
For example,

if $F = \{-4, \frac{3}{8}, 2, 18\}$, then $2 \in F$, but $-11 \notin F$.

The union of two sets is the set of all elements that are in either (or both) of the sets. The notation is as follows:

$X = A \cup B$, where X, A, and B are sets.

As an example, if $A = \{0,2,4,6,8\}$ and $B = \{1,4,9,16,25\}$, then $A \cup B = \{0,1,2,4,6,8,9,16,25\}$.

Though not exclusive to set theory, another symbol that will be of use is :=, which is used like

$x := E$,

and reads as "x is defined to be E" or "let x equal E".

1.2. Games

In the study of games, there are a variety of attributes that may or may not apply to a game. Likewise, there exist many game types. What follow are some important terms and types of games.

1.2.1. Terminology of Games

Competitive games

A game is (strictly) *competitive* when the game's players' interests are opposed to each other (Osborne, 2000). Games in which one player's win means a loss for all other players are competitive games.

Cooperative games

In a *cooperative game* players can communicate with each other to form coalitions and make binding agreements. Real-life situations such as international alliances and treaties, and voting behaviour have been studied as cooperative games. (Osborne, 2000; Peters, 2015)

Evolutionary games

In an *evolutionary game*, players are representatives from an evolving population (e.g. animals). The behaviour of an individual is based on the behaviour of their parent(s), possibly with a slight mutation. The study of evolutionary games first came about in biology, but has since also been applied to the study of human behaviour and other topics. (Osborne, 2000; Peters, 2015)

Bargaining games

In a *bargaining game*, (usually) two players have to make an agreement on choosing one from a selection of options. The study of bargaining games has two main sub-fields, the first being the cooperative approach, which tries to maximize the total gain for all players, and the second is the competitive approach, which tries to maximize the gain for a single player. (Peters, 2015)

One-shot

In a *one-shot game*, players make only a single, simultaneous move (Peters, 2015). Flipping a coin and betting on the result is an example of a one-shot game.

Repeated games

In a *repeated game*, players play multiple one-shot games against each other. During any individual game of such a repeated game, the players have the knowledge of any previous games, and will either expect or know that they will play more games after the current one. (Ross, 2023)

Extensive form

In a game in *extensive form*, there are finitely many players (usually two), who make moves sequentially instead of simultaneously. A game in extensive from can (and usually does) have players make more than one move over the course of the game. (Bonanno, 2018; Peters, 2015)

Finite

A game is *finite* when the game will always end after a finite number of moves, each player has a finite number of options, and there are a finite number of players (Albert et al, 2007; Peters, 2015). Finite games include *Scrabble*, *connect four*, and *chess*, as their rules always force the games to have a finite number of moves.

Perfect information and imperfect information

A game has *perfect information* if and only if all players have access to all information relevant to the game. This includes the rules of the game, what possible moves either player could make, what the result of those moves would be, and any preceding moves that may have happened. (Bonanno, 2018)
Logically follows that a game with *imperfect information* has some information that players do not have access to.
Any game involving chance or randomness has imperfect information, since neither player can know the outcome of a random decision. Any game where players make moves at the same time also has imperfect information, because players cannot know for certain beforehand what move the other player(s) will make. In games such as *Nim* or *tic tac toe*, however, all players can know at all times every move that has been made, as well as the outcome of every move that can be made in the future. And thus, these games do have perfect information.

Payoff

The *payoff* of a game, for a player, is the value that player gains (or loses) after having played the game. The payoff for each player can be different.

Strategies

A *strategy* is a planned sequence of moves to play a full game (Peters, 2015). In a one-shot game, a strategy consists of only one move, but in a game in extensive form or in a repeated game, a strategy consists of many moves, which may depend on the moves made by other players.

Options

Given a position of a game, if player 1 may make a move, then she will have some number of choices she can make. All possible positions that can come fourth from these choices are the *player 1 options*. If player 2 may make a move, then he will similarly have some number of choices he can make, and all positions that can come therefrom are *player 2 options*. The same holds true for any other players. The *options* are the union of the options of all individual players. (Albert et al, 2007)

1.2.2. Types of Games

Zero-sum Games

Zero-sum games usually refer to *finite two-person zero-sum games*.
A finite two-person zero-sum game is a one-shot game in which both players have some number of options for their move. The chosen options together determine the payoffs for the players. A zero-sum game has the property that the payoff of player 1 is the opposite of the payoff of player 2, making the sum of the payoffs 0. The options and the corresponding payoffs can be summarized in a matrix[5]. Hence, these games are also called *matrix games* (Peters, 2015)

In the game *matching pennies*, two players each have a coin which they simultaneously throw and then reveal. If both coins show the same side (heads and heads, or tails and tails), then player 1 gets player 2's coin, for a payoff of 1 (coin). If the two coins show different sides (heads and tails, or tails and heads), then player 1 gives their coin to player 2, for a payoff of -1.
The payoff matrix for matching pennies look as follows:

$$\begin{array}{c} \\ \text{Heads} \\ \text{Tails} \end{array} \begin{array}{c} \begin{array}{cc} \text{Heads} & \text{Tails} \end{array} \\ \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \end{array}$$

where player 1 "chooses" a row *i*, player 2 "chooses" a column *j*, the entry of row *i* and column *j* is the payoff for player 1 (which is the number of coins gained), and the payoff for player 2 is $-(payoff\ player\ 1)$.

Nonzero-sum Games

Nonzero-sum games usually refer to *finite two-person games*.
A nonzero-sum game is the same as a zero-sum game, except that the sum of the players' payoffs does not have to equal zero. This means that a finite two-person game needs two payoff matrices: one for player 1 and one for player 2, causing such games to also be called *bimatrix games*. Instead of writing two separate matrices, however, usually these matrices are written as one matrix with two numbers in each entry. (Peters, 2015)

In *the prisoner's dilemma*, two criminals committed a crime together and have been arrested. They are interrogated separately, without any way to communicate with one another. The police admit they have no actual proof the criminals committed the crime, but are planning to sentence both prisoners for 1 year on a lesser charge. However, the police also tell both prisoners that if one of them confesses (C) and the other prisoner stays silent (S), then the prisoner that confessed will be set free, while the prisoner that stayed silent has to serve 5 years in prison. If both prisoners confess, however, they will each serve three years.
This can be summarized in the following payoff matrix:

$$\begin{array}{c} \\ \text{S} \\ \text{C} \end{array} \begin{array}{c} \begin{array}{cc} \text{S} & \text{C} \end{array} \\ \begin{pmatrix} 1,1 & 5,0 \\ 0,5 & 3,3 \end{pmatrix} \end{array}$$

Prisoner 1 (player 1) chooses a row *i*, prisoner 2 (player 2) chooses a column *j*, the left number in the entry of row *i* and column *j* is the payoff for prisoner 1 (the number of years she will serve in prison), and the right number of the same entry is the payoff for prisoner 2.

Since nonzero-sum games do not disallow for the sum of the players' payoffs to be zero, each zero-sum game is a special case of a nonzero-sum game.
The zero-sum game *matching pennies*, can also be summarized as a nonzero-sum game with the following payoff matrix:

[5] Since the payoff of player 2 is the opposite of the payoff of player 1, only the payoff of player 1 is needed.

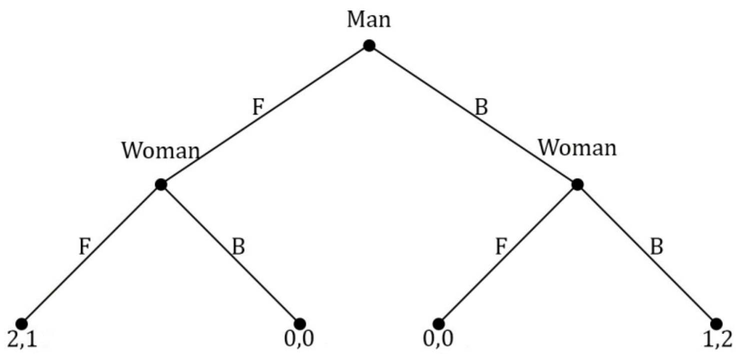
$$\begin{array}{ccc} & \text{Heads} & \text{Tails} \\ \begin{array}{c}\text{Heads}\\ \text{Tails}\end{array} & \multicolumn{2}{c}{\begin{pmatrix} 1,-1 & -1,1 \\ -1,1 & 1,-1 \end{pmatrix}} \end{array}$$

with the left number of each entry being the payoff for player 1 and the right number being the payoff for player 2.

Finite Extensive Form Games

Finite extensive form games have players move sequentially, and they may each make multiple moves if the game allows it. Additionally, there may only be a finite number of players, total moves in the game, and options per move. Finite extensive form games are best described by a game tree, which shows whose move it is and what a player's information is have when that player must make a move. (Peters, 2015)

In the *battle of the sexes* a man and a woman want to go out together. They had decided to go to either a football match (F) or a ballet performance (B) but forgot to agree which they would go to. They have no way to communicate with each other, but must decide which to go to. The man would like to go to the football match and the woman wants to see the ballet performance, but they both prioritize being together.
This is a nonzero-sum game, but can be made into a finite extensive form game (called the *sequential battle of the sexes*), if we assume the man decides first and the woman chooses after that (or vice versa). This results in the following game tree:



From this game tree can be seen that, if the man chooses to go to the football match, the woman can choose to do the same, resulting in a payoff of 2 (which could represent the amount of enjoyment) for the man and a payoff of 1 for the woman, if the man chooses to go to the ballet performance and the woman does the same, then the man gets a payoff of 1 and the woman a payoff of 2, and if the woman chooses the option the man did not choose, then they will both get a payoff of 0.

Combinatorial Games

Combinatorial games are very similar to finite extensive form games. But their major difference is that a combinatorial game must have perfect information, where a finite extensive form game does not (but can).

The sequential battle of the sexes as described previously could be made a combinatorial game if the woman got to know what the man's choice was before deciding for herself where to go to.

This paper's analysis on how to solve games using game theory focusses specifically on combinatorial games, so a more extensive and formal definition will follow in the next chapter.

1.2.3. The Players

When analysing a (combinatorial) game with two players, the players are usually called *Left* (or L) and *Right* (or R). There are, however, also other names that are given to the players:

Left	**Right**
Louise	Richard
Positive	Negative
Black	White
Blue	Red
Vertical	Horizontal
Female	Male

(Albert et al, 2007)

In this paper we will refer to the players as Left and Right and with female and male pronouns respectively, unless a game already has names for the players, such as White and Black in chess, and X and O in tic-tac-toe, in which case those may be used instead.

There are also games that have a neutral colour. Standard neutral colours are green (in a game between blue and red) and grey (in a game between black and white). (Albert et al, 2007)

When colours are needed to visualize a game, this paper will use a dark grey for Left, a light grey for Right, and no colour (white/the background) as the neutral colour.

2. Combinatorial Games

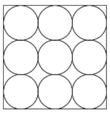
A *combinatorial game* has two players who sequentially make moves. The game continues until the current player has no legal moves they can make. Under *normal play*, the last player to make a move wins. Under *misère play*, the last player to make a move loses. A combinatorial always has perfect information. This paper will mostly focus on the more common finite combinatorial games, though being finite is not a requirement for combinatorial games. (Albert et al, 2007)
Continuing, a game will be assumed to be under normal play, unless stated otherwise.
Examples of combinatorial games are *domineering*, *hex*, and *Nim*.
There also exist games that, though they fall outside the definition of a combinatorial game, can still be studied with combinatorial game theory. Examples are *dots and boxes*, where a player may sometimes make two moves in a row, *tic-tac-toe*, which contains draws (meaning there is not always a winner), and *go*, since the winner is based on the player that has the most pieces regardless of who played last. However, some sources do call games like these combinatorial games as well.

2.1. Positions and Game Trees

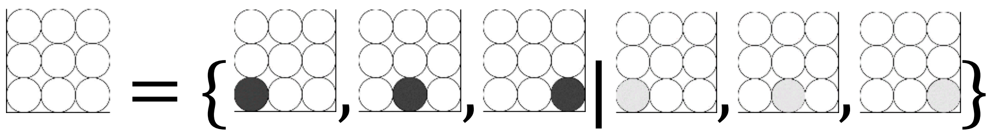
2.1.1. Positions

A game or a position of a game is defined by its options: $G = \{\mathcal{G}^L|\mathcal{G}^R\}$, with $\mathcal{G}^L$ and $\mathcal{G}^R$ being the sets of the options of Left and Right respectively. (Albert et al, 2007)

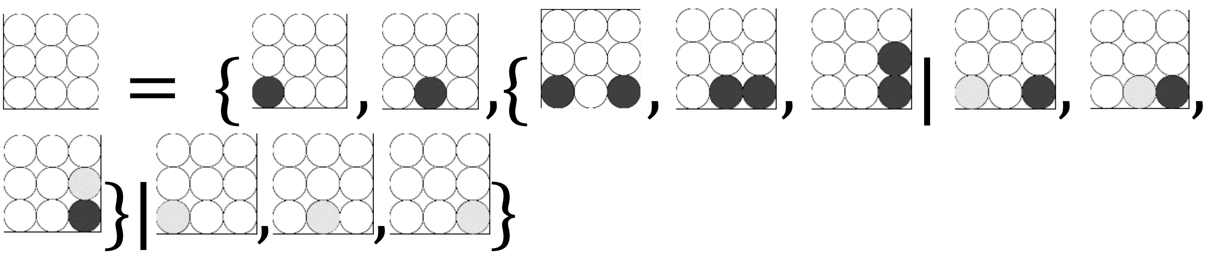
Consider a game of *connect three*[6] on a 3 by 3 grid:



The game would be defined as the following set:[7]



Note that each of these options each also have some options that define them. The game could therefore also be defined with the options of some or all of the options, for example:



When describing a game by its options or game tree, – which the next section will introduce, – two or more consecutive moves for one player are often shown (in the case of this example, two consecutive Left moves). Though it may seem wrong to do this, as a player may never make two moves in a row in a combinatorial game, doing it this way is useful when a game can be decomposed into several subgames and lets us analyse the options of a game as games of their own, which helps with the analysis of the full game. (Albert et al, 2007)

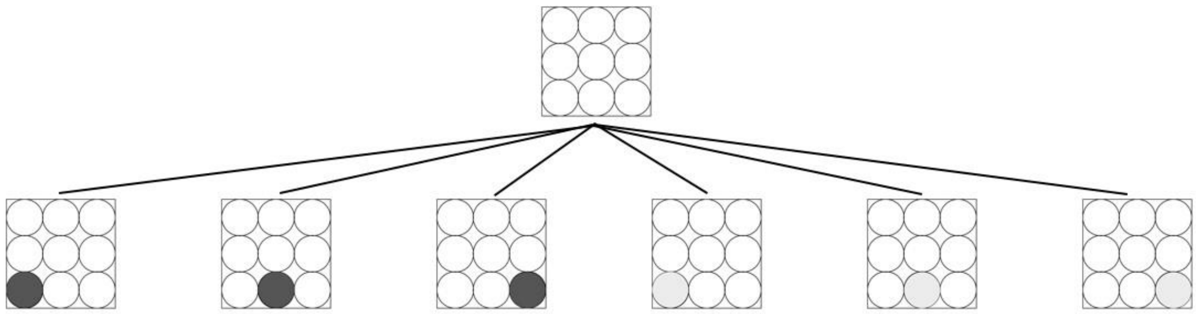
[6] *Connect three* is not a combinatorial game but works for the example.
[7] Even though the Left options and Right options are sets, the curly brackets are usually omitted.

2.1.2. Game Trees

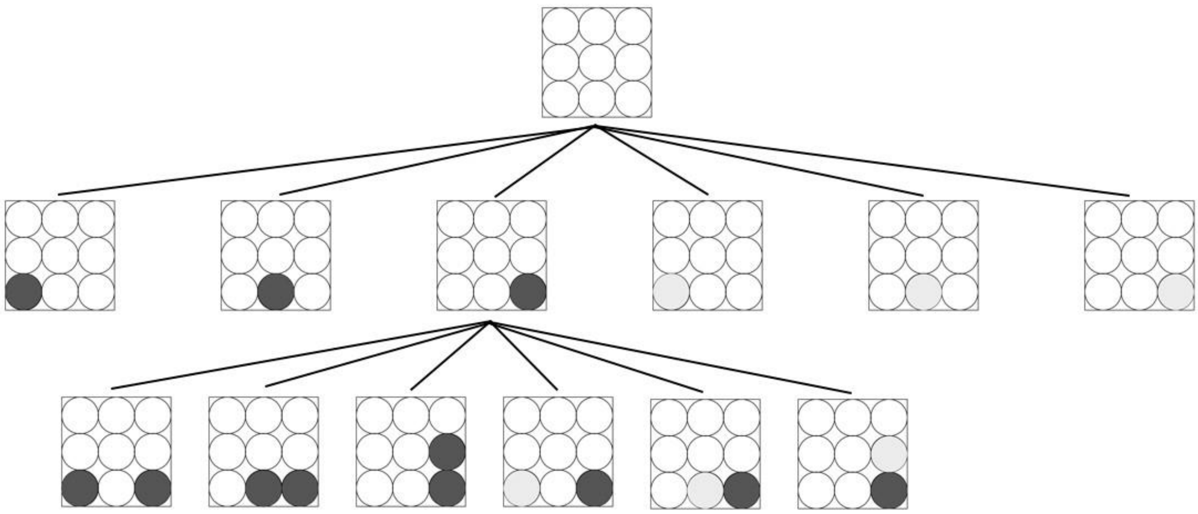
A game can also be described by a *game tree*. A game tree is defined by its *nodes* and *edges*. Nodes are points where a player must make a decision and edges are actions of a player. The upper node or *root* of the tree is the start of the game (or position), where the first player chooses an option (Peters, 2015). The bottom nodes or *leaves* are the end positions of a game, where neither player can make a move (Albert et al, 2007).

The edges pointing down and to the left from a node together form the Left options and the edges pointing down and to the right form the Right options. A game tree may show the options of a position by listing only the options themselves, or also listing options of (some of) those options, or by showing the full tree, like with the definition of a game. (Albert et al, 2007)

A game tree of the 3 by 3 *connect three* game may then look like:



Or like:



2.2. Outcome Classes

2.2.1. Fundamental Theorem of Combinatoric Games

"Fix a game G played between Albert and Bertha, with Albert moving first. Either Albert can force a win moving first, or Bertha can force a win moving second, but not both." (Albert et al, 2007)

This theorem holds, because any end position is a win for either Albert or Bertha, and a loss for the other player. This means that in a position before one of these, the current player can win if an end position that is a win for them can be reached. If no such position exists, then the previous player wins. Going one move back, the then current player can force a win if a move of theirs forces the next player to make a losing move, and if no such move exists, then the previous player can win. Extrapolating backwards, the same is true for all positions of the game, meaning that, from the initial position, it must be true that Albert can force a win moving first, or Bertha can force a win moving second, but not both.

Given the fundamental theorem of combinatoric games, *perfect play* of a player means that that player employs a *winning strategy*, which is a set of moves that will guarantee the player a win, if they have one. If a player has no winning strategy, playing perfectly means making any move. (Albert et al, 2007).

2.2.2. Outcome Classes

Using the fundamental theorem of combinatoric games, there are four possibilities that arise when a game G is fixed, with either Left as the first to move or Right as the first to move. These are the *outcome classes*:

Class	Name	Definition
$\mathcal{N}$	fuzzy	The next player to play (regardless of if it is Left or Right) can force a win
$\mathcal{P}$	zero	The previous player to play (regardless of if it is Left or Right) can force a win
$\mathcal{L}$	positive	Left can force a win (regardless of whether Left or Right moves first)
$\mathcal{R}$	negative	Right can force a win (regardless of whether Left or Right moves first)

Another representation of the outcome classes is:

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

(Albert et al, 2007)

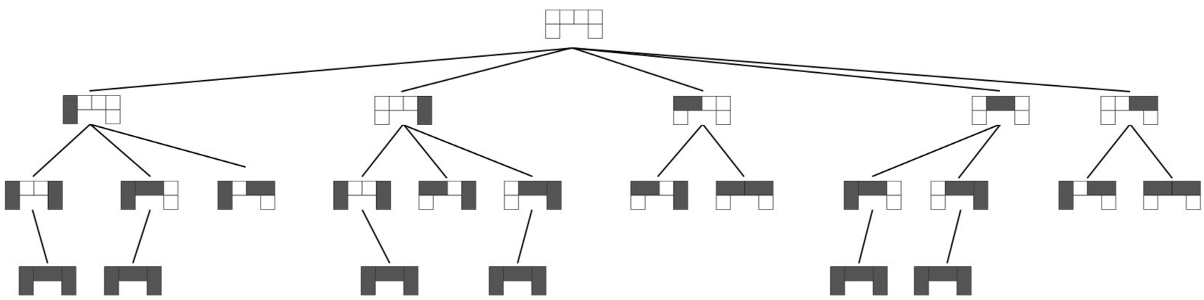
Given the outcome class of its options, the outcome class of a position or game can be determined as is shown in the following table, where G^L is a left option and G^R is a right option:

Outcome Classes	Some $G^R \in \mathcal{R} \cup \mathcal{P}$	All $G^R \in \mathcal{L} \cup \mathcal{N}$
Some $G^L \in \mathcal{L} \cup \mathcal{P}$	$\mathcal{N}$	$\mathcal{L}$
All $G^L \in \mathcal{R} \cup \mathcal{N}$	$\mathcal{R}$	$\mathcal{P}$

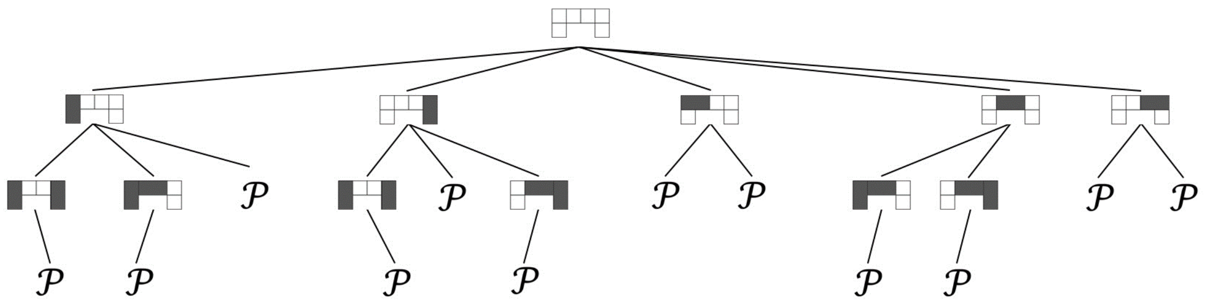
(Albert et al, 2007)

Consider that, in normal play, the last player to make a move wins. For the resulting position $G \in \mathcal{P}$ must then be true. In a game tree, these positions are the leaves of the tree. With this knowledge, it is possible to recursively determine the outcome class of every position of a game.

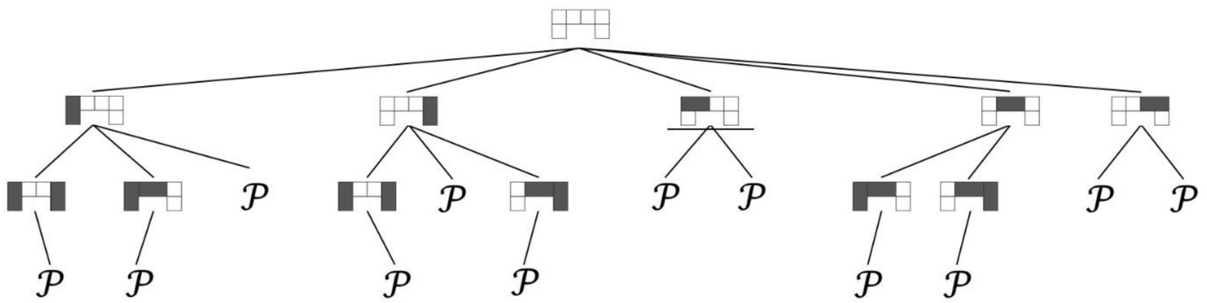
For example, take the following game of *domineering* and its game tree:



Replacing all leaves with the zero outcome class gives:

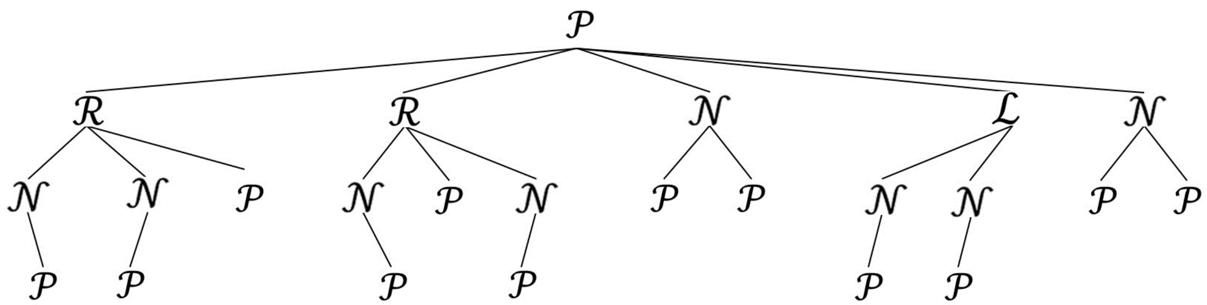


And with this, the rest of the tree's outcome classes can be determined. This is done by looking at the Left and Right options of a position, and then looking at the third table for outcome classes. Since the leaves are all the zero outcome class, the positions leading to only leaves can have their outcome class found. For example, take this position (underlined):



Looking at the position's Left option(s), we can see that it falls under "Some $G^L \in \mathcal{L} \cup \mathcal{P}$" (because (at least) one Left option has the $\mathcal{L}$ or $\mathcal{P}$ outcome class), from the table. Likewise, we can tell that the Right option(s) fall under "Some $G^R \in \mathcal{R} \cup \mathcal{P}$" (because (at least) one Right option has the $\mathcal{R}$ or $\mathcal{P}$ outcome class). Taking that row and that column, we find that the position's outcome class is $\mathcal{N}$, or fuzzy.

Then doing the same for all other positions, working backwards, the outcome class of the game itself can be ascertained:



This shows that the game of *domineering* can be won by whomever plays second, regardless of whether that is Left or Right.

2.2.3. Impartial and Partizan Games

In some games, such as in *Nim*, it makes no difference whether Left moves first or Right; the Left options and Right options are the same: $\mathcal{G}^L = \mathcal{G}^R$. Such games are called *impartial*. For other games, such as *hex*, it does matter which player moves first, as the Left options and Right options are not the same: $\mathcal{G}^L \neq \mathcal{G}^R$. Those games are called *partizan*. (Albert et al, 2007)

This means that an impartial game must have the zero outcome class or the fuzzy outcome class, while a partizan game can have all four outcome classes.

Though what follows in this paper can be applied to impartial games, it sees more use in analysis of partizan games. Because impartial games can only have the zero of fuzzy outcome classes, there is a different way that those games can be analysed that does not work for partizan games.

2.3. Games and their Values

2.3.1. More on Outcome Classes

To further be able to describe games, it is useful to be able to assign each game a value.

The simplest possible game is $G = \{|\}$, where neither Left nor Right have any options. It is trivial to see that whoever starts the game loses, as the first player to move has no legal moves. This came is called *zero* and is written as $0 \coloneqq \{|\}$. With this game defined, other games can be constructed, such as $G = \{0|\ \}$. This game is a win for Left, as, if she moves first, she chooses to play 0, after which Right has no legal moves, and if Right moves first, he has no legal moves and loses. Similarly, $G = \{\ |0\}$ is a win for Right. Another game is $* \coloneqq \{0|0\}$ which was given its own symbol, and is named *star*, which is a win for the first player, regardless of who that is. (Blom, 2021)

With these four games, the outcome classes can be described in a new way:

If $G = 0$, the second (or previous) player to move wins. This is outcome class $\mathcal{P}$.
If $G > 0$, Left can force a win. This is outcome class $\mathcal{L}$.
If $G < 0$, Right can force a win. This is outcome class $\mathcal{R}$. And
if $G \parallel 0$, the first (or next) player to move wins. This is outcome class $\mathcal{N}$. (e.g. $G = *$)

(Blom, 2021)

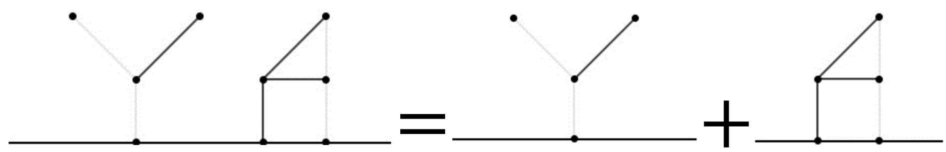
2.3.2. Game Sums

Many games can be divided into independent components, which can each be viewed as their own game. Since these components are independent, the full game can be described as a sum of its components. (Albert et al, 2007)

This leads to the definition of *game sums*:
"The direct sum of two games G and H is written G+H, given by $G + H \coloneqq \{\mathcal{G}^L + H, G + \mathcal{H}^L | \mathcal{G}^R + H, G + \mathcal{H}^R\}$*."*[8]
(Blom, 2021)

In a game sum of positions, the player to move can move in any of the summands and can treat that as an individual game, during play a new (sub-)position may be able to be written as another game sum, and the last player to move in the final summand still wins. (Albert et al, 2007)

Take the following game of *Red-Blue Hackenbush* and its game sum as an example:



A game can also have a negative: $-G \coloneqq \{-\mathcal{G}^R | -\mathcal{G}^L\}$, and corresponds to swapping the players. The sum of a game and its negative is always zero, $G + (-G) = 0$. This is not too difficult to see, as, if the first player plays to a position H in G, then the second player can move to position –H in –G. If the first player moves in –G instead, the second player just moves in G. Player 2 can keep this up until player 1 has no moves left, thus winning. One more property of the negative of a game, is that the negative of the negative of a game is

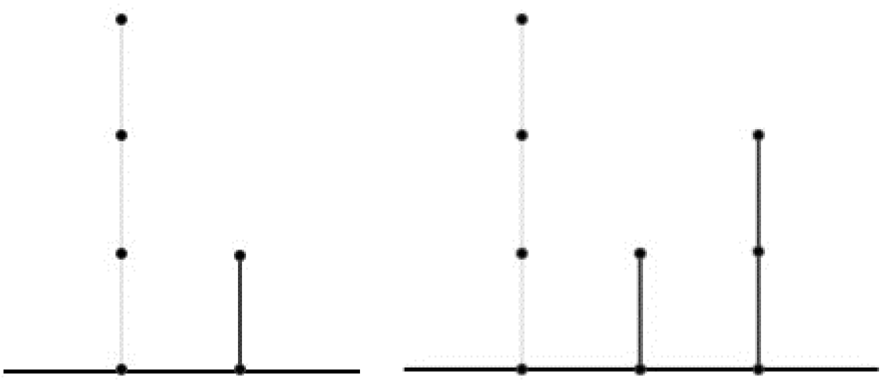
[8] Where $\mathcal{H}^L$ and $\mathcal{H}^R$ are the Left and Right options respectively of game H.

the same as the original game: $-(-G) = G$, as that corresponds to swapping the players twice, which puts them back to their original places. (Blom, 2021)

2.3.3. The Value of a Game

The value of any zero game is 0 and the value for any fuzzy game is the same as that of *. For impartial games, this means that for every game G, $G = 0 \lor G = *$. Partizan games, however, can take on many more values.

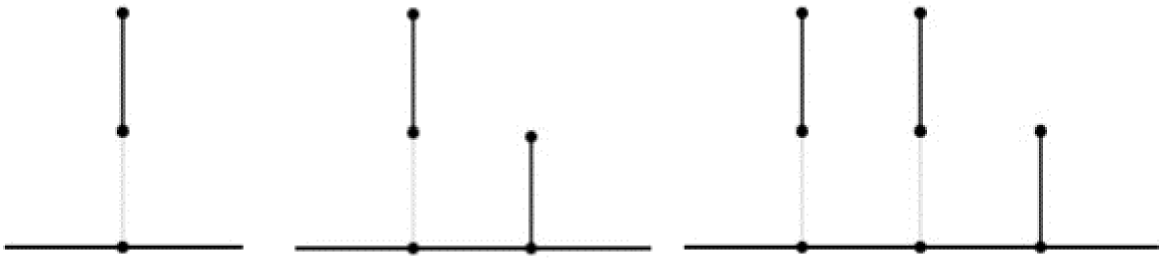
Consider the following games of *Red-Blue Hackenbush*:



G_1 G_2

In G_1, Right (who plays Red, shown as light grey) can force a win. This means that $G_1 < 0$. But how much less is it exactly? Well, in G_2 two lines for Left (who plays Blue, shown as dark grey) have been added to G_1, which has turned it into a zero game. So, adding 2 (lines) to 1 (line) for Left, cancels out Right's 3 (lines). And since the value of a game is $G > 0$ if Left can force a win and is $G < 0$ if Right can force a win, a line for Left has a value of 1 and a line for Right has a value of –1. Now the value of G_2 can be written as $G_2 = -3 + 3 = 0$, and the value of G_1 can be computed also: $G_1 = -3 + 1 = -2$.

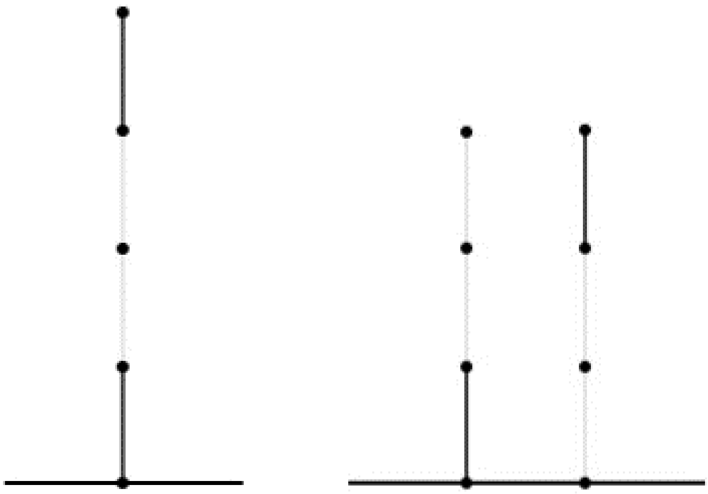
But now take a look at the following games:



G_3 G_4 G_5

Again, G_3 is a win for Right. However, this game does not fit in with the system of values as described with the first set of examples. G_4 shows that $G_3 > -1$, as adding 1 makes it a win for Left, but since G_3 *is* a win for Right, $-1 < G_3 < 0$, and thus G_3 must (and by association games in general can) have a non-integer value. To find the value of G_3, we look at G_5, which has two copies of it placed side by side and with a blue line next to them. For G_5 can be determined that it is a zero game, meaning $G_5 = 0$. So $2 \cdot G_3 + 1 = 0$, which leads to the conclusion that $G_3 = -\frac{1}{2}$. (Bartlett, 2006)

G_6 and G_7 below are a few more fractional games.



G6 G7

G_6 has value $\frac{3}{8}$, which can be found by adding seven more copies (making 8 total) of G_6 as well as 3 lines for Right, which makes a zero game. This gives the following equation to solve: $8G_6 - 3 = 0$, leading to $G_6 = \frac{3}{8}$. The value of G_7 can be found to equal $-\frac{5}{4}$ in a similar way, and is left as an exercise for the reader to try if they want. A useful note to add for G_7 is that it is composed of two games, of which the values can be found separately, and then summed.

2.4. Viewing Games as Numbers

2.4.1. Dyadic Rational Numbers

It has been shown that games can take on integer values and rational values, but it is good to get a better grasp of this in a more general situation.

A game where Left has *n* free moves[9] available has a value of *n*, and a game where Right has *n* free moves has a value of –*n*. More formally, for any $n \in Z^+$, the games are defined as:

$n \coloneqq \{n - 1|\ \}$

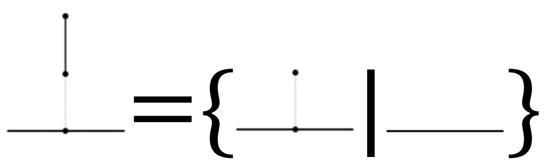
and for any game –*n* as the negative of *n*:

$-n = \{\ |1 - n\}$.

(Albert et al, 2007)

With the integers defined, we can continue to the *dyadic rational numbers*. A dyadic rational number (or binary rational number), is any number that can be written as a fraction whose denominator is a power of 2. In *short games*, that is to say, games with a finite game tree and wherein a position may not be repeated, the value of a game will always be a dyadic rational number. (Albert et al, 2007)

An example: we know that G_3 from the previous section has a value of $\frac{1}{2}$. Its options are



which is equal to $\{-1|0\}$. And since $G_3 = -\frac{1}{2}$, $\{-1|0\} = -\frac{1}{2}$ must be true.

[9] "Free moves" meaning the moves left after subtracting the amount of moves the other player has.

More generally, for any odd m and $j > 0$, numbers are defined as:

$\frac{m}{2^j} = \{\frac{m-1}{2^j} \mid \frac{m+1}{2^j}\}$. (Albert et al, 2007)

2.4.2. Surreal Numbers

When expanding scope beyond short games, a game can take on any value that is a *surreal number*, which includes the integers, the reals (so also all rationals), the ordinals, and more. (Albert et al, 2007)

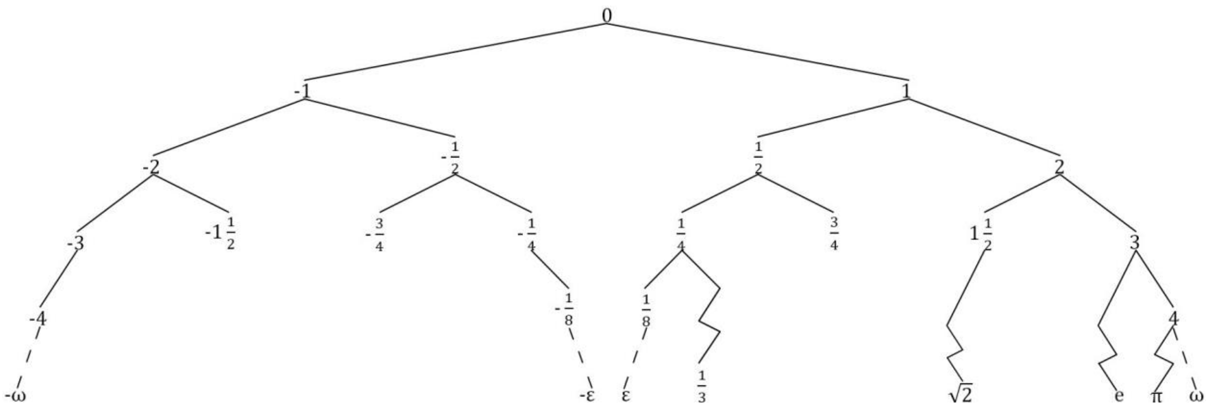
"A surreal number $x = \{x^L|x^R\}$ is a game whose options are all surreal numbers, with $x^L < x^R$ for all x^L and x^R." (Blom, 2021)

All previously constructed numbers fit into this definition, though now many more numbers can be made.

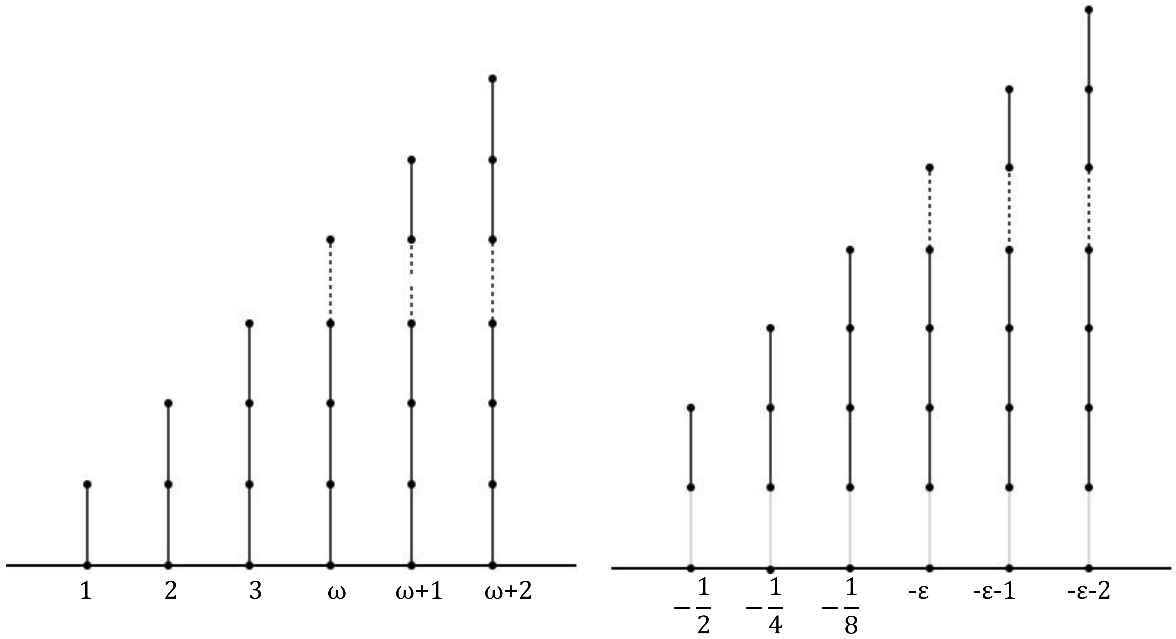
It is time to construct, or at least show how to construct, every surreal number, from the ground up.
On the zeroth layer of the tree, also called the zeroth day, the game $0 = \{|\}$ is found. On the first day the games $\{0|\}$, $\{|0\}$, and $* = \{0|0\}$ are discovered. The first two are (named) 1 and –1 respectively, but, as $0 \not< 0$, * is not a number. On day two, we get the following numbers $2 = \{1|\} = \{1,0|\} = \{1,-1|\} = \{1,0,-1|\}$, $\frac{1}{2} = \{0|1\} = \{-1,0|1\}$, $0 = \{|1\} = \{-1|1\} = \{-1|\}$, $-\frac{1}{2} = \{-1|0\} = \{-1|0,1\}$, and $-2 = \{|-1\} = \{|0,-1\} = \{|1,-1\} = \{|1,0,-1\}$. (Blom, 2021)
In {1,0,-1| }, Left would always choose 1 over 0 and –1, as 1 is a win for Left, whereas –1 is a win for right and 0 is a "less decisive" win for Left. Similarly, in { |1,0,-1} Right would always choose -1 over 0 and 1. In general, all numbers but the largest for Left and all numbers but the smallest for Right may be omitted while keeping the value the same. This simplest form of a game is its *canonical form*. (Albert et al, 2007)

Initially, only the dyadic rational numbers can be made this way. However, in infinite games, and thus games with infinitely many positions, all real numbers can be constructed, as well as a few "new" numbers:
$\omega := \{0,1,2,3,\dots|\}$, which is a surreal number larger than any real number, and $\varepsilon := \frac{1}{\omega} = \{0|1,\frac{1}{2},\frac{1}{4},\dots\}$, which is a surreal number closer to zero than any positive real number (while not being equal to 0), but also $\omega + 1$, $2 \cdot \omega$, ω^2, and more. (Blom, 2021)
One may note that ω and ε seem not to be in their canonical form, but they are. Because they contain infinitely many numbers, there is no "largest" number in ω's Left options and no "smallest" number in ε's Right options, and thus it is not possible to remove "all but the largest/smallest" numbers. The same is true for all surreal numbers that are not a dyadic rational number.

Shown below is a (partial) tree of the surreal numbers, including the first few layers fully, positive and negative ω and ε, as well as a few other numbers.



And to illustrate how ω and ε would look as games, here are games with those values in *Red-Blue Hackenbush*:



2.4.3. Comparing Games

Now we know how to find a game's value. But to truly be able to compare two games' values, instead of assuming we can as we have done so far, operations for comparisons need to be defined.

For all these comparisons, we will let G and H be games.

First definitions are given to the ≥ and ≤ operations:

- $G \geq H \leftrightarrow$ (no $G^R \leq H$ and $G \leq$ no H^L), or: G is greater than or equal to H if and only if there exists no Right option of G that is smaller than or equal to H and there exists no Left option of H that G is smaller than.
- $G \leq H \leftrightarrow H \geq G$, or: G is smaller than or equal to H if and only if H is greater than or equal to G.

(Conway, 2001)

Now the =, >, and < operations can be defined:

- $G = H \leftrightarrow (G \geq H \text{ } and \text{ } H \geq G)$, or: G is equal to H if and only if G is greater than or equal to H and H is greater than or equal to G.
- $G > H \leftrightarrow (G \geq$ H and H $\ngeq$ G), or: G is greater than H if and only if G is greater than or equal to H and H is not greater than or equal to G.
- $G < H \leftrightarrow H > G$, or: G is less than H is H is greater than G.

(Conway, 2001)

The next operation is the identity:

- $G \equiv H \leftrightarrow (all\ G^L \equiv some\ H^L\ and\ all\ G^R \equiv some\ H^R, and\ all\ H^L \equiv some\ G^L\ and\ all\ H^R \equiv some\ G^R)$, or: G is identical to H if and only if all Left options of G are identical to a Left option of H and all Right options of G are identical to a Right option of H, and vice versa.

(Conway, 2001)

The distinction between equality (=) and identity (≡) is necessary, for even if two games are equal, they might not be identical. For instance, if $G = \{3,1,0|\ \}$ and $H = \{3,2|\ \}$, then $G = H$ as $G = 4$ and $H = 4$, but not all G^L have a representative in H^L (namely 1 and 0) and not all H^L have a representative in G^R (namely 2). That games that are equal may not be identical poses an issue when it comes to certain operations, as, even if two games G and H are equal, the result of a function with another game I f(G,I) may not be equal to f(H,I) (Conway, 2001). If we take a number to represent the canonical form of all games with that value, then this no longer is an issue.

And the last comparing operations necessary for games:
- $G \mid\mid H \; if \; (G \ngeq H \; and \; H \ngeq G)$, or: G is confused with (or incomparable to) H if G is not greater than or equal to H and H is not greater than or equal to G.
- $G \mathrel{|\!\rhd} H \; if \; (G > H \; or \; G \mid\mid H)$, or: G is greater than or incomparable to H if G is greater than H or G is incomparable to H, which is equivalent to $G \nleq H$.
- $G \mathrel{\lhd\!|} H \; if \; (G < H \; or \; G \mid\mid H)$, or: G is smaller than or incomparable to H if G is smaller than H or G is incomparable to H.

(Albert et al, 2007)

All of these definitions work inductively. Namely, to see if a comparison of G and H holds or not, we compare some of G^L, G^R, H^L, and H^R to themselves and to G and H. With inductive proofs, a basis is needed for which the property we want to proof is true. However, when going down the chain of induction for these definitions, one will always come to the empty set, about which's elements any statement is true. Thus, no basis is required for them. (Conway, 2001)

With these definitions, certain properties about how the numbers work can be proven. The proofs themselves will not be given, but the following properties are true.
Given games $x = \{x^L|x^R\}$, $y = \{y^L|y^R\}$, and $z = \{z^L|z^R\}$,
0. $x \ngeq x^R$; $x^L \ngeq x$; $x \geq x$; $x = x$;
1. If $x \geq y$ and $y \geq z$, then $x \geq z$;
2. $x^L < x < x^R$;

Thus, surreal numbers are totally ordered[10].

3. $x + 0 \equiv x$; $x + y \equiv y + x$; $(x + y) + z = x + (y + z)$;
4. $-(x + y) = -x \pm y$; $-(-x) = x$; $x + -x = 0$;

Thus, addition of surreal numbers is commutative and associative.

5. $y \geq z \leftrightarrow x + y \geq x + z$;
6. 0 is a number; if x is a number, so is $-x$; if x and y are numbers, so is $x+y$;

Thus, surreal numbers form a totally ordered Group under addition[11].
(Conway, 2001)

With that, it turns out the surreal numbers, and thereby games, can be used much like one would use a different sets of numbers, such as the reals. [12]

[10] Meaning that any two surreal numbers can be compared.
[11] Meaning that any two additions of surreal numbers can be compared.
[12] Do note that surreal numbers also work for multiplication, addition, and other operations, which is necessary for the statement. However, those are out of the scope of this paper and thus were not discussed.

3. Strategies

In chapter 1, a strategy was defined as a planned sequence of moves to play a full game. A more formal definition of a strategy is: "*a strategy is a list of actions, exactly one at each information set of that player*". (Peters, 2015).

There are a variety of basic strategies that prove useful when solving games, when proving theorems, or when trying to improve as a human player. What follows is a non-exhaustive list of such strategies.

3.1. Greedy

Employing a *greedy strategy* means choosing the options that maximize or minimize a value of the game (Albert et al, 2007). Usually, a player that plays greedily wants to maximize their own value or minimize their opponent's value.

A greedy strategy is easy to implement and is often effective in simple games and still useful in some more complex games. A value can be assigned to a position of a game, either by estimating it like is done for *chess* or by having the players accumulating a score over the course of the game. Then calculating what move maximizes a player's score is fairly simple.

3.2. Symmetry

A player that uses a *symmetry strategy* effectively copies the strategy of their opponent. If the opponent makes a move, then the player using the symmetry strategy mimics the move. (Albert et al, 2007)
The mimicking move may be a mirrored version of the mimicked move (in *chess*, for example, there are a few openings where Black mirrors the moves made by White), or a rotated version of the move.

Using a symmetry strategy is very strong in the game of *Cram*. When played on an *m* x *n* grid, if *m* and *n* are even, then the second player can always win by playing the same move as the first player but rotated 180 degrees. Similarly, if *m* or *n* is odd (but not both), then the first player can win by playing in the centre first, and then employing the same symmetry strategy. (Albert et al, 2007)

3.3. Change the Game

Some games are more easily intuited than others. It would then be helpful if an unintuitive game could be changed into a more intuitive one. As it turns out, that is sometimes possible, as some games are the same, even if they have different rules. (Albert et al, 2007)

An example of this with widespread use in the analysis of impartial games, is that any short impartial game is equivalent to a game of *Nim*. (Conway, 2001)

3.4. Parity

The *parity* of a number states whether the number is odd or even. Parity has an intrinsic importance to combinatorial games. In normal play the first player wants a game to last an odd number of moves, while the second player wants the game to last an even number of moves, as they each win in those situations. Thus, the parity of the number of moves of a game determines the winner. (Albert et al, 2007)

In some games, this allows the second player to view the moves of the two players as a pair, as opposed to as separate moves, which might give him insights into the game he otherwise would not have gained.

3.5. Give Them Enough Rope

The *Enough Rope Principle* says to choose the move that will result in the most complicated position for the opponent. Especially when in a losing position, following the principle might turn a loss into a win. (Albert et al, 2007)

Important to note, though, is that the Enough Rope Principle is usualy only applicable when playing against another human, and generally one that is not an expert of the game being played. The point of the principle is to confuse the opponent, possibly into making a bad move, and to get more time to analyse the position (Albert et al, 2007).

Another implication of the Enough Rope Principle is that, when confused about what move to make in a position, simplifying to the point where the opponent is not confused might be harmful.

3.6. Do Not Give Them Any Rope

Contrary to the Enough Rope Principle, when in a winning position it is a good strategy to simplify the position, to not allow the opponent the chance to create a confusing situation. (Albert et al, 2007)

3.7. Strategy Stealing

Strategy stealing is a technique where a player adopts the opponent's strategy as their own. (Albert et al, 2007)

Strategy stealing can be useful to a player if their strategy seems worse than their opponent's. It is also a common solving method, and thus will get more attention in the next chapter.

4. Solving Methods

At last it is time to solve some games. However, we have never defined what it means for a game to be solved, so that will be the first order of business.

4.1. Types of Solves

When speaking of a *solved game*, there is some aspect of the outcome of the game that is known given the rules and starting position. Generally, this means that the winner given perfect play is known. It is not a seldom occurrence, however, that this is not all that is meant when a game is said to be solved. Thus, there are three definitions for what it means when a game is *solved*:

- *Ultra-weakly solved*: Given the initial position, the game's value is known,
- *Weakly solved*: Given the initial position, a strategy to obtain the game's value is known for both players, and
- *Strongly solved*: Given any legal position, a strategy to obtain that position's value is known for both players.

(Allis, 1994)

For a game that is ultra-weakly solved, even though the game's value – and thus who wins assuming perfect play – is known, the sequence of moves that follows perfect play is not (necessarily) known as well. A game that is weakly solved has a known strategy for both players that achieves the game's value, regardless of what moves the opponent makes. A strongly solved game has a strategy for both players to play perfectly from any legal position. Between these types of solves exists an order. Namely, that any strongly solved game is also weakly solved and that any weakly solved game is also ultra-weakly solved. (Allis, 1994)

Now, when speaking of a *solved game*, we can be more specific. When the specificity is deemed unnecessary or is not given by the user of the term, we know that it means that such a game is at least ultra-weakly solved.

Having defined what is means for a game to be solved, we can continue to various solving methods.

4.2. The Strategy Stealing Argument

As mentioned before, strategy stealing is a strategy that can be employed in which the player using the technique adopts their opponent's strategy as their own.

The *strategy stealing argument* is a method by which, for some games, the winning player can be proven. The method uses a proof by contradiction, in which an assumption is made about which player can win, which is contradicted, therefore proving that the other player must have a winning strategy. When using this method, we never proof what the strategy for the winning player is, making any game solved using the strategy stealing argument an ultra-weakly solved game.

To illustrate is a proof regarding the game *Hex*:

Theorem: The first player wins a game of *Hex*, assuming perfect play.

Sketch of proof: Assume that player 2 can force a win. Let Alice and Bob play two games. In the first game Bob goes first and Alice goes second, and in the second game Alice goes first and Bob goes second. As she is player 2 in the first game, Alice has a strategy *S* with which she can win. In the second game, going first, Alice can start by making an arbitrary move, and then continue as though she were player 2, using *S*. If, on a later turn of hers, *S* requires Alice to colour the hex she coloured as her first move, then Alice can colour an empty hex and continue using *S* from there. Thus, when going first Alice can win also. The assumption was that player 2 can force a win. However, following that assumption, both player 1 and player 2 have been shown to have a winning strategy. This contradiction means that the assumption must be wrong. Therefore, player 1 must have a strategy to always win *Hex*. (Albert et al, 2007; Karlin, Peres, 2017)

4.3. Top-Down Induction

4.3.1. Proofs by Induction

To prove a statement *P(n)*, a "traditional" *proof by induction*, a base case *P(0)* or *P(1)*[13] and a general case *P(k+1)* are proven. For the general case, it is assumed that *P(k)*, for some *k*, holds, which is the induction hypothesis, after which the inductive step is taken, where is proven that if *P(k)* holds, then *P(k+1)* is also true. The base case and inductive step together prove *P(n)* for all $n \geq 0$ (if 0 is the base case). (Dobson, Slomson, n.d.)

An example:

Theorem: The sum of the first *n* positive integers equals $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.

Proof: Let *P(n)* be the statement $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.

(a) Base case: For $n = 0, \frac{0(0+1)}{2} = \frac{0 \cdot 1}{2} = 0$. Thus, *P(0)* is true.

(b) Induction hypothesis: Assume that *P(k)*, where *k* is some integer, is correct, meaning that $1 + 2 + 3 + \cdots + k = \frac{k(k+1)}{2}$.

(c) Induction step: Now we will show that, if *P(k)* is correct, *P(k+1)* is also correct.

$$\begin{aligned} 1 + 2 + 3 + \cdots + k + (k+1) &= (1 + 2 + 3 + \cdots + k) + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1), \text{ by the induction hypothesis} \\ &= \frac{k^2+k}{2} + \frac{2k+2}{2} \\ &= \frac{k^2+3k+2}{2} \\ &= \frac{(k+1)(k+2)}{2} \\ &= \frac{(k+1)((k+1)+1)}{2} \end{aligned}$$

So *P(k+1)* is indeed true.

And thus, we have proven that *P(n)* is true for all $n \geq 0$.
(Dobson, Slomson, n.d.)

4.3.2. Top-Down Induction

Top-down induction, like "traditional" induction, uses a smaller case to prove something about a larger case. However, the method is different: to prove a statement *P(n)* for all $n \in N$, a proof by top-down induction may at any time assume *P(k)* is true for some $k < n$. When making such an assumption, "by induction" must be written. When the proof is done, a review must be done to make sure any base cases are proven. Two perks of top-down induction are (1) that base cases are often not necessary, as they are often vacuously true and (2) that it works for any set *N* and any partial ordering[14] <. (Albert et al, 2007)

To illustrate, we will prove the same theorem as we did with "traditional" induction:

Theorem: The sum of the first *n* positive integers equals $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$.

Proof: Let *P(n)* be the statement $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.

(a) If *P(n)* is true, then $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$ holds.

$$\begin{aligned} 1 + 2 + 3 + \cdots + n &= (1 + 2 + 3 + \cdots + n - 1) + n \\ &= \frac{(n-1)((n-1)+1)}{2} + n, \text{ by induction} \end{aligned}$$

[13] In proofs by induction, the base case is usually n=0 or n=1.
[14] Meaning that there are elements in *N* that can be compared, such that one precedes the other.

$$= \frac{(n-1)n}{2} + \frac{2n}{2}$$
$$= \frac{n^2-n+2n}{2}$$
$$= \frac{n^2+n}{2}$$
$$= \frac{n(n+1)}{2}$$

And the base case, $n = 0$, is also holds, as $\frac{0(0+1)}{2} = 0$.

Thus, for any $n \geq 0$, *P(n)* is true.
(Albert et al, 2007)

Hopefully the example made the workings of top-down induction clear, for it is the type of induction used most often in combinatorial game theory, and can (help) solve games, as we will now see:

There is a variant on the game *Nim*, where there is only one heap of *n* counters, but where a player can take a minimum of one and a maximum of 10 counters on their turn. This variant can be solved with top-down induction.

Theorem: The second player wins if and only if $n \bmod 11 = 0$.

Proof:
(a) If $n \bmod 11 = 0$, then any legal move leaves a heap that is not equivalent to 0, which loses by induction.
(b) If $n \bmod 11 \neq 0$, but instead $n \bmod 11 = a$, then player 1 can take a counters, leaving $n - a$ counters, which modulo 11 will be 0. Continuing, player 1 can use the strategy named in (a), acting as though they were player 2, and win.

No base case is required for this proof, as the base case would be $n = 0$, for which any statement about "all legal moves" is vacuously true.

So, if $n \bmod 11 = 0$, then player 2 has a winning strategy, but if $n \bmod 11 \neq 0$, then player 1 has a winning strategy. In other words, the second player wins if and only if $n \bmod 11 = 0$.
(Albert et al, 2007)

4.4. The Minimax Algorithm

The *minimax algorithm* computes computes the best move in a game or position. The algorithm starts by looking at the leaves of the game's tree, which it calculates the values of. Then, recursively, going backwards, it makes the rest of the tree. A game tree made by the minimax algorithm is different from a normal game tree, as it only considers the options of the player whose turn it is and not the options of both players. Each position is also given a value that is determined by finding which of the position's options gives the maximum value if the position is player 1's turn and the minimum value if the turn is player 2's. By looking at the whole game tree this way, the best move of any position of a game is found. This makes a game solved with the minimax algorithm a strongly solved game. (Russel, Norvig, 2010)

The minimax algorithm finds use, for example, in chess engines/AI. However, the game of chess is much too large in the number of options to calculate the full game tree. Instead, a chess engine that uses the minimax algorithm searches all moves up to some depth (number of moves). Then the engine evaluates the leaves, which are unlikely to be final positions of the game, based on various factors. These are the values used for the rest of the minimax algorithm. (AI Chess Algorithms, 2003)

4.5. Retrograde Analysis

Retrograde analysis approaches a game not from the initial position but from the final ones. Starting with the final positions, a computer using retrograde analysis would calculate what previous moves there are, and that way construct a game tree in reverse. Doing so, the outcome of any sequence of moves can be known. (Neeleman, 2015; Van den Herik et al, 2001)

Retrograde analysis is used a lot for the construction of endgame databases for games too complicated to fully solve using the method. For example, retrograde analysis was one of two solving methods used for the game *fanorona*. (Schadd et al, n.d.)

4.6. Decomposition Search

The *decomposition search* method finds the best move in a game or position, like the minimax algorithm, but can handle much larger problems than it, as decomposition search uses concepts from combinatorial game theory to improve the search. (Muller, n.d.)

Decomposition search uses a four-step algorithm to determine the best move in a game G, like so (from Muller, n.d.):

Let G be a game that can be written as the sum of its subgames: $G = G_1 + \cdots + G_n$[15] and let the combinatorial evaluation of G be $C(G)$.

4.6.1. Game Decomposition and Subgame Identification

The first step is to decompose G into its subgames, in other words, to find the sum of subgames $G = G_1 + \cdots + G_n$. How and if this can be done depends on the game being analysed.

4.6.2. Local Combinatorial Game Search (LCGS)

LGCS makes a game tree of relevant moves individually for all subgames. A tree made by LCGS is different from one made by a minimax algorithm, as such an algorithm only looks at the options of the player whose turn it is in a position, whereas LCGS considers all options, like how game trees were defined in chapter 2.

This means that LCGS must generate every legal move for every option for every subgame for both players. Unless, that is, an option can be *pruned* or is a *terminal position*.
An option can be *pruned*, meaning that it and its branches are removed from the tree, if it is equivalent to another position or if it is dominated (there exists a move with a greater value if it is Left's move, or a move with lesser value if it is Right's move) by another position, which do not help in finding the best move. LCGS defines a *terminal position* as a position with (1) no legal moves; (2) no good move, the position is seen as constant; or (3) the value of the position is already known. In cases 1 and 2, the positions are evaluated and given a local score. In the third case, which might arise for example if the position is a transposition[16] of another sequence of moves, the local score of the position is already known, and thus LCGS needs not look further into the tree.

4.6.3. Local Evaluation

Local evaluation calculates the value of a game given the values of its game tree's leaves. Let the players be Black and White, where a positive value is good for Black, and name the moves for Black from local position p $b_1, \dots, b_n$ and name the moves for White $w_1, \dots, w_n$. These are the leaves of p's game tree, of which the values have been calculated by LCGS. The evaluation of p is

$$C(p) = \{C(b_1), \dots, C(b_n) | C(w_1), \dots, C(w_n)\},$$

which can be reduced to the canonical form.
This calculation can be repeated from the leaves to the root, evaluating each node in the game tree for each subgame.

---

[15] Note that this only states that the subgames exist. To make use of them, they must be calculated first.
[16] If two or more different sequences of moves result in the same position, they are transpositions.

4.6.4. Sum Game Play

The final step is to select what move to play. This is done by calculating the incentives, which determine how much a player gains by playing that move, in all subgames. The optimal move is defined as the move with an incentive that dominates all others. It is possible, though, that there is no one incentive that dominates all others, in which case the optimal move is found by a more complex procedure.

Conclusion

It turns out that the question *"How are combinatorial games solved?"* has multiple answers.

In chapter 1 we discovered what game theory is, and some important terminology that is important for the mathematical field: *competitive*, *cooperative*, *evolutionary*, and *bargaining games*; *one-shot*, *repeated*, and *extensive form games*; the attributes *finite* and *(im)perfect information*; and *payoff*, *strategies*, and *options*, and a few game types: *zero-sum games*, *nonzero-sum games*, *finite extensive form games*, and a brief look at *combinatorial games*, while also learning about a bit of set theory.

Then in chapter 2, we defined what a combinatorial game, namely a set of its options: $G = \{\mathcal{G}^L|\mathcal{G}^R\}$, and how to express is as a set of options or as a game tree. Furthermore, with help of fundamental theorem of combinatoric games, we defined the outcome classes $\mathcal{N}$ (fuzzy), $\mathcal{P}$ (zero), $\mathcal{L}$ (positive), and $\mathcal{R}$ (negative), which are the first glimpse into solving a game works. Building from that, we found a way to give values to games and to use these as *surreal* numbers with which we can do arithmetic such as adding a subtracting. And with the arithmetic for the *surreal numbers*, we defined ways to compare them, again getting closer to an answer for the research question.

In chapter 3, some basic strategies for combinatorial games were discussed: the *greedy strategy*, the *symmetry strategy*, changing the game to another, using *parity*, giving the opponent enough or no rope, and *strategy stealing*.

At last, in chapter 4, we got to a few solving methods using knowledge from the previous two chapters. But not before defining what a "solved game" is, instead of going by a vague understanding. That way, we found that there are three ways in which a game can be solved: *ultra-weakly*, *weakly*, or *strongly*. And then were the solving methods. First was the *strategy stealing argument*, which can ultra-weakly solve a game by using the *strategy stealing* technique, as we showed with *Hex*. Next was *top-down induction*, using a specific type of mathematical induction to determine who should win, as showed with the example with a *Nim* variant which we solved ultra-weakly. The *minimax algorithm* and *retrograde analysis* are not specific for combinatorial game theory, but show very common techniques to strongly solve games and thus are important. Finally, we explained the *decomposition search* method and its four-step algorithm to strongly solve games in a, at ground level, similar way to the *minimax algorithm*, but which uses a variety of concepts from combinatorial game theory to improve the process.

To conclude the paper, the answers to *"How are combinatorial games solved?"* that we have found are: by using the *strategy stealing argument*, by using *top-down induction*, by using the *minimax algorithm*, by using the *retrograde algorithm*, and by using *decomposition search*.

References

AI chess algorithms. (2003). Retrieved Januari 7, 2024, from https://www.cs.cornell.edu/boom/2004sp/ProjectArch/Chess/algorithms.html

Albert, M. H., Nowakowski, R. J., & Wolfe, D. (2007) *Lessons in play: An introduction to combinatorial game theory*. A K Peters.

Allis, L. V. (1994). *Searching for solutions in games and artificial intelligence*. Rijksuniversiteit Limburg. https://doi.org/10.26481/dis.19940923la

Apt, K. R., Löding, C., Jurdziński, M., Grädel, E., Kučera, A., Doyen, L., Rasking, J.-F., Kreutzer, S., & Halpern, J. Y. (2011). *Lectures in game theory for computer scientists*. Cambridge University Press. https://books.google.nl/books?hl=en&lr=&id=K7Q-eOxYJ2wC

Bartlett, P. (2006). *A short guide to Hackenbush*.

Blom, A. (2021). *Combinatorial game theory: An introduction guided by the game of go*.

Bonanno, G. (2018). *Game theory* (2nd edition).

Brams, S. J. (2004). *Game theory and politics*. Dover Publications. https://books.google.nl/books?hl=en&lr=&id=ft0oAwAAQBAJ

Conway, J. H. (2021). *On games and numbers* (2nd edition). A K Peters.

Dobson, K., & Slomson, A. (n.d.) *Proof by induction*. Mathcentre community project.

Jaeger, G. (2008). *Applications of game theory in linguistics*.

Karlin, A. R., & Peres, Y. (2017). *Game theory, alive*. American Mathematical Society. https://books.google.nl/books?hl=en&lr=&id=9_vFDgAAQBAJ

McNamara, J. M., & Leimar, O. (2020). *Game theory in biology: Concepts and frontiers*. Oxford University Press. https://books.google.nl/books?hl=en&lr=&id=UZr9DwAAQBAJ

Muller, M. (n.d.) *Decomposition search: A combinatorial games approach to game tree search, with applications to solving go endgames*.

Neeleman, J. (2015). *Optimizations and generalisations of retrograde analysis*.

Osborne, M. J. (2020). *An introduction to game theory: by Martin J. Osborne*.

Peters, H. (2015). *Game theory: A multi-leveled approach* (2nd edition).

Ross, D. (2023) *The Stanford encyclopedia of philosophy*: *Game theory*. Edward N. Zalta & Uri Nodelman (eds.). https://plato.stanford.edu/archives/fall2023/entries/game-theory/

Russel, S., & Norvig, P. (2010) *Artificial intelligence: A modern approach* (3rd edition).

Samuelson, L. (2016). Game theory in economics and beyond. *Journal of Economic Perspectives, 30*(4), 107-130.

Schadd, M. P. D., Winands, M. H. M., Uiterwijk, J. W. H. M., van den Herik, H. J., & Bergsma, M. H. J. (n.d.). *Best play in fanorona leadsto draw*.

van den Herik, H. J., Uiterwijk, J.W.H.M., & van Rijswijck, J. (2001). *Games solved: Now and in the future*. Elsevier Science.

von Neumann. J. (1928). *Zur Theorie der Gesellschaftsspiele*. Mathematische Annalen, 100, 295– 320.

von Neumann, J., & Morgenstern, O. (1944/1947). *Theory of games and economic behavior*. Princeton: Princeton University Press.

Zermelo, E. (1913). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the fifth international congress of mathematicians* (Vol. 2, pp. 501-504) Cambridge: Cambridge University Press.

Index A: Terminology

Index B: Symbols

Appendix: Rulesets

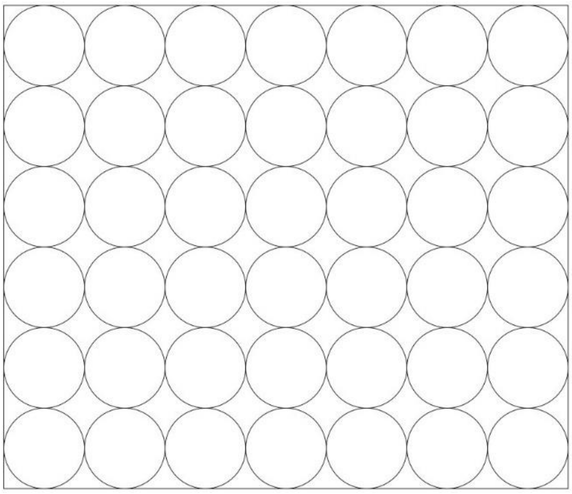
Cram

Cram is a variant of *domineering*. See *domineering* for the rules.

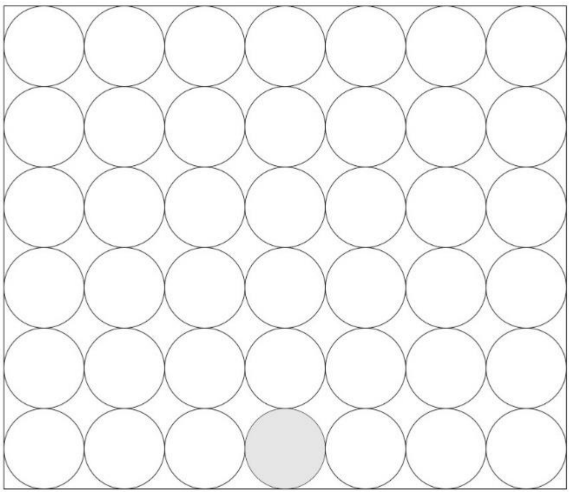
The difference between *cram* and *domineering*, is that in *cram* both players may place their dominoes in a vertical or horizontal orientation.

Connect Four

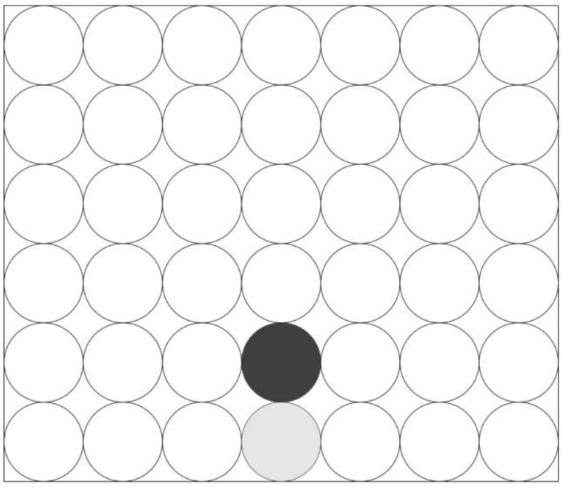
Connect four is played on a grid of (as is the standard) seven columns and six rows:



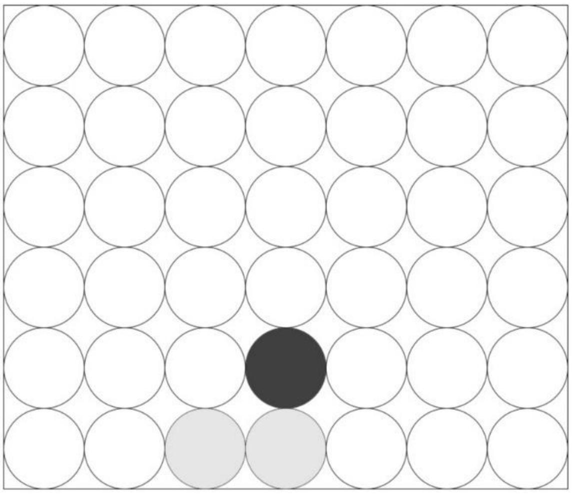
Two players, Yellow (dark grey) and Red (light grey), alternate placing a stone of their colour in the grid. A stone is placed by choosing a column. The stone then falls to the lowest spot in the grid that does not contain another stone.

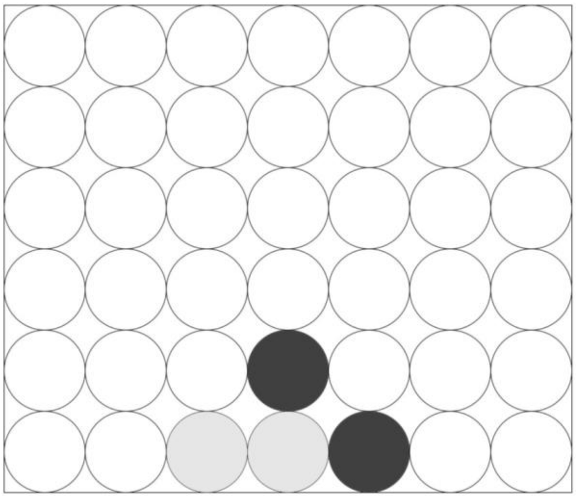


A first move (by Red)

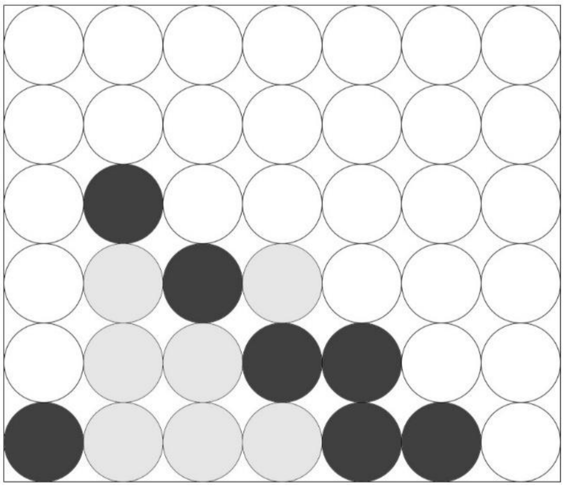


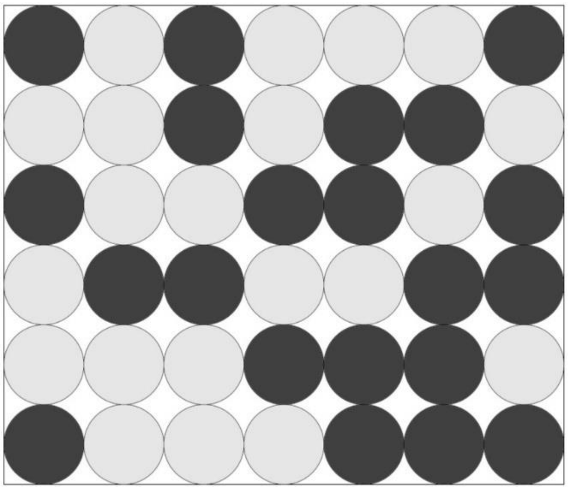
A second move (by Yellow)


A third move (by Red)


A fourth move (by Yellow)

If a player is able to connect four of their stone either vertically, horizontally, or diagonally, they have won. If the whole board is filled with no four connected stones, then the game is a draw.

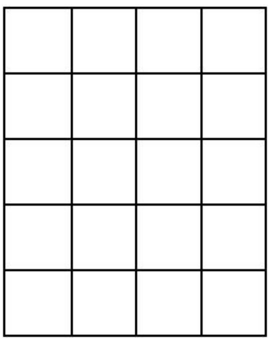

Yellow has four (diagonally) connected stones. She has won.


The board is full and neither player has managed to connect four of their stones. The game is a draw.

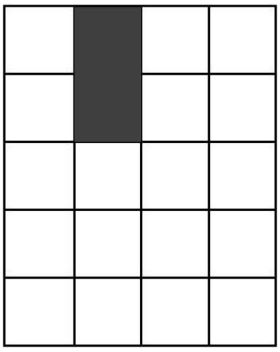
Connect four also has variants with differently sized grids and variants where the number of stones that need to be connected for a win is different. If *n* connected stones give a win, then the game is called *connect n*.

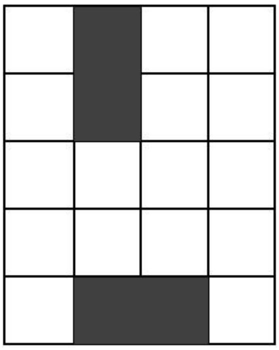
Domineering

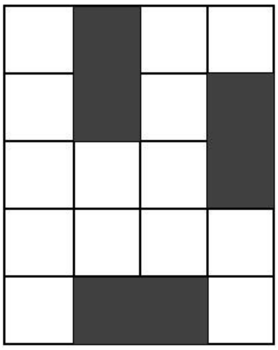
Domineering is played on an arrangement of squares, usually a grid, like so:

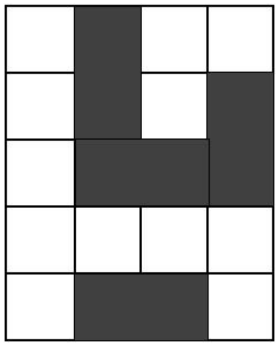


Two players, Left and Right, take turns placing a domino. Left may only place her dominoes in a vertical orientation and Right may only place his dominoes in a horizontal orientation. A domino must be placed in such a way that is covers exactly two squares and may not be placed (partially) outside the board nor (partially) on another domino.

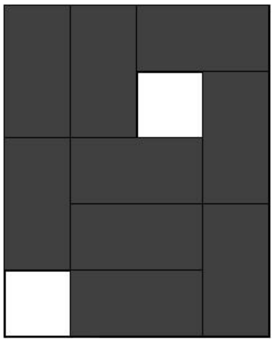

A first move (by Left)


A second move (by Right)


A third move (by Left)

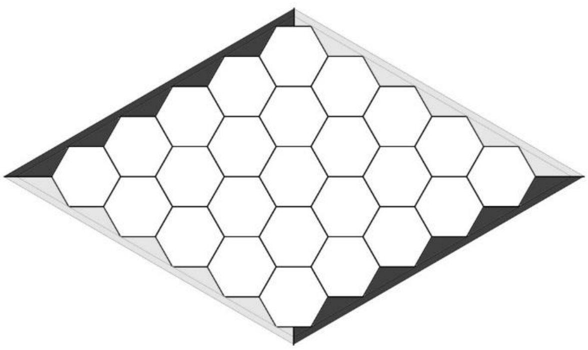

A fourth move (by Right)

The winner is the last player that is able to place a domino.

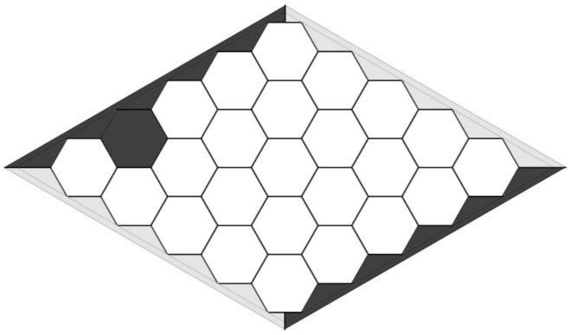

It is Right's turn, but he has no legal move. Thus, Left has won.

Hex

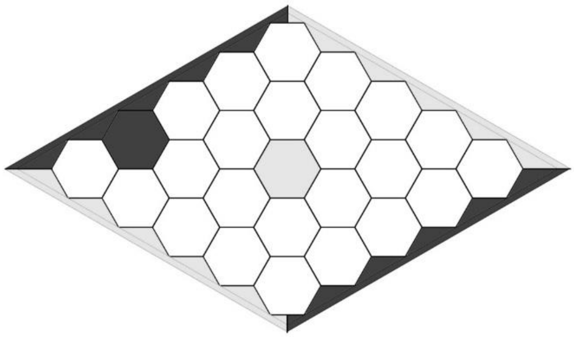
Hex is played on a rhombic grid of hexagons. The standard is 11x11, but other sizes, such as 13x13 and 19x19 are also common. A 5x5 version is shown here:



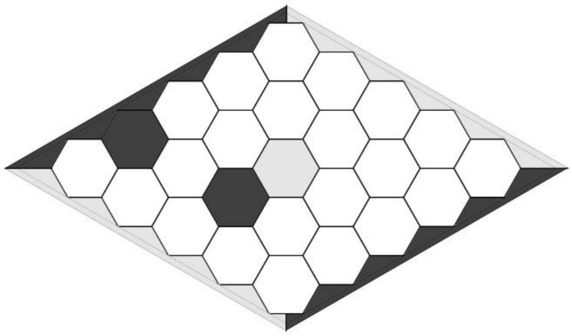
Two players, Black (dark grey) and White (light grey), take turns colouring an empty hexagon (hex) in their colour. Each player colours exactly one hex.



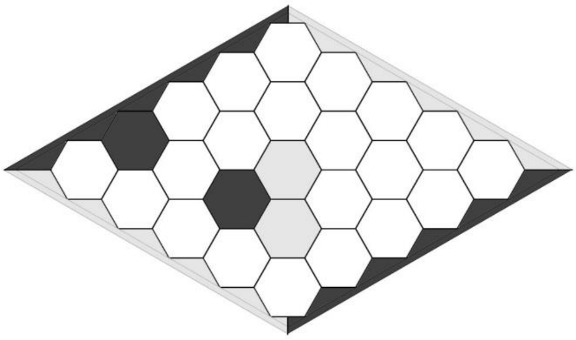
A first move (by Black)



A second move (by White)

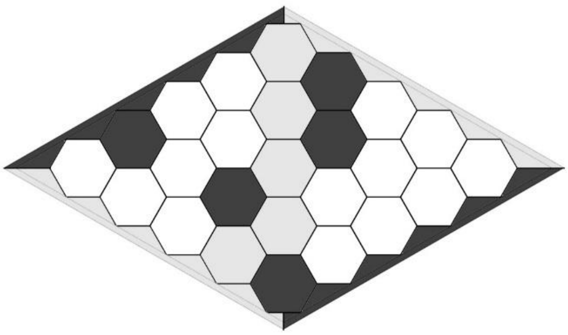


A third move (by Black)



A fourth move (by White)

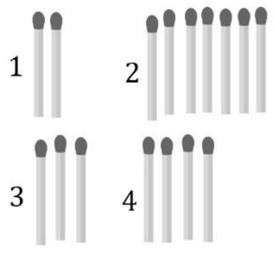
Each player has two opposite sides of the board coloured in their colour. A player wins if they manage to connect both of their sides of the board by a linkage of neighbouring hexes of their colour.



White has connected their sides with a chain of white hexes and has won.

Nim

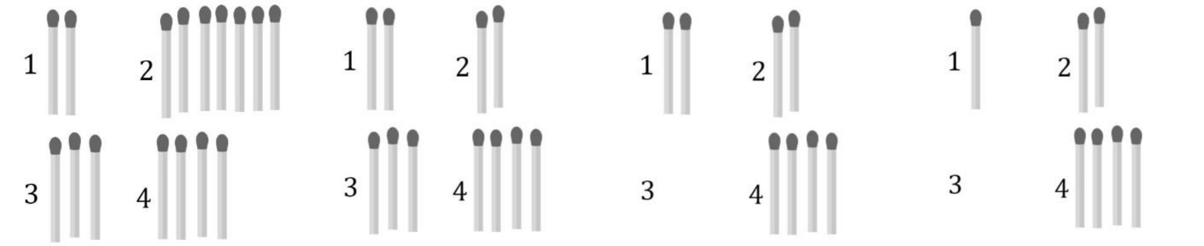
A game of *Nim* is played with counters. These counters can be stones, coins, match sticks, etc. Heaps (or piles) are made from some number of counters. The number of counters in each heap may differ.
Here is an example of a starting position of a game of *Nim*, where the counters are rectangles, and the heaps are marked with numbers.



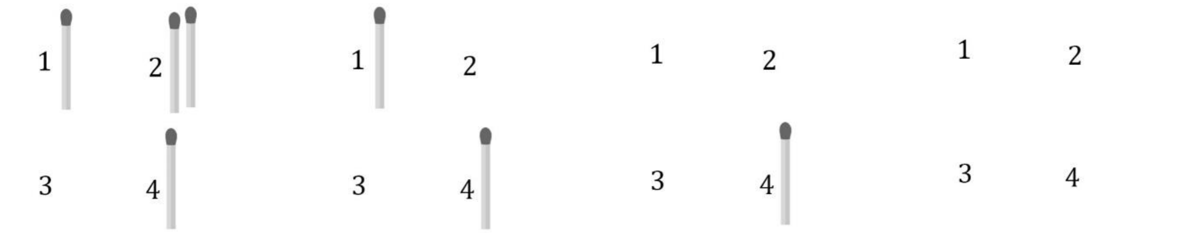
Two players alternate moves. A move consists of taking at least one (with no maximum) counter from one of the heaps.

The game continues until no counters are left. The player to take the last counter (or the last heap) is the winner.

A full game of Nim



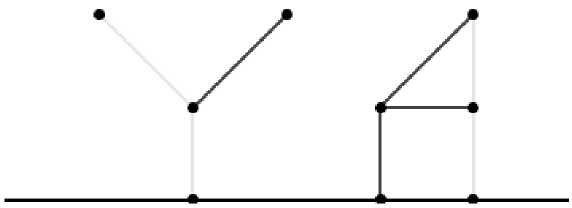
Player 1 removes 5 counters from heap 2. *Player 2 removes 3 counters from heap 3.* *Player 1 removes 1 counter from heap 1.*



Player 2 removes 1 counter from heap 4. *Player 1 removes 2 counters from heap 2.* *Player 2 removes 1 counter from heap 1.* *Player 1 removes the last counter and wins.*

Red-Blue Hackenbush

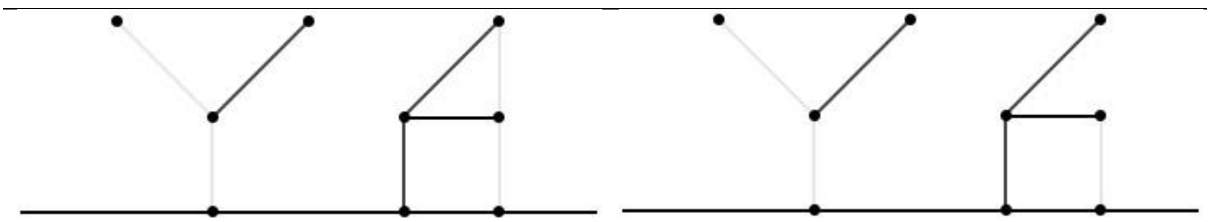
Red-Blue Hackenbush is played on a piece of paper, a white board, a chalkboard, etc. A thick horizontal black line is the ground, on which a picture made of red (light grey) and blue (dark grey) lines is drawn. A game of R*ed-Blue Hackenbush* may look like this:

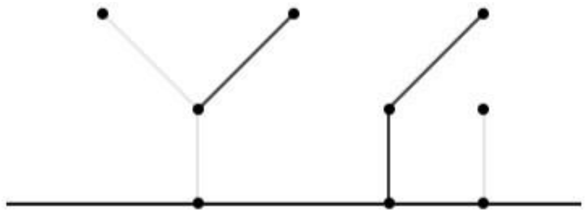


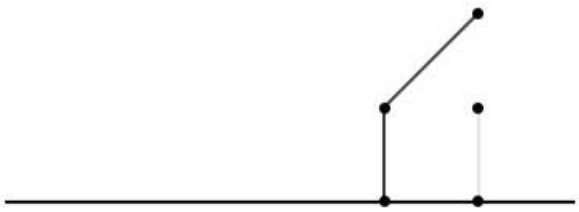
Two players, Left and Right, take turns erasing lines. On Left's turn, they must erase any one blue line. On Right's turn, they must erase any one red line. If, after a line is erased, one or more lines are not connected to the ground (they are floating), then those are erased also.
Note: the ground cannot be erased.

The last player to erase a line, is the winner.

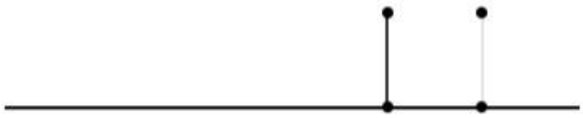
A full game of Red-Blue Hackenbush







In the position above, Right removed the bottom line of the "tree" on the left. Because this disconnected the lines on it from the ground, they were removed too.



It is Right to move, but he cannot. Thus, Left has won.