MASTER'S DEGREE THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Combinatorial Game Theory: The Dots-and-Boxes Game

Author: Alexandre Sierra Ballarín

Advisor: Anna Lladó Sánchez

Department: Matemàtica Aplicada IV

Academic year: 2015-16

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master Thesis

Combinatorial Game Theory: The Dots-and-Boxes Game

Alexandre Sierra Ballarín

Advisor: Anna Lladó Sánchez

Departament de Matemàtica Aplicada IV
Barcelona, October 2015

I would like to thank my advisor Anna Lladó for help and encouragemt while writing this thesis.

Special thanks to my family for their patience and support.

Games were drawn using LaTeX macros by Raymond Chen, David Wolfe and Aaron Siegel.

Resum

Paraules clau: Dots-and-Boxes, Jocs Imparcials, Nim, Nimbers, Nimstring, Teoria de Jocs Combinatoris, Teoria d'Sprague-Grundy.

MSC2000: 91A46 Jocs Combinatoris, 91A43 Jocs amb grafs

La *Teoria de Jocs Combinatoris* és una branca de la Matemàtica Aplicada que estudia jocs de dos jugadors amb informació perfecta i sense elements d'atzar. Molts d'aquests jocs es descomponen de tal manera que podem determinar el guanyador d'una partida a partir dels seus components. Tanmateix això passa quan les regles del joc inclouen que el perdedor de la partida és aquell jugador que no pot moure en el seu torn. Aquest no és el cas en molts jocs clàssics, com els escacs, el go o el *Dots-and-Boxes*. Aquest darrer és un conegut joc, els jugadors del qual intenten capturar més caselles que el seu contrincant en una graella quadriculada. Considerem el joc anomenat *Nimstring*, que té gairebé les mateixes regles que *Dots-and-Boxes*, amb l'única diferència que el guanyador és aquell que deixa el contrincant sense jugada possible, de manera que podem aplicar la teoria de jocs combinatoris imparcials. Tot i que alterant la condició de victòria obtenim un joc completament diferent, parafrasejant Berlekamp, Conway i Guy, "no podem saber-ho tot sobre *Dots-and-Boxes* sense saber-ho tot sobre *Nimstring*".

L'objectiu d'aquest projecte és presentar alguns resultats referits a *Dots-and-Boxes* i *Nimstring*, com guanyar en cadascun d'ells, i quina relació hi ha entre ambdós, omplint algunes llacunes i completant algunes demostracions que només apareixen presentades de manera informal en la literatura existent.

Abstract

Keywords: Combinatorial Game Theory, Dots-and-Boxes, Impartial Games, Nim, Nimbers, Nimstring, Sprague-Grundy Theory.

MSC2000: 91A46 Combinatorial Games, 91A43 Games involving graphs

Combinatorial Game Theory (CGT) is a branch of applied mathematics that studies two-player perfect information games with no random elements. Many of these games decompose in such a way that we can determine the outcome of a game from its components. However this is the case when the rules include the *normal play convention*, which means that the first player unable to move is the loser. That is not the case in many classic games, like *Chess*, *Go* or *Dots-and-Boxes*. The latter is a well-known game in which players try to claim more boxes than their opponent. We consider the game of *Nimstring*, which has almost the same rules as *Dots-and-Boxes*, slightly modified by replacing the winning condition by the normal play convention so we can apply the theory of impartial combinatorial games. Although altering the winning condition leads to a completely different game, paraphrasing Berlekamp, Conway and Guy, "you cannot know all about *Dots-and-Boxes* unless you know all about *Nimstring*".

The purpose of the project is to review some results about Dots-and-Boxes and Nimstring, how to win at each one and how are they linked, while filling in the gaps and complete some proofs which are only informally presented in the existing literature.

Notation

$G' \in G$	G' is an option of G (Definition 1)
$\bar{G}$	A position of game G (Definition 3)
$\mathcal{P}$	Set of games in which the previous player wins (Definition 6)
$\mathcal{N}$	Set of games in which the next player wins (Definition 6)
$\mathcal{P}$*-position*	A game which belongs to $\mathcal{P}$ (Definition 6)
$\mathcal{N}$*-position*	A game which belongs to $\mathcal{N}$ (Definition 6)
$G = H$	Equal games (Definition 13)
$*n$	Nimber (Definition 16)
$mex(S)$	Minimum excluded number from set S of non-negative numbers (Definition 18)
$\mathcal{G}(G)$	Nim-value of game G (Subsection 6.2 of Chapter 1)
$\oplus$	Nim-sum (Definition 25)
g	The distinguished vertex in Nimstring called *the ground*. (Subsection 1.2 of Chapter 2)
☽	Loony option of a game. (Definition 32)
j	Number of joints of a game (Definition 40)
v	Total valence of a game (Definition 40)
$d(G)$	Number of double-crosses that will be played on strings of G (Definition 42 and Proposition 43)
$e(G)$	Number of edges of game G (Notation 44)
$b(G)$	Number of boxes (nodes) of game G (Notation 44)
$Previous$	The previous player (Notation 44)
$Next$	The next player (Notation 44)
$\equiv$	Congruent modulo 2 (Notation 46)
BIG	Any game G such that $\mathcal{G}(G) \geq 2$ (Notation 57)
G^*	A loony endgame that is a position of game G (Notation 57)
$Right$	The player in control in a loony endgame (Notation 64)
$Left$	The first player to play in a loony endgame (Notation 64)
$V(G)$	Value of game G (Definition 67)
$V(G\|H)$	Value of game G assuming H is offered (Definition 68)
$V_C(G\|H)$	Value of game G assuming Left offers H and Right keeps control (Definition 68)
$V_G(G\|H)$	Value of game G assuming Left offers H and Right gives up control (Definition 68)
$FCV(G)$	Fully controlled value of game G (Definition 76)
$CV(G)$	Controlled value of game G (Definition 79)
$TB(G)$	Terminal bonus of game G (Definition 81)

Contents

List of Figures

Introduction

In 1982, Berlekamp, Conway and Guy published the seminal work in combinatorial game theory *Winning Ways (for your Mathematical Plays)* [**Ber01**]. In more than 800 pages they not only tried to solve many two-person games, but also described a theory that they had developed to encompass them all. Many of these games had a common characteristic: they decomposed into sums of smaller ones. Thanks to this, games can be considered to be formed by components that are also games, in such a way that we can determine the outcome of a game from its components.

A combinatorial game, as defined in *Winning Ways*, is a two-player, complete information game without random elements, that is guaranteed to end in a finite number of turns, in which the player who is unable to complete his turn loses. This winning criteria is called *normal play*, and it is key so that the game decomposes.

Some advances have been made in games that do not verify all the above conditions. For instance, in loopy games (where a previous position can be repeated), or in misère games (where the player unable to play wins instead of losing). Although many games were considered in *Winning Ways* and other games have been considered since, it has been hard to find classic games where Combinatorial Game Theory (CGT) can be applied. Many classic games do not verify the *normal play* condition: in chess the goal is to checkmate your opponent, in tic-tac-toe, gomoku and similar games the goal is to achieve a straight line, in hex you have to create a continuous path between two opposite sides of the board, in go and mancala games you have to outscore your opponent, etc. Notice that all these games verify all the other conditions of combinatorial games, so we can consider the *normal play* winning condition the main obstacle to include classic games in the standard definition of combinatorial game.

The main goal of this work is to analyse the combinatorial game known as Dots-and-Boxes (also known as Dots). Although we only have evidence of the existence of Dots-and-Boxes as far as the late 19th century (it was described by Édouard Lucas [**Luc**] in 1889), and it is therefore younger than most classic games, Dots-and-Boxes is widely known and played, probably because of its easy rules and the fact that it can be played with just pencil and paper, without need of a board or counters. Dots-and-Boxes is a rare case among classic games where CGT not only can be applied but where it is required in order to master the game. Any high-level player must be aware of the underlying combinatorial game theory.

Dots-and-Boxes, as the other classic games mentioned before, does not use the *normal play* winning condition. In order to circumvent this difficulty, a slightly modified game is considered. Nimstring has the very same rules of Dots-and-Boxes, except the winning condition: it uses *normal play*, that is, first player unable to move loses. We could say that knowledge on Dots can be subdivided in three areas: specific game knowledge that does not use CGT, knowledge about Nimstring (using CGT), and the link between both games (how Nimstring helps to win at Dots).

Another apparent handicap of Dots is that it is an impartial game. A combinatorial game is impartial when both players have the same options. Impartial games tend to be difficult to play, because you seem to lack a *sense of direction* (to begin with, there are no pieces you can call your own). Surprisingly enough, this difficulty is not a major barrier in the game Dots-and-Boxes.

Winning Ways devoted a whole chapter to Dots-and-Boxes, centered on Nimstring and some basic techniques specific to Dots. Two basic concepts in this chapter were those of *chain* and the *chain rule*: in a Nimstring game each player should aim to obtain a given parity of chains in order to win. This fact helps to tackle the *impartiality* of Nimstring, because each player has a clear goal since the beginning of the game.

Impartial games form a subset of combinatorial games that was, to some extent, already solved by the theory developed by Sprague and Grundy for Nim-like games. In short: as long as we are able to find some value of a game, called the *nim-value*, a winning strategy is to move to a position whose nim-value is zero. If no such a position is available, then the player who last moved will win the game assuming perfect play. In particular, this theory applies to Nimstring. The problem is that, in general, finding the *nim-value* of a position is not easy.

In 2000, a book by Berlekamp [**Ber00**], interspersed with problem chapters, summarized the *Winning Ways* chapter on Dots and included some new insight on the link between Nimstring and Dots-and-Boxes. One result determines conditions in which we can assume that a position has the parity of its number of chains fixed even though the chains are *not there* yet, so we can *forget* components already resolved from a sum of games. Another result combines the two ways of studying a game, chain counting and nim-values, so that we can use both simultaneously when analysing a game. These important results were presented without proof.

The purpose of this project is to review the results on the Dots-and-Boxes game by filling in some gaps and complete the proofs that are not formally presented.

As for the structure of our work, in Chapter 1 we introduce impartial combinatorial games: its representation as the set of the possible positions reachable in a single move, what is a winning strategy, and the outcome classes (who wins). How to determine the winner by determining the $\mathcal{P}$*-positions*. We explain how to sum disjoint games, and which games are equivalent. We introduce *nimbers* and *nim-values*, that lead us to the Sprague-Grundy theory for impartial games, which allows us to match any impartial game to a number, and reduce solving an impartial game to knowing its nim-value. Finally we show some examples of impartial games.

In Chapter 2 we introduce the rules of several related games: Dots-and-Boxes, Strings-and-Coins and Nimstring. We show the difference between short and long chains. We expose the strategy of *keeping control*, which is very useful in Dots-and-Boxes endgames, and set the goal of reaching the last stage of the game, known as *loony endgame*, being the second to play. We show how to compute the nim-value of a game recursively. Then we hint how knowing how to win at Nimstring may help to win at Dots-and-Boxes. Finally, we show what happens in the stage that precedes the loony endgame, the short chains' exchange.

In Chapter 3 we center the study on Nimstring. We extend loony endgames to a more general category than in the previous chapter, where all LE were simple, by generalising the notion of *independent chain* to that of *string*. We introduce the key concept of *double-cross*. We provide, as an original contribution, a generalisation of the chain formula in WW, which links the winner with the number of double-crosses that will be played.

In Chapter 4 we simplify the number options available to a player by considering canonical moves (in both Nimstring and Dots). Then we prove that games which nim-value is 0 or 1 can be considered resolved, so that in any game we only need to worry about components whose nim-value is ≥ 2. We also prove that these components of higher value cannot be considered resolved.

In Chapter 5 we go back to Dots-and-Boxes, analyzing the optimal play by each of the players in the loony endgame, as well as what will be the final score differential, called the *value* of the game. We define a lower bound of the value of a game, the *controlled value*, which is easier to compute, and in certain conditions is equal to the actual value. We end the chapter exposing what is known about the computational complexity of Dots-and-Boxes.

A more detailed outline of the main contributions of this work is presented in the Conclusions. We refer the reader to that part to clarify the original parts of the work.

Chapter 1
Impartial games

In this chapter we introduce impartial combinatorial games. Impartial games can be partitioned in two classes, depending on which player can win with perfect play, either the next player to move or his opponent. We define game equivalence and show how a game with disjoint components can be expressed as a sum of these components. This leads to prove that games with this addition are an abelian group. Then we discuss the game called Nim and the Sprague-Grundy theory, which shows that any impartial game is equal to some simple Nim game. This allows us to solve any impartial game as long as we can "translate" it into its equivalent Nim game. We end the chapter with some examples of impartial games.

1. Combinatorial Games and Impartial Games

A *combinatorial game* has the following characteristics:

- It is played by *2 players*.
- From each position there is a set of possible moves for each player. Each set is called the *set of options* of the respective player.
- *Games must end*: there must be some ending condition, and the game must end in finitely many turns. In particular, a position cannot be repeated.
- The player that is unable to complete his turn loses the game. This is the *normal play convention*. In particular, there are no ties.
- *Perfect information*: all information related to the game is known by both players.
- There are *no random elements*, such as dice.

In fact, as mentioned in the introduction, combinatorial game theory can be applied to games that do not verify all the conditions above, as it is the case with Go (see [**Ber94**]), Chess (see [**Elk**]) and Dots-and-Boxes, but in any case the games being considered always verify the last two conditions.

An *impartial combinatorial game* verifies an additional condition:

- Both players have the same options, that is to say, the moves available to one player are the same as the moves available to the other player.

In particular, games where each player has his own set of pieces are not impartial, because no player can move his opponent's pieces[1].

1.1. Representation of Impartial Combinatorial Games. Let us describe some usual ways to represent impartial combinatorial games formally.

First of all, we can consider than an impartial combinatorial game is a directed graph with labelled arcs, where each node represents a position and each arc a possible move (option) from that position. The ending condition can be represented in two ways: in a more general setting some nodes are labelled terminal, so the game ends when one of them is reached, while when playing with the normal play convention it is enough to consider that the game ends when a node without outcoming arcs is reached, being the player who last played (i.e., the one who reached that node) the winner. Therefore we can describe an impartial game by an acyclic directed graph $G = (V, A)$ with set of vertices V and set of arcs $A : V \to V$, with a distinguished vertex $v_0 \in V$ that represents the starting position. The elements of V represent the different states of the game that can be reached from the starting position during the game. A complete sequence of play is any path $\{v_0, v_1, v_2, \cdots v_k\}$ beginning in v_0 and ending in some v_k without outgoing arcs. Each arc $(v_l, v_{l+1}), \forall l \in \{0, 1, \ldots k-1\}$ of the path corresponds to a turn. When l is even the turn has been played by the player who made the first move, while when l is odd it has been played by his opponent.

We can also consider that a game is simply defined by its options. That is to say, if from G the possible moves are to $G_1, \cdots, G_n$, we will write $G = \{G_1, G_2, \cdots, G_n\}$. This is how we will usually represent games, and for this reason we formally define impartial games in the following way:

DEFINITION 1. *An* impartial game *G is a set of impartial games $G = \{G_1, G_2, \cdots, G_n\}$. The elements of the set are called* options *of G. All impartial games are constructed in this way.*

Note that this self-referring definition is correct: we can build all games hierarchically, considering at each level the games whose options where "created" in previous levels. Level 0 is solely formed by the game without options $G_0 = \{\}$ (according to the definition, G_0 is a game). At level 1 we have $G_1 = \{G_0\}$, the game whose only option is to move to G_0. At level 2 we have $G_{21} = \{G_0, G_1\}$ and $G_{22} = \{G_1\}$, and so on. That is to say, at each level the games whose options belong to lower levels and have not been considered yet "are created"[2]. The second sentence in the definition tells us that there are no games outside this construction.

Then it is sound to use the following

NOTATION 2. *If G' is an option of G we will write $G' \in G$.*

1.2. "Game" versus "position". Before continuing we need to clarify the use of some terms to avoid any confusion. We will use *game* as in the sentence *"Some games of Dots-and-Boxes are quite difficult to analyse"*. Occasionally we will use

[1]A game that is not impartial is called *partizan*.

[2]The games at level k are said to have *birthday* k.

game to mean a set of rules, as in the sentence *"the game of Dots-and-Boxes is NP-hard"*. The context will suffice to distinguish which is the meaning of *game* in each case. We will use the word *position* in the following sense:

DEFINITION 3. The positions of a game G *are all the games that are reachable from* G *(maybe after several moves; compare with the* options of G*, which are the positions reachable from* G *in* one *move). The positions of* G *also include* G *itself.*

Therefore in general we will use the term *game* when referring to what colloquially we would call "a position", while we will use *position* according to its definition, i.e., a *position of* G is a game that can be reached from the game G. So, given a *game* G, we move to one of its *options*, that is also a *game*, and after several moves from G we reach a *position* of G, which is itself another *game*. In conclusion, we will always use the term *game* unless we would like to point out that the game we are considering is reached from another game G, either in a single move (and we will write *option of* G) or in an indeterminate number of moves (and, in this case, we will write *position of* G).

1.3. Short Games. Even though the number of turns of a combinatorial game must be finite, it may have infinitely many positions.

DEFINITION 4. *A game is* short *if it has finitely many positions.*

Unless otherwise noted, we will only consider short impartial games from now on. There are some examples of short impartial games in the last section of this chapter; maybe the reader will find suitable to have a look at them at this point to get an idea of what kind of games are we considering.

2. Outcome Classes

We will consider games not only from their starting position, but in a broader sense. Therefore, instead of *first player* and *second player*, terms that we will reserve for starting positions, we prefer the expressions *next player* for the player whose turn it is, and *previous player* for the player that has made the last move, without worrying about which move it was (in fact, perhaps no move has yet been made, as in an starting position; in this case by "previous player" we simply mean the second player).

2.1. Winning Strategies. We say that a player has a winning strategy if he can win no matter what his opponent plays.

DEFINITION 5. *We say that the next player has a* winning strategy *in* G *if* $\exists G_1 \in G$ *such that* $\forall G_2 \in G_1\ \exists G_3 \in G_2$ *such that* $\forall G_4 \in G_3\ \exists G_5 \in G_4 \dots$ *where this sequence terminates for some game without options* G_k*, with* k *odd. Analogously, there is a* winning strategy *in* G *for the previous player if* $\forall G_1 \in G\ \exists G_2 \in G_1$ *such that* $\forall G_3 \in G_2\ \exists G_4 \in G_3 \dots$ *where this sequence terminates for some game without options* G_k*, with* k *even.*

So the player who has a winning strategy has at any moment at least a move that ensures that he will make the last move and win no matter what his opponent plays.

We will always assume *perfect play* by both players, which means that they are able to thoroughly analyse the game and will always choose a winning move when available. What this means is that the player that can win will always make a winning move, while his opponent, in fact, can choose his move at random[3], because all of his moves are equally good (or rather "equally bad", since all of them are losing moves).

As we consider that both players play perfectly, we assume that a player that has a winning strategy will win the game. Notice that there is always a winner: the game must end in a finite number of turns, which means that we end up reaching a game without options, and when this happens the player who last played is the winner.

2.2. Outcome Classes.

Definition 6. *Let $\mathcal{N}$ be the set of impartial games where the next player has a winning strategy and $\mathcal{P}$ the set of games where the previous player has a winning strategy. A $\mathcal{P}$-*position *is a game that belongs to $\mathcal{P}$. An $\mathcal{N}$-*position *is a game that belongs to $\mathcal{N}$.*

Proposition 7. *Let G be an impartial game. Then either the next player has a winning strategy or the previous player has a winning strategy. Therefore the set of all impartial games can be partitioned between $\mathcal{P}$ and $\mathcal{N}$.*

Proof. We will prove that a game G belongs to $\mathcal{P}$ iff all its options belong to $\mathcal{N}$. If G has no options, the game is in $\mathcal{P}$: the condition that "all its options belong to $\mathcal{N}$" is true[4]. Otherwise, either all options of G belong to $\mathcal{N}$ or there is some option of $G' \in G$ that belongs to $\mathcal{P}$. In the former case $G \in \mathcal{P}$, because the next player can only make a losing move, while in the latter $G \in \mathcal{N}$ because the next player can win by moving to G', which is a $\mathcal{P}$*-position*. □

So the player who can win an impartial game G will always choose a move to a $\mathcal{P}$*-position* (which will always be available, because $G \in \mathcal{N} \Rightarrow \exists G' \in G$ such that $G' \in \mathcal{P}$), while his opponent will always be forced to move to an $\mathcal{N}$*-position* (because all of the options of $G' \in \mathcal{P}$ are in $\mathcal{N}$). $\mathcal{P}$ and $\mathcal{N}$ are called *outcome classes*[5].

2.3. Isomorphism of Games. By identifying a game by its options we are assuming that two games that have the same set of options are "the same game". For

[3]In real play, against a non-perfect player, he will rather choose a move that makes the position as complicated to analyse as possible, to try and force a mistake, in what is known as *"to give enough rope"*.

[4]We will find that in many proofs by induction the base case is *vacuously satisfied* because we are requiring that some property holds for the elements of an empty set.

[5]In partizan games, where the players are usually called *Left* and *Right*, there are two more outcome classe: $\mathcal{L}$ and $\mathcal{R}$, which correspond to games where a player, respectively Left or Right, can win no matter who starts.

instance, any two games where the next player cannot move are $G = \{\}$ (the set of options is the empty set).

DEFINITION 8. *Two games G and H are* isomorphic games*, $G \equiv H$, if they have identical game trees*[6]*.*

In the case of impartial games, G and H are isomorphic iff they have the same set of options.

We will soon introduce the concept of *equal games*, which are games that, while not being isomorphic, can be considered equivalent in our theory.

3. Sums of Games

Assume that we have two boards of the same game, for instance two chessboards, and that we play in both at the same time in this way: in his turn, a player chooses one of the boards and makes a legal move in that board. What we have is a game consisting of two components G and H, where each one can be considered in itself a game. We will call this situation the *disjoint sum of G and H*, $G + H$. In the disjoint sum of two games a player in his turn can choose to make a move either in G or in H. For instance, if he moves in H to one of its options $H' \in H$, the resulting game will be $G + H'$.

In the case of chess we would have to define how determine the winner (for instance, it could be the first player to checkmate his opponent in one of the two boards), but if we consider a game that follows the *normal play* convention, the first player unable to move in $G + H$ loses, therefore the game ends when there are no possible moves in neither G nor H.

The interest of considering the disjoint sum of two games lies in the fact that many games decompose into disjoint subpositions or components, where each one of them behaves as a game that is independent from the rest (observe that this is the case in all the examples in the last section of this chapter). We will be able to analyse games by analysing its disjoint components.

We define the sum of two games in a recursive way:

DEFINITION 9. *If we have two arbitrary games $G = \{G_1, G_2, \ldots, G_n\}$ and $H = \{H_1, H_2, \ldots, H_m\}$, then $G + H = \{G_1 + H, G_2 + H, \ldots, G_n + H, G + H_1, G + H_2, \ldots, G + H_m\}$.*

This definition corresponds to the idea that the options of $G + H$ are obtained by making a move in one of the components, either G or H, while leaving the other unchanged.

Let us consider the particular case when we have a game whose set of options is empty, $G = \{\}$. Then, for any game H, the available moves in $G + H$ are the same as the available moves in H, and therefore $G + H \equiv H$. So it seems to make sense to consider that any such G is the zero element of the addition, and write $G \equiv 0$.

[6]Some authors write $G \cong H$ instead.

PROPOSITION 10. *The addition of games is well-defined.*

PROOF. We will use a technique called *top-down induction*, which makes some proofs very short. It is a sort of "upside down" induction where we prove that a property of games holds if it is satisfied by the options of any game. The base case usually needs not to be checked, because it corresponds to games without options, $G = \{\}$, and therefore the property is vacuously satisfied. Note that there is no infinite sequence of games $K_1 \in K_2 \in K_3 \in ...$, because a game must finish in a finite number of turns by definition.

In our case it is enough to make two observations. Firstly, the sum of any two impartial games is also an impartial game, because G, H and their respective options are. Secondly, the base cases correspond to sums where at least one game, say G, has no options. In this case the sum of G and H is well-defined because we have $G + H \equiv H$. □

4. Game Equivalence

4.1. Zero Games. In general, we will consider $G + H$ and H to be *equal* not only when the set of options of G is empty.

DEFINITION 11. *A* a zero game[7]*, is any game G such that $G \in \mathcal{P}$. In this case, we will write $G = 0$.*

The reason for that definition is that, if $G \in \mathcal{P}$, for any game H, the outcome class of $G + H$ is the same as the outcome class of H, i.e., $G + H$ and H are either both in $\mathcal{P}$ or both in $\mathcal{N}$. Therefore adding G to H does not change the outcome of H.

PROPOSITION 12. *Let G and H be games. If $G \in \mathcal{P}$, then $G + H$ and H belong to the same outcome class.*

PROOF. If $H \in \mathcal{P}$ then $G + H \in \mathcal{P}$, because the previous player can use the following strategy: each turn reply in the same component where the opponent has just played with a move that would guarantee a win in that component if the whole game were that component alone (we will call such a move *local reply*). This strategy works because both G and H are second player wins, and therefore at some point the first player will be unable to play in both G and H.

On the other hand, if $H \in \mathcal{N}$, then $G + H \in \mathcal{N}$, because the next player can make as first move a winning move in H to some $H' \in H \cap \mathcal{P}$, and proceed as in the preceding case.

□

[7]Compare with *the* zero game, $G \equiv 0$, which means that G has no options.

4.2. Negative of a Game. In combinatorial game theory the negative of a game G is defined as the game $-G$ that has the options of G swapped, that is to say, the possible moves of the players are interchanged (as if they swap colours in a game of chess). In an impartial game this operation has no effect, as each player has the same options as his opponent. Therefore, while in an arbitrary combinatorial game we only have $-(-G) = G$, any impartial game is its own negative, $-G = G$.

4.3. Game Equivalence. Using the concept of negative of a game and that $G = 0 \Leftrightarrow G \in \mathcal{P}$ (Definition 11), we can define game equivalence for arbitrary games.

DEFINITION 13. *Two combinatorial games G and H are* equal[8], *$G = H$, if $G + (-H) = 0$, i.e., if $G + (-H) \in \mathcal{P}$.*

In particular, two impartial games G and H are equal if $G + H = 0$. Observe that any impartial game G verifies $G + G = 0$, because $G + G \in \mathcal{P}$: the previous player can win by replicating his opponent's moves in any component in the other one[9].

PROPOSITION 14. *The equivalence classes of impartial games[10] form an abelian group with respect to the addition, where the zero element is any G such that $G = 0$.*

PROOF. We have already seen closure, that the zero element is any game G without options, and that the inverse of G is itself. The addition is commutative by definition, so all that remains is to prove that we have an abelian group is associativity, and that the equivalence of games as defined is actually an equivalence relation.

We outline the proof of the associativity, $(G+H)+K = G+(H+K)$, by induction on the maximum number of turns playable from $(G + H) + K$ (recall that this number is finite). The base cases are those in which at least one, G, H or K, is zero, where the associativity holds trivially.

The options in $(G+H)+K$ are of the form $(G+H)'+K$ with $(G+H)' \in G+H$ and $(G + H) + K'$ with $K' \in K$. In the former case, either $(G + H)' = G' + H$ with $G' \in G$ or $(G + H)' = G + H'$ with $H' \in H$. Therefore we have three types of options, which verify, by induction hypothesis, $(G' + H) + K = G' + (H + K)$, $(G + H') + K = G + (H' + K)$ and $(G + H) + K' = G + (H + K')$.

Analogously, from $G + (H + K)$ there are three types of options, which correspond to the three types on the right of the previous equalities, respectively, therefore proving that $(G + H) + K = G + (H + K)$.

Now let us prove that game equivalence is an equivalence relation:

- Reflexive: We already proved that $G + (-G) = 0$.
- Symmetric: Assume that $G = H$, so $G + (-H) = 0$. Then $H + (-G) = H + (-G)+0 = (H+(-G))+(G+(-H)) = (G+(-G))+(H+(-H)) = 0+0 = 0$, so $H = G$.

[8] Though this defines an equivalence relation, usually we talk of *equal games* rather than of *equivalent games*.

[9] In fact, G+(-G)=0 for any game G, impartial or not, as long as it verifies the normal ending condition.

[10] In fact, this is true for the equivalence classes of all combinatorial games.

- Transitive: Assume $G = H$ and $H = K$. We have that $G + (-K) = G + (-K) + 0 = G + (-K) + H + (-G) = (G + (-G)) + (H + (-K)) = 0 + 0 = 0$. Therefore $G = K$.

□

4.4. Game Substitution. Using the associativity and commutativity of the addition of games, we can prove a useful property: given a game G, we can substitute any of its components by an equivalent game, and the outcome of G will remain unchanged.

COROLLARY 15. *If $H = H'$ then, for any K, we have $K + H = K + H'$.*

PROOF. (K+H)+(K+H')=(K+K)+(H+H')=0. □

5. Nimbers and Nim

5.1. Nimbers.

DEFINITION 16. *For any $n \in \mathbb{N}$, we define inductively the game $*n$ in this way:*

- $0 = \{\}$*, the game without options.*
- *For $n \geq 1$, $*n = \{0, *1, *2, \cdots *(n-1)\}$.*

The games of the form $*n$ are called nimbers*.*

Therefore from $*1 = \{0\}$ the next player can only move to 0, and he wins because his opponent has no options from there. In the game $*3 = \{0, *1, *2\}$ the next player can move to either the game 0, the game $*1 = \{0\}$ or the game $*2 = \{0, *1\}$. In particular, moving to 0 is the only winning move, because any other option gives his opponent the possibility to move to 0 and win. Therefore $0 \in \mathcal{P}$ and $*n \in \mathcal{N}, \forall n \geq 1$. So the game $*n$ does not seem very interesting. But what if we consider a game obtained by adding nimbers, like $*3+*1$?

We already know that, for any impartial game G, $G+G = 0$. We can prove directly that $*n + *n \in \mathcal{P}$, because any move to some $*m + *n$, where m must be strictly less than n, can be countered replicating the move in the other component so as to obtain $*m + *m$. Therefore $*n + *n = 0$, and we have that any nimber is its own inverse. On the other hand, for $n < m$ we have $*n + *m \in \mathcal{N}$ because an option is to move to $*n + *n = 0$.

5.2. Nim. Now we are going to introduce the game of *Nim*, which, as we will see, has a central role in the theory of impartial games. Nim was analysed by Charles L. Bouton [**Bou**] in 1902, in what could be considered the first paper in Combinatorial Game Theory.

Given several heaps of counters, a move consists in removing as many counters as desired (at least one, up to all) from a single heap. The player that takes the last counter wins (so Nim verifies the normal ending condition). Clearly if we have only one heap, we have the game $*n$, where n is the size of the heap. A game with three heaps can also be seen as three games with a single heap each, therefore a game G with three heaps of size 1, 3 and 5, respectively, is $G = *1 + *3 + *5$. So G is the disjoint sum of the games $*1$, $*3$ and $*5$, which in turn are the components of G.

5.3. Poker Nim. Let us consider a variation of Nim, called *Poker Nim*, in which each player has a (finite) number of counters aside. A move consist in either a standard Nim move, or in adding some of the counters the player has aside to any of the heaps.

PROPOSITION 17. *Let G be a game of Nim. Then the player that can win in G can also win in the same position of Poker Nim, no matter how many counters each player has.*

PROOF. The player with a winning strategy in Nim plays the Poker Nim game exactly as he would do in the Nim game, except when his opponent adds some counters to one of the heaps. In this case, he simply removes the added counters. □

6. Using Nimbers to Solve Impartial Games

6.1. All Impartial Games are Nim Heaps. In the 1930s, Sprague [**Spr**] and Grundy [**Gru**] independently proved that any impartial game can be treated as "equivalent" to a single Nim heap, in a sense we will explain soon, and therefore equivalent to some nimber. This is the main result in the theory of impartial games.

Let us consider games where all options are of the form $*k$, like $G = \{*2, *4, *5\}$.

DEFINITION 18. *The* mex (minimum excluded number) *of a set of non-negative integers $\{n_1, n_2, \ldots, n_k\}$ is the least non-negative integer not contained in the set.*

For instance, $mex\{2, 4, 5\} = 0$ and $mex\{0, 1, 2, 6\} = 3$.

DEFINITION 19. *When a player moves from $*n$ to $*m$ in a component of a game, and his opponent immediately replies by moving in the same component from $*m$ back to $*n$, we say that the move has been* reversed*. Note that this is not always possible, but it is when $n < m$, since then $*n \in *m$. A move that can be reversed is called* reversible move*.*

PROPOSITION 20. *The games $G = \{*n_1, *n_2, \ldots, *n_t\}$ and $*m$, where $m = mex\{n_1, n_2, \ldots, n_t\}$, are equivalent.*

PROOF. The idea is the same as in the proof that Poker Nim is equivalent to Nim (Proposition 17): the equivalence of G and $*m$ holds because G must contain the set of options of $*m$, $\{0, *1, *2, \ldots, *(m-1)\}$, while any additional options of G (if any), must be nimbers $*k$ with $k > m$, which can be reversed by moving back to $*m$.

It is enough to show that $G + *m = 0$, i.e., that $G + *m \in \mathcal{P}$. This is how the previous player wins in the game $G + *m$:

- If the next player moves in component G to any $*k$ such that $k < m$, the previous player replies moving in $*m$ to $*k$, so the resulting game is $*k + *k = 0$.
- If the next player moves in component G to any $*k$ such that $k > m$, the previous player moves in G back to $*m$, so the resulting game is $*m + *m = 0$.
- If the next player moves in component $*m$ to any $*k$, the previous player replies moving in G to $*k$, so the resulting game is $*k + *k = 0$. Notice $*k \in G$ because $m = mex\{n_1, n_2, \ldots, n_t\}$.

In all the cases the previous player can reply moving to some game that is equal to zero. Therefore, given that $0 \in \mathcal{P}$, he wins. □

Now we will show that we can replace options of games by equal games.

PROPOSITION 21. *Let G be any game, let $H \in G$ and let H' be any game such that $H = H'$. Let G' the game obtained by replacing H by H' in G. Then $G = G'$.*

PROOF. We need to prove that $G + G' \in \mathcal{P}$. If the first player moves in G to some position other than H, or in G' to some position other than H', the second player can reply moving in the other component to the same position, and the sum of the two components will be zero. On the other hand, if the first player moves in G to H, or in G' to H', the second player can reply making the other of the two moves, obtaining $H + H' = 0$. □

COROLLARY 22. *If $G = \{G_1, G_2, \ldots, G_n\}$, and $\forall i$ $G_i = *k_i$, then $G = *m$, where $m = mex\{k_1, k_2, \ldots, k_n\}$.*

PROOF. $G = \{G_1, G_2, \ldots, G_n\} = \{*k_1, *k_2, \ldots, *k_n\} = *m$. □

We are ready to prove the Sprague-Grundy Theorem.

THEOREM 23 (Sprague-Grundy). *If G is a short impartial game, then $G = *m$ for some m.*

PROOF. The options of G are impartial games. By induction hypothesis each option is equal to a nimber. There are finitely many options, because G is short, so $G = \{*n_1, *n_2, \ldots, *n_t\}$, which implies that $G = *m$, where $m = mex\{n_1, n_2, \ldots, n_t\}$. As the game must finish in a finite number of turns, when one of the players cannot move, the base case is the zero game, $G = \{\} = 0$. □

Therefore an impartial game can be solved by recursively by finding to which nimber is equivalent each of its options. This does not mean that solving an impartial game is easy, because finding those nimbers can be really challenging.

As a corollary of the Sprague-Grundy Theorem, we have a result we already know.

COROLLARY 24. *Let G be an impartial game. Then $G = 0$ iff $G \in \mathcal{P}$.*

PROOF. If $G = 0$, Previous wins by reversing Next's moves until reaching $0 = \{\}$, the game without options.

If $G \neq 0$, by Theorem 23 there is some $m \neq 0$ such that $G = *m$. Next wins by moving to 0 (as $0 \in *m$), and then reversing any of his opponent's moves. $\square$

6.2. The Grundy Function. Now that we know that any impartial game is equivalent to some $*n$ we can describe an impartial game either by its set of options or by its equivalent nimber: $G = \{0, *1, *4\} = *2$. Note that $*2$ is a class of equivalent games, the canonical representant of which is $\{0, *1\}$, given that $*4$ is a reversible option. In general, the *canonical representant* of $*n$ is $\{0, *1, *2, \cdots *(n-1)\}$.

Observe that we have a mapping $\mathcal{G}$ of the set of impartial games into the non-negative integers, defined in this way:

$$\mathcal{G}(G) = n \Leftrightarrow G = *n$$

If $\mathcal{G}(G) = n$, we say that n is the *nim-value* of G.

Therefore, given an impartial game G, the assertions "G is equal to $*n$" and "G has nim-value n" are equivalent. We call $\mathcal{G}$ *Grundy function*. We have that $\mathcal{G}(G)$ can be computed from the options of G, as

$$\mathcal{G}(G) = \underset{G' \in G}{mex}\, \mathcal{G}(G')$$

Note: There is an order relation usually considered in general combinatorial games, which we do not introduce because it is not needed in this thesis. It is a partial relation in which all nimbers, i.e., all impartial games, are incomparable (see [**Ber01**] or [**Alb**]). Therefore, if we have games $G = *3$ and $G' = *2$, we will not write $G > G'$, but we can write $\mathcal{G}(G) > \mathcal{G}(G')$ instead, since $\mathcal{G}(G) = 3$ and $\mathcal{G}(G') = 2$.

6.3. Adding Nimbers. If we have a game consisting of two nim heaps of n and m counters respectively, by the Sprague-Grundy Theorem this game must be equal to some $*k$. To find such k we define the nim-sum of non-negative integers.

DEFINITION 25. *The* nim-sum *of $n, m \in \mathbb{N}$, $n \oplus m$, is the decimal expression of the XOR logical operation (exclusive OR) applied to the binary expressions of n and m.*

For instance, $5 \oplus 3 = 6$, because 101 XOR 011 = 110. We will need some properties that are derived from the definition:

LEMMA 26.
- *Nim-sum is commutative and associative.*
- *For any a, b, we have that $a \oplus b = 0 \Leftrightarrow a = b$.*
- *For any $a' < a$ and b, we have that $a' \oplus b \neq a \oplus b$.*

The following proposition proves that the nim-sum holds the answer we are looking for:

PROPOSITION 27. *If $n \oplus m = k$, then $*n + *m = *k$.*

Proof. By the lemma, we have that $a \oplus b = 0 \Leftrightarrow a = b$. We also know that $*a + *b = 0 \Leftrightarrow a = b$. Therefore the proposition can be restated as: "if $n \oplus m \oplus k = 0$ then $*n + *m + *k = 0$".

We will use induction on $n+m+k$. Base case is for $n = m = 0$. For the general case assume, w.l.o.g., that the next player moves in the first heap to some $*n' + *m + *k$, where $n' < n$. We will prove that there is a move for his opponent to a $\mathcal{P}$-*position*, which that proves $*n + *m + *k \in \mathcal{P}$, i.e., $*n + *m + *k = 0$.

Let $*j = *n' + *m + *k$. Observe that $*j \neq 0$ by the previous lemma and the induction hypothesis:

$$*j = *n' + *m + *k = n' \oplus m \oplus k \neq n \oplus m \oplus k = 0$$

Consider the binary expressions of all those numbers. Take the most significant digit of j that is a 1. Either n', m or k have a 1 in that position and 0 in its most significant digits (if any); assume w.l.o.g it is m. Obtain m' by changing in m the aforementioned digit and also any digit whose digit in the same position of j is a 1 (either from 0 to 1, or from 1 to 0, depending on its value). By construction $m' < m$ and $n' \oplus m' \oplus k = 0$. By induction hypothesis $*n' + *m' + *k \in \mathcal{P}$. □

Continuing with our example, we had $5 \oplus 3 = 6$, so $*5 + *3 + *6$ is a $\mathcal{P}$-*position*. If, for instance, the next player moves in the last component to $*5 + *3 + *2$, we have

$5 = 101_2$

$3 = 011_2$

$2 = 010_2$

$5 \oplus 3 \oplus 2 = 4 = 100_2$

So the opponent must only change the most significant 1 in 5, obtaining $1 = 001_2$. The resulting position is $*1 + *3 + *2$, which turns out to be a $\mathcal{P}$-*position* (by induction) because $1 \oplus 3 \oplus 2 = 0$.

Proposition 27 leads to an important result:

Corollary 28. *For any two games G and H, we have that $\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$.*

6.4. How to Win at Nim. So how do we win at Nim? We have to move to a $\mathcal{P}$-*position*, i.e., some position whose nim-sum is zero. Obviously if we already are in a $\mathcal{P}$-*position* we cannot win, at least against a perfect opponent. The $\mathcal{P}$-*positions* can be found recursively:

- $\emptyset \in \mathcal{P}$.
- All games with one heap belong to $\mathcal{N}$, because the first player can move to 0 by taking all the counters (i.e., $\forall n, 0 \in *n$).
- The games with two heaps belong to $\mathcal{P}$ iff the two have the same size, because we have proven that $*n + *m = 0$ iff $n = m$.

- Let us prove that a game with three heaps belongs to $\mathcal{P}$ iff is of the form $\{n, m, n \oplus m\}$: we know that $\{n, m, k\}$ belongs to $\mathcal{P}$ iff $*n + *m + *k = 0$. Using the properties of impartial games, we have

$$*n+*m+*k = 0 \iff (*n+*m+*k)+*k = 0+*k \iff *n+*m+(*k+*k) = *k \iff *n+*m = *k$$

  which is equivalent to $n \oplus m = k$.
  The $\mathcal{P}$-*positions* with three heaps of 6 or less counters each are $\{1, 2, 3\}$, $\{1, 4, 5\}$, $\{2, 4, 6\}$ and $\{3, 5, 6\}$. In all these cases the nim-sum of the three numbers is zero, or, what is the same, the nim-sum of any two numbers in each triplet gives as result the third one. An example of $\mathcal{N}$-*position* is $\{2, 3, 5\}$. We have $2 \oplus 3 \oplus 5 = 4$, therefore the game $*2 + *3 + *5$ is equivalent to $*4 \in \mathcal{N}$, and the next player can win by moving to a zero nim-sum position, which can only be achieved by removing four counters from the heap of five, as to obtain $*2 + *3 + *1 = 0$.

6.5. How to Win at any Impartial Game. To determine who wins at any given impartial game G, we must compute $\mathcal{G}(G)$. If $\mathcal{G}(G) = 0$, then $G \in \mathcal{P}$ and the previous player wins. Otherwise the next player wins by moving to some $G' \in G$ with $G' = 0$ (which exist because $G \neq 0$).

Notice that if $G = G_1 + G_2 + \cdots + G_n$ has several disjoint components and we know the nim-value of each one, we can compute the nim-value of G using

$$\mathcal{G}(G_1 + G_2 + \cdots + G_n) = \mathcal{G}(G_1) \oplus \mathcal{G}(G_2) \oplus \cdots \oplus \mathcal{G}(G_n)$$

In conclusion, to solve an impartial game it is enough to find the nim-value of its disjoint components.

7. Examples of Impartial Games

7.1. Kayles. Kayles is a game by H.E. Dudeney played with a row of bowling pins. A turn consists in (virtually) throwing a ball that strikes either one or two adjacent pins. Notice that a move can separate a row into two unconnected rows. Let K_n be the nim-value of a row of n pins. The options of this game are $K_r + K_s$ with $r+s \in \{n-1, n-2\}, r, s \geq 0$. Therefore we can compute the values recursively, as in Figure 1.

The nim-values of K_n are 12-periodic for $n > 70$.

7.2. Cram. A classic game in Combinatorial Game Theory is *Domineering*. Played on a rectangular board, each player places a domino in his turn, covering exactly two adjacent squares. One of the players plays the dominoes horizontally and the other vertically, so it is a partizan game. The impartial version of Domineering, where both players can place the dominoes horizontally as well as vertically is called *Cram*.

By making symmetric replies, the second player can win in nxm boards for n and m even, so the nim-value of these games is 0. Analogously, when n is even and m

$K_0 = 0$
$K_1 = \{K_0\} = \{0\} = *1$
$K_2 = \{K_0, K_1\} = \{0, *1\} = *2$
$K_3 = \{K_1, K_2, K_1 + K_1\} = \{*1, *2, 0\} = *3$
$K_4 = \{K_2, K_3, K_1 + K_1, K_1 + K_2\} = \{*2, *3, 0\} = *1$
$K_5 = \{K_3, K_4, K_1 + K_2, K_1 + K_3, K_2 + K_2\} = \{*3, *1, *2, 0\} = *4$

FIG. 1. Recursive computation of the first values of the Kayles game K_n.

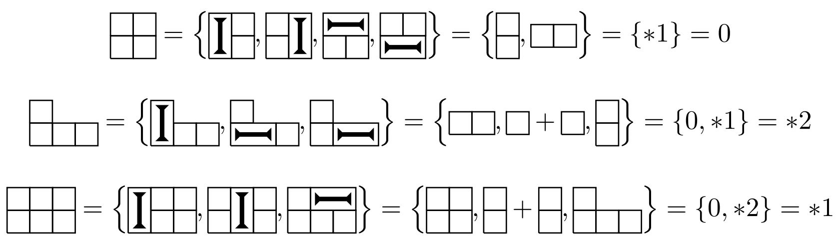

FIG. 2. Cram: Computing the nim-value of the 2x3 rectangle.

odd the first player can win by making the first move in the two central squares, and then replying symmetrically. The tricky case is when both n and m are odd.

We show how to recursively compute the nim-value of the 2x3 rectangle in Figure 2.

7.3. Chomp. Chomp is a game by David Gale played on a rectangular grid (or *chocolate bar*). In his turn, each player selects a square and eliminates all the squares that are neither to its left nor below it. The player that takes the bottom left corner square (the *poisonous square*) loses. At first it may seem that Chomp does not follow the normal play convention, but it is enough that we forbid that a player takes the poisonous square to circumvent this problem.

PROPOSITION 29. *The first player wins Chomp in any rectangular grid other than* 1x1.

PROOF. The non-constructive *strategy stealing* argument is a technique often used in Combinatorial Game Theory:

Suppose the second player has a winning strategy. Consider that the first player starts *eating* the upper right corner. By hypothesis, the second player has a winning move. But then the first player could have made this same move before and win!

Unfortunately this argument does not tell us which move is a winning move for the first player. □

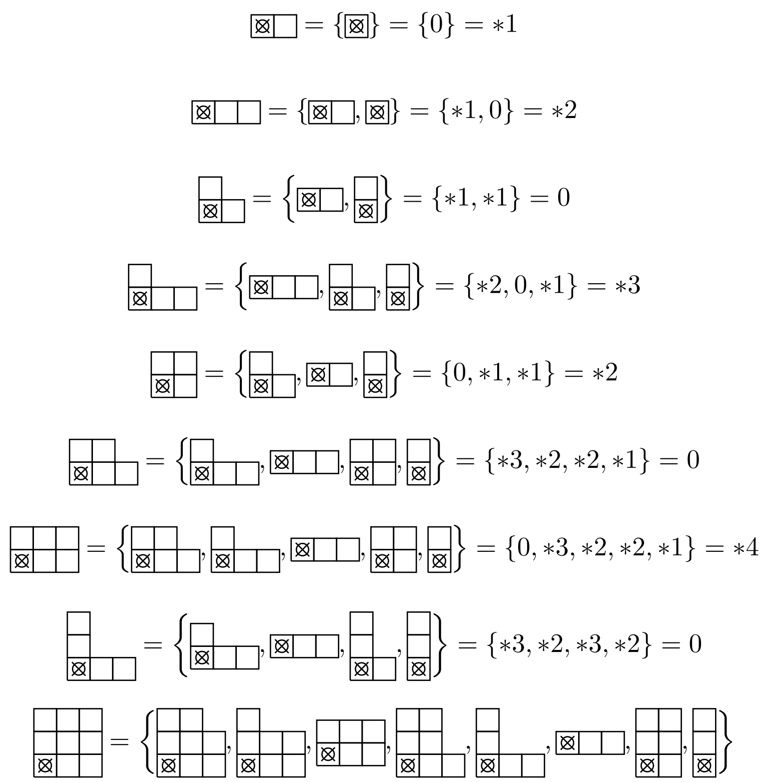

Fig. 3. Computing the nim-value of some Chomp positions.

We show how to compute the nim-values of some games in Figure 3. For clarity we indicate the nim-values of each position in the same order that they appear, even though then we have to write some of the nim-values more than once.

Observe that in the 3x3 square we only show its options, without computing its value. But we do not need it if all we want is to determine the outcome. Recall that $G \in \mathcal{N}$ iff there is a position $G' \in G$ such that $G' = 0$. Since the L-shaped position has nim-value 0, this is a winning move, and so the game is an $\mathcal{N}$-*position* .

Chapter 2
Dots-and-Boxes and Nimstring

General Combinatorial Game Theory cannot be directly applied to Dots-and-Boxes because it is a scoring game, which means that the winner is not necessarily the player who makes the last move. If we just replace the winning condition by the normal ending condition then we obtain a game that we can analyze with this theory and is, in fact, very related to the original game. Therefore to fully understand Dots-and-Boxes we must know how to win at another combinatorial game, called Nimstring. In this chapter we introduce the rules of some related games, briefly show how a game of Dots-and-Boxes is played (mainly focusing on the endgame), and finally motivate the study of Nimstring by showing its relation with Dots-and-Boxes.

1. Game Rules

1.1. Dots-and-Boxes. The rules of Dots-and-Boxes are very simple:

i) Start with a rectangular dots grid.
ii) On his turn each player draws a horizontal or vertical segment joining two (not previously joined) adjacent dots.
iii) If a player draws the fourth side of a square ("box") then he claims it and must make another move. Notice that several captures can be made in a single turn.
iv) When all the boxes have been claimed, the player who owns the most is the winner.

Observe that this game verifies the standard conditions of combinatorial games as defined in Chapter 1 except for the normal play condition.

1.2. Strings-and-Coins. Consider the game defined by the following rules:

i) Start with a graph $G = (V, E)$, with set of vertices V and set of edges E, and a distinguished vertex, $g \in V$.
ii) On his turn each player removes an edge.

iii) When a player removes the last edge incident to a vertex other than g he claims the vertex and must remove another edge. Notice that several vertices can be claimed in this way in a single turn.
iv) When there are no more edges, the player who claimed the most vertices is the winner.

This game is called *Strings-and-Coins*, because it is described considering that the vertices are coins, the special uncapturable vertex g is called *the ground*, and the edges are strings attached to the coins that you have to cut[1]. The player that cuts the last edge attached to a coin claims it. Note that the special uncapturable node g plays the role of the "exterior face". It corresponds to the artificial, imaginary node that is the other end of all the edges that are drawn incident to only one vertex (Figure 5 provides an example; observe that g is not drawn as a vertex). A way to circumvent having to consider this special node g would be, as suggested in [**Hea09**], to substitute the edges that go from a vertex to the ground by loop edges in that vertex.

Dots-and-Boxes is a particular case of Strings-and-Coins, considering the graph $G = (V, E)$ of a grid, whose set of vertices is $V = \{v_1, v_2, \cdots v_k, g\}$, which correspond, with the exception of g, to the set of boxes $\{b_1, b_2, \cdots b_k\}$, with edges $\{v_i, v_j\} \in E$ iff the boxes b_i and b_j share an (unplayed) edge, and $\{v_i, g\} \in E$ iff the box b_i has an (unplayed) edge in the perimeter of the grid. Note that Strings-and-Coins is more general than Dots-and-Boxes: for instance, the number of edges incident to a coin is not restricted to a maximum of 4, nor has the graph to be planar.

Sometimes it is clearer to look at a Dots-and-Boxes game by looking at his Strings-and-Coins counterpart. An example of a Dots-and-Boxes game and its dual Nimstring game is shown in Figures 4 and 5.

1.3. Nimstring. We obtain the game called *Nimstring* if we modify the winning condition of Strings-and-Coins. Nimstring is played exactly as Strings-and-Coins, except that it follows the normal play condition: the first player who is unable to move loses. In particular, the number of coins captured by each player is irrelevant.

Although Nimstring, as Strings-and-Coins, can be played on arbitrary graphs, we will only consider Nimstring games whose graph is the dual of a Dots-and-Boxes game (in particular, planar graphs, with maximum degree 4 -except for g-, no cycles of odd length, etc.).

2. Playing Dots-and-Boxes

Let us see how a typical Dots-and-Boxes game can be played.

Assume that both players avoid to draw the third edge of a box while possible, so that the opponent cannot capture any boxes. In this case we reach an endgame

[1]We reserve the word "string" for a different object that we will introduce in the next chapter, so we will always use the word "edge", as in Dots-and-Boxes. We will also refrain from using "coin" and will use "box", "node" or "vertex" instead.

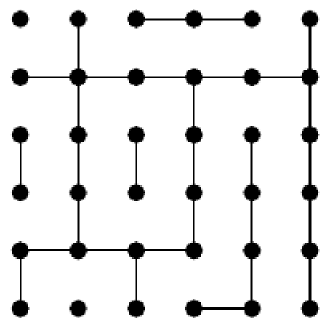

FIG. 4. A Dots-and-Boxes endgame where any move will concede a box to the opponent.

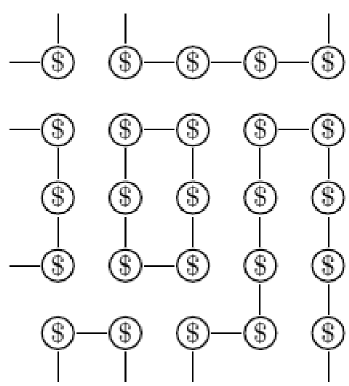

FIG. 5. The dual Strings-and-Coins position of the game in Figure 4.

position like that of Figure 4. Any move from this position will offer some boxes to the opponent.

DEFINITION 30. *a) A* independent k-chain *is a component consisting on a cycle of length k that includes vertex g.*

b) An independent l-loop *is a connected component consisting on a cycle of length l that does not include vertex g.*

c) A simple endgame *is a game where each component is either an independent chain or an independent loop.*

The game shown in Figure 4 is a simple endgame: its only components are an independent 6-loop and several independent chains, as you can also see in its equivalent Nimstring form in Figure 5.

We will only consider simple endgames in this chapter.

2.1. Keeping Control in the Endgame.

Definition 31. *Chains of length 1 or 2 are called* short chains*, while chains of length* $l \geq 3$ *are called* long chains*, or simply* chains.

Long chains are much more relevant in the game than short chains. This is the reason why when we write *chains* we will assume that we are referring exclusively to long chains.

Definition 32. *a) A* loony move *is a move that offers at least 3 boxes. We will use the symbol* $\mathfrak{D}$ *to denote a loony option of a game.*

b) A loony endgame *is a game where all the available options are loony moves.*

c) The player in control *is the player who last played before reaching the loony endgame.*

d) The strategy consisting in declining the last two boxes of each chain and the last four of each loop is called keeping control.

Let us see how the player in control plays the game game on Figure 6 (top) in order to minimize the number of boxes captured of his opponent. Note that it corresponds to the loony endgame obtained from the game in Figure 4 once the two short chains have been claimed. When a (long) chain is offered, the player in control will not take all the boxes but will decline the last two instead, as in Figure 6 (bottom). In this way he forces his opponent to play on ("offer") another chain or loop. The player in control will play in an analogous way when offered a loop, but in this case he will decline 4 boxes, as in Figure 7. Can he win using this strategy?

Proposition 33. *Let* G *be a loony endgame with* n *chains and* m *loops, for a total of* b *unclaimed boxes. Suppose that the net score (number of boxes claimed minus number of boxes claimed by opponent) for the player in control at this point is* s*. If* $b + s > 4n + 8m$ *then the player in control wins the game.*

Proof. Keeping control guarantees the player in control all the boxes except $2n + 4m$, that is, $b - 2n - 4m$ boxes, while his opponent captures $2n + 4m$. Therefore he obtains a net gain of $b - 2n - 4m - (2n + 4m) = b - 4n - 8m$ boxes in the loony endgame, which means that a sufficient condition for him to win is $b + s > 4n + 8m$. □

For instance, in Figure 6 (top) player A is in control. Before any move we have $b = 22$, $s = -1$, $n = 3$, $m = 1$, so $b + s = 21$ and $4n + 8m = 20$. Therefore A wins by 1. Observe that we do not claim that this strategy is the best possible; in fact, in the example, A can win by more than 1 box because he does not need to decline any boxes in the last component he is offered.

Observe that in each chain of length ≥ 5 and on each loop of length ≥ 10 the player in control obtains a positive net gain, while only on 3–*chains*, 4–*loops* and 6–*loops* he gives more boxes away than he takes. So if all the chains and loops in the loony endgame are "long enough", or there are "long enough" chains and loops to compensate for the "not-so-long" chains and loops, this strategy will suffice to win the game.

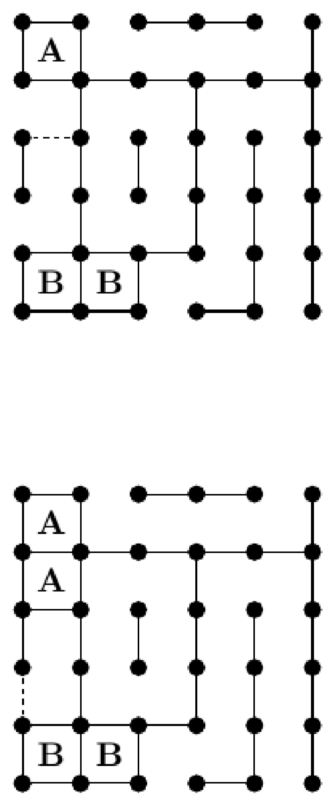

FIG. 6. a) In this loony endgame, after taking the two boxes of the 2-chain at the bottom, Player B has to offer a (long) chain or a loop, and choses to offer the 3-chain on the left.
b) Player A, who is in control, only takes one box of the 3-chain.

2.2. Why try to reach the loony endgame being the second to play. We have just shown a strategy that will be a winning one in many games. But what happens when it is not? We are going to show that you can probably win anyway, provided that you force your opponent to be the first to play in the loony endgame. That is because keeping control is not the only possible strategy.

THEOREM 34. *In a loony endgame G the player in control will, at least, claim half the remaining uncaptured boxes.*

PROOF. By definition of loony endgame, the first move in G must be a loony move, i.e., a move offering either a long chain or a loop.

Assume it is a k-chain. Let us call G' the rest of the game, i.e., G minus the k-chain. Let x be the net gain of the player who plays second in G'. When offered the k-chain, we will consider two possible strategies for the player in control: either take all the boxes, or decline the last two. If he claims all boxes in the chain, he

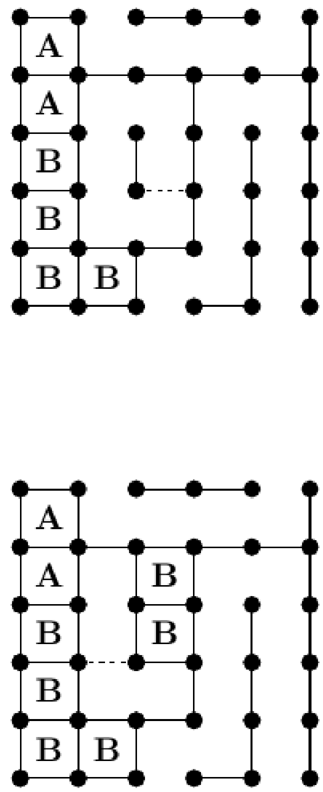

FIG. 7. Player B offers a loop, then Player A declines the last four boxes.

will play first in G', obtaining a total net gain of $k - x$ (because his opponent will get a net gain of x in G', as he will play second there). If he declines the last two boxes in the chain, his opponent will play first in G', so the player in control will obtain a total net gain of $k-4+x$ (because his net gain in the chain is $k-4$, since he claims $k-2$ boxes and his opponent 2). If he chooses the strategy of those two that guarantees him a higher net gain, this net gain is $k-2+|x-2|$, as he will choose the first strategy if $x \leq 2$ and the second one if $x \geq 2$. Since $k > 2$, this net gain is positive, which means that he will claim more than half the uncaptured boxes.

The same applies when the player in control is offered an l-loop, only that in this case his decision depends on the sign of $x-4$, and his net gain will be $l-4+|x-4|$. This net gain is non-negative because $l \geq 4$, which implies that the player in control will claim, at least, half the remaining boxes. □

We will prove in chapter 4 (Proposition 56) that the optimal strategy when offered a chain (resp., loop) is, in fact, one of the two strategies considered in the proof: either to claim all boxes, or to take all but two (resp. four).

The conclusion is that forcing your opponent to play first on the loony endgame will often give you the victory. While this is not the case in all games, the fight is always on "obtaining control", and only when the player in control has made too many sacrifices on the road, while not succeeding on obtaining an enough big net gain in the loony endgame, he will lose.

Although we can have far more complicated loony endgames than the one in Figure 6 if they are not simple (as we will see in the next chapter), the idea is the same; at this point our only intention is to motivate the study of Nimstring showing its relation with Dots-and-Boxes.

3. Playing Nimstring to Win at Dots-and-Boxes

In this section we prove that in Nimstring, forcing your opponent to play first in the loony endgame is even better than in Dots-and-Boxes: it always guarantees a win. Besides, if a player has a winning strategy in a game of Nimstring, and he uses that same strategy in the dual Dots-and-Boxes game, he will be the player in control.

We end the section with an example of how to compute the nim-value of a game.

3.1. Loony moves in Nimstring. We will use the concepts of *loony endgame* and *loony move* in the same sense as in Dots-and-Boxes, so we do not need to redefine them. Therefore a *loony endgame* is a game in which any move offers a box and there are no short chains. Any move in such a position, which implies offering a (long) chain or a loop, is a *loony move*. We will prove that the first player to make such a move will lose the game.

First of all we are going to show that there is a kind of move on a short chain that is also a losing move.

Let us consider a game that includes a short chain of length 2. The are two possible moves in the 2-chain:

i) Play the edge that separates the two boxes (*hard-hearted handout*), as in Figure 8.
ii) Play another edge (*half-hearted handout*), as in Figure 9.

LEMMA 35. *A half-hearted handout is always a losing move.*

PROOF. Consider a game G containing a 2-chain. Let H be the rest of the game. Assume Player A plays a half-hearted handout on the 2-chain. We have to prove that the other player can win in G. Consider for the moment H as a game alone. Either $H \in \mathcal{N}$ or $H \in \mathcal{P}$. In the first case, the winning strategy in G for Player B is to take the two boxes of the 2-chain and move in H so as to win the game. In the second case, the winning strategy is to decline the 2 boxes (as in Figure 9), forcing the opponent to move first in H.

□

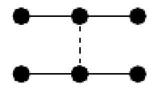

FIG. 8. A hard-hearted handout.

FIG. 9. A half-hearted handout by A, answered by B by declining the two boxes, forces A to move first in the rest of the game.

Therefore we can assume that when a player is offering a 2-chain he always plays the edge that separates the two boxes.

PROPOSITION 36. *A loony move is a losing move.*

PROOF. When offered a chain the player in control takes all the boxes except the last two. At this point he has to play again, and the situation is the same as if his opponent had just made a half-hearted handout in a 2-chain, which, by the lemma, means that the player in control wins the game.

The proof that a move that offers a loop is also a losing move is analogous to the proof of the lemma: the player in control can take all the boxes and move first in the rest of the game H, or take all the boxes but the last four and force his opponent to move first in H. One of the two choices must be a winning move, depending on the class of H, either a $\mathcal{P}$*-position* or an $\mathcal{N}$*-position* . □

Given that in a loony endgame the next player must make a loony move, we have the following result:

COROLLARY 37. *If G is a loony endgame, $G \in \mathcal{P}$.*

In the bibliography, a half-hearted handout is also considered a loony move (being a move which allows the opponent to choose who moves next in the rest of the game), but we did not include it as such to simplify the definition. This has no effect, because we can assume that perfect players will always offer 2-chains making hard-hearted handouts instead. By our definition, offering a short chain is never a loony move. This explains why we distinguish between short and long chains. This distinction could also be justified by the fact that when your are offered a chain you can decline the last two boxes only if the chain is long (note you could also decline the last two boxes after a half-hearted handout, which we will assume will never happen).

3.2. Ignoring loony moves. As a loony move is a losing move, no player will make a loony move unless no other possibility is available, and we can consider that loony moves are not available options. If a loony move is a losing move, why play it? What we mean is that we can remove loony moves from the options of a game without changing the outcome.

PROPOSITION 38. *Let* $G = G' \cup \{\mathfrak{D}\}$ *be any game. Then* $G = G'$.

PROOF. When the next player makes a move in one component of $G + G'$, the previous player can replicate that move in the other component. The only exception is if the next player plays the loony move in G; in this case the previous player takes all the boxes except the last 2 (if offered a chain) or the last 4 (if offered a loop). With this strategy, Previous wins in $G + G'$, so $G + G' = 0$. □

Therefore when computing the *mex* of a multiset we just ignore loony moves.

For this reason we can consider that $\mathfrak{D}$ is a sort of ∞, given that we can informally write that $mex(S) \cup \infty = mex(S)$ for any set S of non-negative integers. (Besides, when adding games, we can consider that $G + \mathfrak{D} = \mathfrak{D}$ for any G, given that playing a loony move in any component affects the whole game.)

3.3. Computing nim-values. Let us see how do we compute nim-values with some examples. The nim-value of a game G consisting only of a chain or a loop is 0 (because all its options are loony moves), which is consistent with the fact that G is a $\mathcal{P}$*-position*.

In Figure 10, for example, we compute the nim-value of the figure on the top by considering their options and finding their nim-values recursively (so we compute them from the bottom upwards). The nim-value of each game in the figure is determined by finding the *mex* of its options. The number in each edge indicates the nim-value of the option obtained by playing that edge.

4. Outline of Expert Play at Dots-and-Boxes

We have just shown that the first player to move in the loony endgame will lose the Nimstring game. In Dots-and-Boxes that player is likely to lose too, as we already discussed. So, when playing Dots-and-Boxes, advanced players try to win at the corresponding Nimstring game, which is equivalent to being the player in control in the loony endgame. Sometimes they have to sacrifice some boxes to achieve it. Except on small boards (where not many chains and loops are produced, and they are not very long) the winner of the Nimstring game is often the player who wins the Dots-and-Boxes game. This is because he is guaranteed, as we proved, at least half the boxes in the loony endgame, but often many more. The net gain in the loony endgame in favour of the player in control is decisive: he can only lose when his opponent had forced him to make enough sacrifices to overcome that net gain.

4.1. Sacrifices. As an illustration of how a sacrifice of some boxes can lead to winning, we consider the game in Figure 11 (top). Imagine that the next player (B) makes the move shown by a dashed line, which sacrifices two boxes. Why would he do so? Let us continue with the game to see that, in fact, this is a winning move.

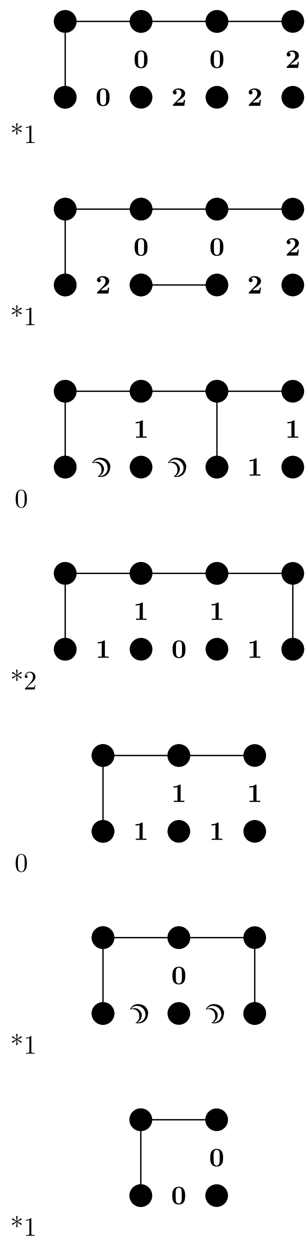

Fig. 10. Computing the nim-value of a game.

Player A will take the two offered boxes and play the only move that does not offer any boxes: the middle edge in the loop (Figure 11, middle). After the sacrifice we reach a phase where the players offer the short chains available (this is not always the case; in many games sacrifices are played earlier). In this exchange Player B will claim 2 boxes and Player A will claim 3.

Then we reach the loony endgame (Figure 11, bottom). Player A will offer the loop and the two chains. Supose he offers the 9-chain last. Player B will take all boxes, with the exception of the last four of the loop and the last two in the 3-chain. He does not need to decline the last boxes in the 9-chain because it is the last component of the game.

To sum up, Player B entered the loony endgame trailing 2 boxes to 5, due to his sacrifice and the exchange of short chains. However, thanks to being the player in control and keeping it until the end, he obtains a score of 12-6 in his favour in the loony endgame, winning the game 14-11. (The reader can check that offering the 3-chain last leads to the same score, while offering the loop last would be worse for Player A, because Player B would not need to decline the last four boxes of the loop and would win by a broader margin, 16-9.)

Had Player B not made the sacrifice, playing the safe move in the centre of the loop, Player A would have created a third chain in the north, leading to the endgame in Figure 4, which we saw in subsection 2.1 was a win by Player A. In fact, Player B played the only winning move. How did he know that he had to make the sacrifice? This question will be answered in the next chapter, but we can advance that Player B needed the number of chains to be even in order to win, and that his sacrifice was aimed at preventing the creation of a third chain.

4.2. Short chains' exchange. In the example of the previous section, the player in control obtained less boxes than his opponent in the phase where the short chains were offered. We end this section showing that the player in control cannot have a positive net gain in this exchange, but his opponent can obtain, at most, an advantage of two boxes.

PROPOSITION 39. *Consider a game where all options that are not loony moves offer short chains. Assume that the short chains will be offered before making any loony move. Let x and y the number of boxes that the player in control and his opponent, respectively, will capture from short chains. Then $0 \leq y - x \leq 2$.*

PROOF. Players will offer 1-chains before 2-chains. Let $2 \geq l_1 \geq l_2 \geq \cdots \geq l_k \geq 1$ be the lengths of the short chains in the reverse order in which they are offered. Note that the player in control must be the one who offers the last short chain (l_1), while who offers the first (l_k) depends on the parity of k: either the player in control if k is odd (so he will end up offering one more chain than he will be offered), or his opponent if k is even (because both players will offer the same number of short chains).

Therefore if k is even we have that

$$x = \sum_{\substack{j=2 \\ j \text{ even}}}^{k} l_j \qquad \text{and} \qquad y = \sum_{\substack{j=1 \\ j \text{ odd}}}^{k-1} l_j \quad ,$$

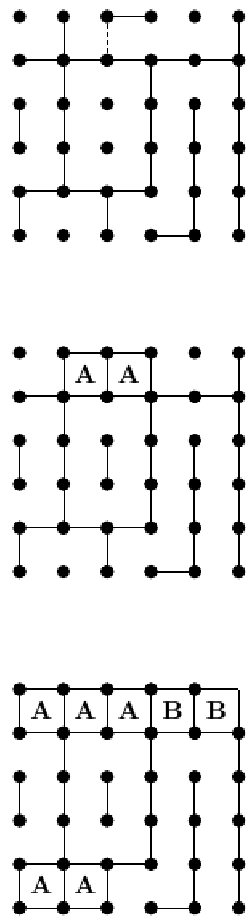

FIG. 11. Player B sacrifices two boxes in order to ensure that he is the player in control in the loony endgame.

while if k is odd then

$$x = \sum_{\substack{j=2 \\ j\ even}}^{k-1} l_j \qquad \text{and} \qquad y = \sum_{\substack{j=1 \\ j\ odd}}^{k} l_j \quad .$$

In particular, if all the short chains are of the same length l, then $x = y = lk/2$ if k is even, and $x = l(k-1)/2$ and $y = l(k+1)/2$ (so $y - x = l$) if k is odd.

If not all the short chains have the same length, then there is a unique i such that $l_i = 2$ and $l_{i+1} = 1$. We have four cases, depending on the parities of k and i. We claim that:

i) If k and i are even, then $y = x$.
ii) If k is even and i is odd, then $y = x + 1$.

iii) If k is odd and i is even, then $y = x + 1$.
iv) If k and i are odd, then $y = x + 2$.

Let us prove the k odd, i even case; the other cases are analogous:

$$x = \sum_{\substack{j=2 \\ j\ even}}^{i} l_j + \sum_{\substack{j=i+2 \\ j\ even}}^{k-1} l_j = \sum_{\substack{j=2 \\ j\ even}}^{i} 2 + \sum_{\substack{j=i+2 \\ j\ even}}^{k-1} 1 = 2 \cdot \frac{i}{2} + \frac{k-i-1}{2} = \frac{k+i-1}{2}$$

$$y = \sum_{\substack{j=1 \\ j\ odd}}^{i-1} l_j + \sum_{\substack{j=i+1 \\ j\ odd}}^{k} l_j = \sum_{\substack{j=1 \\ j\ odd}}^{i-1} 2 + \sum_{\substack{j=i+1 \\ j\ odd}}^{k} 1 = 2 \cdot \frac{i}{2} + \frac{k-i+1}{2} = \frac{k+i+1}{2}$$

So, in this case, $x + 1 = y$.

□

Chapter 3
Winning at Nimstring

As in any impartial combinatorial game, we can win a Nimstring game by moving to a $\mathcal{P}$*-position* if available. However this implies computing the nim-values of the options of the game (to find an option equal to 0, or determine that there is none), which can be extremely hard in some cases. Now we know that an alternative is to play in such a way as to force our opponent to be the first to play in the loony endgame. But how do we have to play in order to achieve this goal? As we will see, the key to answer that question has to do with the number of moves that will be played on *strings*, a concept that generalises the notion of independent chain to non-simple loony endgames.

1. Non-simple Loony Endgames

In Chapter 2 we only considered simple loony endgames, i.e., loony endgames where all components are independent chains or loops. Before analysing Nimstring further we need to have a look at arbitrary loony games.

1.1. Strings. In general, given a loony endgame, when the player in control is offered some boxes by his opponent which do not form an independent loop, the player in control will be always able to take all the offered boxes except the last two, as in an independent chain.

Before proving this assertion we need some definitions:

Definition 40. *Given a game of Nimstring G,*

- *A* joint *is a node of degree ≥ 3. We use j to indicate the number of joints of a game.*
- *A* stop *is either a joint or the ground g.*
- *A* string *is a path which goes from a stop to the same or to another stop, such that its internal nodes have degree 2 with respect to G. The stops at the ends are not part of the string.*
- *The* total valence *of the game, v, is the sum of the degrees of all the stops of G.*

In a loony endgame all nodes have degree ≥ 2 (given that nodes of degree 1 are capturable). In fact, most nodes will have degree 2. Notice that we do not take into account these nodes of degree 2 when computing v.

Since when offered a string a player can always decline the last two boxes as it happens in an independent chain, the concept of string seems to generalise that of independent chain to non-simple loony endgames. In particular, an independent chain is a string from the ground to the ground. All definitions related to independent chains extend naturally to strings. Therefore, a $k-string$ is a string with k nodes, and it is *long* if $k \geq 3$.

In Figures 12 (top), 13 (left) and 14 (left) we have examples of other types of components that we can find in non-simple loony endgames. Observe that in both cases only loony moves are possible.

In general, in a loony endgame, *only strings and independent loops are offered.*

PROPOSITION 41. *In any loony endgame, any move offers either a string or an independent loop.*

PROOF. The nodes that can be captured must have degree 2, so that every time we capture one by playing its only edge we leave the neighbour node with only one edge, making it capturable too. So joints are really stops in the sense that they stop the captures, and the same applies to the ground. So, unless they form an independent loop, the capturable nodes must go from joint to joint, i.e., they are strings. □

Therefore when we consider a loop we are always referring to an independent loop, since when a complete cycle is offered, either it constitutes a connected component, or it goes from a 4-joint to itself. In the latter case it is considered a string by definition.

1.2. Extending some results to non-loony endgames. All that we proved in the previous chapter for simple loony endgames is valid for general loony endgames. The concept of string substitutes that of independent chain.

For simple loony endgames, "keeping control" (definition 32) means declining the last two boxes of each *independent chain* and the last four of each independent loop. For arbitrary loony endgames, "keeping control" also means declining the last two boxes of each *string* and the last four of each independent loop.

Since in the proofs of Theorem 34 and Propositions 36 and 38 we only used that the loony endgames were simple to assume that the player in control could decline the last two boxes of each chain, we only have to change "chain" by "string" in these proofs, and we have that the propositions hold for arbitrary loony endgames.

1.3. Double-crosses. In general, we capture boxes one at a time. For instance, when offered a 5-string, we can capture all the boxes, each one by playing a single edge. But that is not the case if we decline the last two boxes: then our opponent can capture them both with a single stroke.

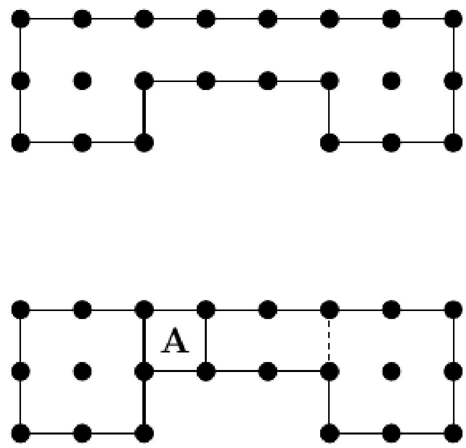

FIG. 12. A pair of earmuffs: after the 3-string ("band") is offered and its two last boxes are declined, we obtain two 4-loops.

DEFINITION 42. *A* double-cross *is a move where two boxes are captured by playing a single edge.*

Therefore, when the player in control declines the last two boxes of a string, his opponent claims them with a double-cross. When a loop is offered, after the player in control declines the last four boxes, his opponent plays two double-crosses, taking all four boxes by drawing just two edges. Note that those are the only two cases when double-crosses can be played, because we proved that the only objects that can be offered in loony endgames are strings and independent loops.

As we will see, the parity of the total number of double-crosses played determines who plays first on the loony endgame. In particular, we will only need to worry about double-crosses played on strings, given that in loops double-crosses are played by pairs.

2. The Chain Formula

Winning Ways gave a strategy for starting positions of Nimstring. In short: if you are the first player, try to make the parity of the number of eventual long chains the same as the parity of the number of initial dots. We also present here a more general formula that is not restricted to starting positions.

Firstly we need to show that the number of double-crosses that will be played on strings, after we reach the loony endgame, is fixed, i.e., does not depend on which moves will be played from then on.

2.1. Counting the double-crosses in a loony endgame. Observe that in a loony endgame the moves of the player in control can be fully predicted: he always takes all the boxes but two when offered a string, and all the boxes but four when offered a loop. On the other hand, his opponent has different lines of play, because he can choose the order in which he offers the available strings and loops. This order can be crucial in some Dots-and-Boxes games but, as we will show, is irrelevant in Nimstring.

The number of offered independent loops can differ depending on the order of the offerings (which, in particular, means that the number of double-crosses played on loops on the loony endgame is not fixed), and this can also be important when playing Dots-and-Boxes but irrelevant in Nimstring. The reason is that in Nimstring what is really important, as we already mentioned, is not the number of double-crosses but its parity.

An example is the earmuffs in Figure 12 (top), where, if the first boxes offered are the middle three, the player in control plays on that 3-string by taking one box and declining the other two (so we have the same behaviour as in an independent 3-chain) and, once his opponent takes these two boxes, what remains are two independent loops. On the other hand, if the first move is on one of the loops on the sides, the boxes in the loop plus the boxes in the middle string behave as a whole as a 7-chain, in the sense that the player in control can take five boxes and decline the last two. However observe that, in any case, there will be only one double-cross played on strings.

Proposition 43. *Let G be a loony endgame of Nimstring with j joints and total valence v. Then the number of double-crosses d that will be played on strings verifies $2d = v - 2j$.*

Proof. Recall that the player that the player in control will win the game by keeping control, i.e., by refusing the last 2 or 4 nodes every time his opponent offers him a string or independent loop, respectively. In this way he forces the opponent to play first in each string and loop. Besides, this is the only winning strategy for the player in control. As each string guarantees that exactly a double-cross will be played on it, we have that d is equal to the number of strings offered.

Observe that, given any loony endgame G, removing an independent loop does not change v, j or d. We will show that removing a string decreases $v - 2j$ by two, which implies, by induction, that $2d = v - 2j$, considering the empty game as base case.

Notice that we only eliminate a joint when the string that we remove has an end that is a 3-joint, or when the string goes from a 4-joint to itself. In the first case v is reduced by 3 and in the second by 4. On the other hand, when an end is the ground, or when an end is a 4-joint and the other end is not the same joint, removing the string only implies a -1 in v and no joint disappears.

On Table 1 we consider the 9 possible cases of removing a string, depending on which type of joint the ends of the removed string are, and compute the variation in v and in j.

As we see in the table, in all the cases we have $\Delta v - 2\Delta j = -2$. □

Remove a string from	**to**	Δv	Δj	$\Delta v - 2\Delta j$
Ground	Ground	-2	0	-2
3-joint	Ground	-4	-1	-2
4-joint	Ground	-2	0	-2
3-joint	Different 3-joint	-6	-2	-2
3-joint	Same joint, 3rd string from joint to the ground	-4	-1	-2
3-joint	Same joint, 3rd string from joint to another joint	-6	-2	-2
3-joint	4-joint	-4	-1	-2
4-joint	Different 4-joint	-2	0	-2
4-joint	Same joint	-4	-1	-2

TABLE 1. Effect on v and j of removing a string, depending on its stops.

FIG. 13. Dipper: the “handle” (3-string on the right) must be offered first, leaving an independent loop (the “cup”).

We have proven that the number of double-crosses that will be played on strings once we reach the loony endgame is fixed, $d = \frac{1}{2}v - j$.

Some examples of single-component loony endgames are the following:

a) A *T* consists on three strings that go from a common joint to the ground. In a T, after offering a string (first double-cross) what remains is an independent chain, so a T is said to be equivalent to two independent chains ($d = 2$).
b) An *X* consists on four strings that go from a common joint to the ground. In an X, after offering the first string, we obtain a T, so we have one more double-cross ($d = 3$).
c) A *dipper* is a loop joined to a string (Figure 13). In a dipper, offering the handle leaves an independent loop, while making a loony move on the cup allows the opponent to take all the boxes as if they belonged to an independent chain. In any case, there is only one double-cross ($d = 1$).
d) *Shackles* are formed by two loops joined by a string (Figure 14, left). In a pair of shackles, when a loony move is played what the player in control is offered is a 3-string. In order to keep control, he must decline the last two boxes. After his opponent takes these two boxes, what remains is an independent loop ($d = 1$).
e) We already commented what happens in a pair of earmuffs (Figure 12).

We compute d in each case in Table 2. By playing each example we can verify that d is effectively the number of double-crosses played on strings.

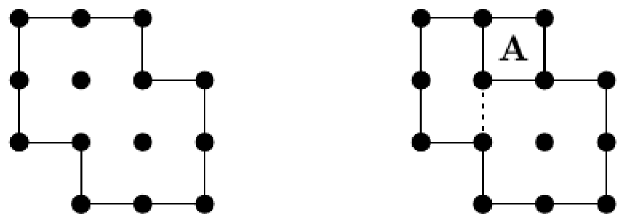

FIG. 14. A pair of shackles: after a 3-string is offered and its two last boxes are declined, we obtain a 4-loop.

Component	v	j	$d = \frac{1}{2}(v - 2j)$
Independent chain	2	0	1
Independent loop	0	0	0
T	6	1	2
X	8	1	3
Dipper	4	1	1
Earmuffs	6	2	1
Shackles	4	1	1

TABLE 2. Value of d for some single-component loony endgames

2.2. Why do we "count chains"? Although we generalised the concept of independent chain to that of string, we do not usually count strings, the reason being that "number of strings" is not the same as "number of double-crosses played on strings". As an example, in a pair of shackles, even though there are two strings with a common joint, only one will be offered, because after the offering and the ensuing capture what will remain will be an independent loop. In fact, we can say that shackles *behave as an independent 3-chain plus an independent 4-loop.* That is the reason why we can informally say that "shackles count as one chain", because, as it happens in an independent chain, $d = 1$. This is nothing more than a shorter way of saying that "the number of double-crosses played on strings will be one". It is not unusual in the bibliography to refer to d in that way, and this is the reason why we usually say that we are "counting chains" in a game, while stricto sensu we should say that we are counting "the number of double-crosses that will be played on strings".

For instance, we can informally say that "a T is equivalent to two (independent) chains", because $d = 2$, or that "a dipper counts as one chain", which, in fact, means that in any game we could substitute a dipper by an independent chain without modifying d (which will turn out to be what matters to determine the outcome).

2.3. The Generalised Chain Formula. We know that the parity of the number d of double-crosses played on strings is fixed once we reach the loony endgame, we are going to show that this parity determines which player is in control.

NOTATION 44. *Given any game G, we denote by*

i) $e(G)$ *the number of (not yet played) edges of* G*, and* $b(G)$ *the number of (uncaptured) boxes of* G*.*
ii) Previous *(short for "previous player") the player who made the last move.*
iii) Next *the player who has to make the next move.*
iv) d *the number of double-crosses that will be played on strings until the end of the game.*
v) t *the number of turns that remain until the end of the game.*
vi) f *the number of turns that will be played before reaching the loony endgame.*

Note that d, t and f may depend on how the game is played, and therefore we may not be able to determine its value, though we have already proven that if G is a loony endgame then d is fixed. The following lemma gives a relation between the values of d and t.

LEMMA 45. *Let* G *be any Nimstring game. Then* t *only depends on* d*, and verifies*

$$t = e(G) - b(G) + d$$

PROOF. Each edge still available to be played allows us to make a move. But each time we claim a box we play again, so we have "wasted" an edge without consuming a turn. Besides, when a player declines the last two boxes of a string his opponent takes two boxes with a single strike (double-cross). To sum up, each edge contributes by one to the number of turns, each box subtracts one, but for each double-cross we have to count two boxes as one. So we have that $t = e(G) - b(G) + d$. Given that $e(G)$ and $b(G)$ are fixed, t only depends on d. □

Now we can prove that the parity of d determines the winner.

NOTATION 46. *In* $n \equiv m$*, where* $n, m \in \mathbb{N}$*,* $\equiv$ *always means congruence modulo 2.*

THEOREM 47. *(Generalised Chain Formula) If* G *is a game of Nimstring, then Previous wins the game iff the* $d \equiv e(G) + b(G)$*.*

PROOF. Observe that the loser of a Nimstring game is, as in any combinatorial game following the normal ending condition, the player who is unable to play his turn. We are assuming that, even in the last component, the player in control will refuse the last 2 or 4 boxes. Once his opponent takes those boxes, he will not be able to play again as required by the rules, so he will not be able to complete a proper turn and will lose[1].

Therefore Previous will win if the number of turns played until the end is even. According to Lemma 45, $t = e(G) - b(G) + d$, so Previous wins iff $d \equiv e(G) + b(G)$.

□

[1]On the other hand, if we were playing Dots-and-Boxes instead of Nimstring, the player in control would take all the boxes in the last component and he would be the last to play.

The next corollary corresponds to the "chain rule" given on Winning Ways [**Ber01**], and allows us to determine, given the starting position of a game on a rectangular grid, which parity of d needs each player in order to win.

Corollary 48. *Consider a Nimstring game G in a rectangular grid of $n \times m$ dots where no edges have been played yet. Then the first player wins if the number d of eventual double-crosses on strings has the same parity as the number of dots.*

If particular, in a square grid of $n \times n$ dots the first player wins if $n \equiv d$.

Proof. We have that $e(G) = n(m+1) + m(n+1)$, $b(G) = (n-1)(m-1)$. Then $e(G) + b(G) = 3nm + 1$.

In the starting position, the first player is the next player. Therefore, by Theorem 47, he wins iff $d \not\equiv e(G) + b(G)$, i.e., $d \equiv nm$. Observe that nm is the number of dots.

□

Corollary 49. *In any Nimstring game G, if the parity of d is already fixed then the parity of f is also fixed, and Previous wins iff f is even.*

Proof. The number of turns after reaching the loony endgame, $t - f$, must be even (because the first to play there loses), therefore $f \equiv t \equiv e(G) + b(G) + d$.

□

This corollary says that, once the parity of d is fixed, we can fill the game with arbitrary moves (as long as we keep the given parity of d unchanged) up to the loony endgame in order to find out who wins.

We proved that in any loony endgame we have $d = \frac{1}{2}v - j$. Another characterisation of d is the following:

Corollary 50. *If G is a loony endgame, then $d \equiv e(G) + b(G)$.*

Proof. In a loony endgame $f = 0$, so Previous wins in G and, by Theorem 47, $d \equiv e(G) + b(G)$. □

2.4. Chain Battle Examples. Let us apply the results obtained in the previous subsection. We will use Theorem 47 or any of its corollaries to determine the winning move for Next.

Observe that we can find out the desired parity of d faster using that in any component H that is a loop $e(H) + b(H)$ is even (because $e(H) = b(H)$) while in any component K that is an independent chain $e(K) + b(K)$ is odd (because $e(K) = b(K) + 1$). Therefore when computing $e(G)$ and $b(G)$ we can ignore loops and independent chains, and just add 1 for each short or long chain, without altering its parity.

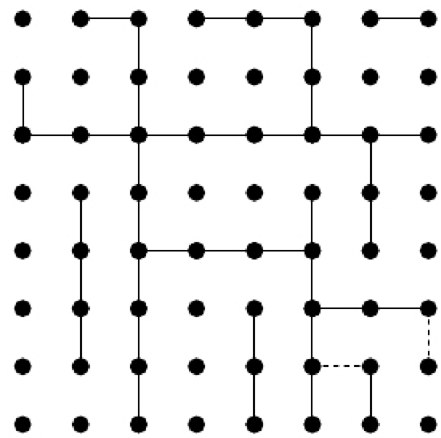

FIG. 15. In this game d depends solely on what happens on the bottom right corner. Next can force one chain there with the vertical dotted move, or no chain at all with the horizontal one.

2.4.1. *Example 1.* Firstly consider again the game in Figure 11 (top) in the light of the formula of Theorem 47: $e(G) = 32$ and $b(G) = 25$, so Next needs an even number of chains. Using the observation of the previous paragraph instead, as after the sacrifice we have 7 edges and 6 boxes in the central component, while the rest of the game is formed by 5 short or long chains, we can find out faster that th parity of d needed by Next is the same as the parity of 7+6+5=18.

We could have used Corollary 48: in the 5×5 boxes starting position we have 36 dots and 25 boxes, so the first player needs an even number of chains. Given that 28 moves have already been played, the first player is the next player.

Now that we know that Next wants d to be even, we observe that there are already 2 chains (a 3-chain on the left and a 9-chain on the right). At most, a third chain can be created in the first row of boxes. So the parity of the number of double-crosses on strings d depends only on what happens on that first row: the next player can either create a 4-chain there (so $d = 3$) or make the dotted sacrifice to avoid a chain there (so $d = 2$). Therefore the right decision was to make the sacrifice. Now we can assert that it was his only possible winning move, because it prevents the formation of a third chain, while if he had played elsewhere his opponent would have won by creating the third chain, obtaining a game like that in Figure 4).

A third way to find the winning move is to use Corollary 49. We can easily see that the after the sacrifice move $f = 4$, while creating the 4-chain would make $f = 3$. Since a winning move must leave f even, the former is the right one.

2.4.2. *Example 2.* Let us consider a bit more complicated example, and solve it also using different results. In Figure 15 any experienced player knows that the number of chains in each component is fully resolved, save for the bottom right corner component. Of the three components on top, the left and the middle ones will form loops, while the one on the right will become a chain (for any move played

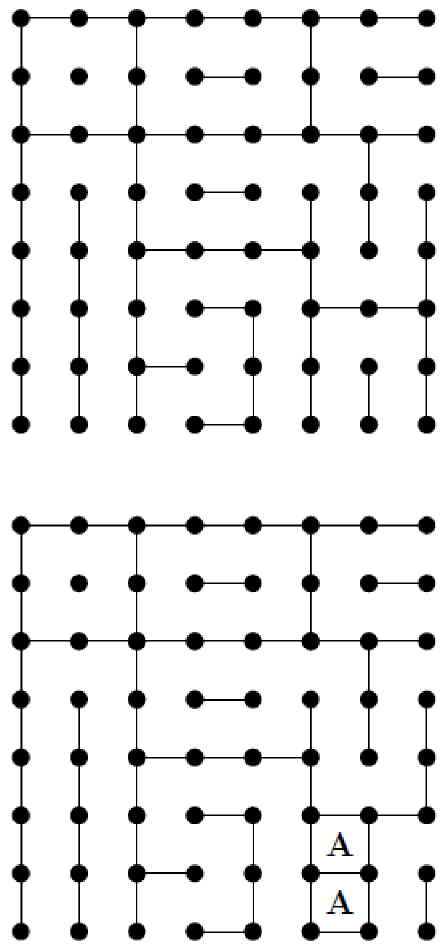

Fig. 16. Depending on the dashed move chosen by Next in Figure 15, we can end up with $d = 5$ (top) or $d = 4$ (bottom).

there, there is an answer that ensures a chain there). In the center we have a dipper, and just below we can end up having a chain or another dipper (in any case, they count as one chain). To the left we have another chain. Finally, on the right bottom corner, Next can force a chain (so $d = 5$) with the vertical dotted move, or ensure there will be no chain with the horizontal one (so $d = 4$). Next must move there in order to obtain the parity of d he needs, or his opponent will do it in his next move.

A first way of determining the winning move would be to use Winning Ways's formula (Corollary 48): there are 64 dots, so the starting player needs d to be even. We can count 38 turns played from the starting position, so Next is the starting player, which confirms that he wins with $d = 4$, so he has to make the horizontal dashed move.

Once Next has made one of the dashed moves, d will be fixed, and so will be the parity of f by Corollary 49. A second way to determine which parity is needed by Next is to use that corollary. Consider the two possible scenarios after each of the dashed moves, and arbitrarily fill the board with edges as long as we do not make any loony move nor alter the number of chains in each component (in fact, it would be enough keeping the parity of d corresponding to each case) until we obtain a loony endgame. A possible way of doing so is shown in Figure 16: in the upper figure $d = 5$ because Next has created a chain in the bottom right corner, while in the one below $d = 4$ because Next has forced no chains there. In order to win, Next must make his move so that after it the value of f is even. In the case $d = 5$ we have filled $f = 19$ edges, while on the case $d = 4$ we have drawn $f = 20$ edges. Therefore the second case (corresponding to the horizontal dotted move on Figure 16) is the winning one. This is a fast method in this case, given that a player can determine f by visual inspection in a few seconds.

2.5. Summary. To sum up: once we reach the loony endgame each string that is offered will contribute with a double-cross and each independent loop with two. Given that the number of double-crosses played on independent loops will always be even, they do not affect the parity of the number of double-crosses, and so we do not need to count them. The parity of the number d of double-crosses that will be played on strings is informally called the "the chain parity" of the loony endgame, because the game is equivalent to d independent chains.

What Theorem 47 and Corollary 48 tell us is that each player must strive for a given parity of the "number of chains" (formally, d) in order to force the opponent to move first in the loony endgame. Therefore, even though we are playing an impartial game, and it may seem that any move cannot benefit or harm a player over the other (because played edges do not "belong" to anyone, as it happens, for instance, with chess pieces), in fact each player has a goal to achieve from the very beginning: one needs d to be an even number, and the other needs it to be odd. We call this "chain battle".

Chapter 4
Resolved components in Nimstring

In this chapter we show two ways of simplifying the analysis of games. A first way is to enforce some *canonical* restrictions that reduce the number of options available to a player in such a way that the outcome is not affected. Another useful simplification is that we can assume, as we will prove, that the parity of the number of chains in components whose nim-value is 0 or 1 is fixed, allowing us to concentrate on the analysis of the components with bigger nim-values.

1. Canonical Play

1.1. Canonical play. Suppose that we have a restriction such that, for any game G, if a player can win in G he can also win playing with that restriction. To simplify the analysis of games, we will assume that games are played according to some such restrictions. This is what we call *canonical play*.

For a restriction to qualify as canonical we have to prove that, in any game G, the winner has a winning strategy in which he follows the restriction, even when his opponent is not willing to follow it. In other words, a canonical restriction verifies that in any game the winner has an option that is both a canonical move and a winning move. Once we impose a canonical restriction, we will assume that no player will make an uncanonical move; note that the loser in G will lose playing canonically or not, so, once we prove that the winner wins with a canonical restriction even if his opponent plays uncanonically, we can assume that the loser also follows the restriction. This consideration allows us to reduce the number of possible options available to a player to the subset of canonical moves, which can considerably simplify the analysis.

1.2. Canonical Restrictions in Nimstring. Let us see some examples of canonical restrictions in Nimstring.

Proposition 51. *We can consider canonical restrictions not playing*

- *Loony moves before the loony endgame.*

- *Half-hearted handouts.*[1]

PROOF. Both kind of moves are losing moves, as proved by Lemma 35 and and Proposition 36. So, in any game, any winning strategy that is available to the winner will not include any of these moves. □

The above example rules out two kinds of moves that are more than uncanonical: they are losing moves. Let us see an example where there are uncanonical winning moves.

In any game where not all the components are zero, we will consider canonical moves only those which decrease the nim-value of a connected component. For instance, if G has two connected components of nimbers $*1$ and $*3$ respectively, any canonical winning move must be one played on the latter component to some $*1$ option, while any move in the first component to $*3$ (if available) would be non canonical. The following proposition proves that there is always a move of the former type available for the winner:

PROPOSITION 52. *If $G \in \mathcal{N}$ then, in any decomposition of G in (not necessarily connected) components, Next has a winning option that decreases the nim-value of a component.*

PROOF. Consider a game $G = G_1 + \cdots + G_k$. Assume Player A, being the next player, can win, and that all the winning moves increase the nim-value of a component. After he plays any of them, his opponent is able to reverse the move[2]. Though after these two moves the component has changed, its nim-value remains the same. Unless Player A plays at some time a move that decreases the nim-value of a component, his opponent will be able to reverse all moves. In this case, the game would not end on a finite number of turns. Therefore, Player A will eventually have to play a move that decreases the nim-value of a component. At this point the game is some $G' = G'_1 + \cdots + G'_k$, where each G'_i may be different that the original component G_i. But, since the nim-values of the components have not changed because Player B has been reversing Player A's moves, we have that $\mathcal{G}(G'_i) = \mathcal{G}(G_i)$ for all $1 \leq i \leq k$. If now Player A has a winning move in, say G'_1, that decreases its nim-value to some $n < \mathcal{G}(G'_1)$, then we claim that he had a winning option that also decreased its nim-value on his first move, in the same component. That is because $*n \in G_1$ since $n < \mathcal{G}(G'_1) = \mathcal{G}(G_1)$, and so Player A could have made his first move from $G = G_1 + \cdots + G_k$ to $*n + G_2 + \cdots + G_k$.

□

Consider, for instance, a game G decomposed into three components, $G = G_1 + G_2 + G_3$, whose respective nim-values are 1, 5 and 7. Observe that, in fact, we do not care about the actual components, but only about its nim-values. In $G = *1 + *5 + *7$

[1]As we already mentioned, half-hearted handouts are considered loony moves in the bibliography; see [**Ber00, Ber01**].

[2]Recall from Definition 19 that to reverse a move is to take the component where it takes place the move back to the nim-value it had before, and that moves to bigger nim-values are always reversible.

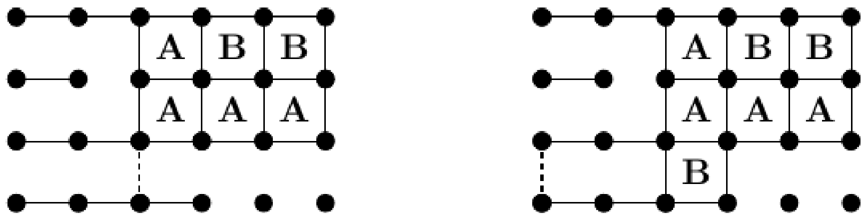

Fig. 17. Left: an example of Dots-and-Boxes game where Next can only win by playing a loony move. Right: the reply by Player B. Player A will eventually win 8-7, after taking the two declined boxes from the 3-chain and the 2-chain on the bottom right.

Next can win by moving in the first component to $*2$ if this move is available, obtaining $*2 + *5 + *7 = 0$. But this move can be reversed by his opponent, and though then we have a game that is not identical to G, it is equivalent, also being $*1 + *5 + *7$. Since $G = *1 + *5 + *7 = *3 \neq 0$, $G \in \mathcal{N}$, so Next must have a winning move that decreases the nim-value and cannot be reversed. In this case, he can move in the last component to $*4$, to obtain $*1 + *5 + *4 = 0$.

Note that different decompositions of G may lead to different decreasing nim-value moves. We will usually consider the decomposition of G in *connected* components.

Corollary 53. *If $G \in \mathcal{N}$ then Next has a winning option that decreases the nim-value of a connected component.*

We will consider canonical only those moves which decrease the nim-value of a connected component.

1.3. Canonical Restrictions in Dots-and-Boxes. Although this chapter is focused on Nimstring, we show some useful canonical restrictions for the Dots-and-Boxes game, as some of the ones shown for Nimstring can also be considered canonical in Dots-and-Boxes. Let us start with one in which that is not the case: a player may be forced to play a loony move before the loony endgame in order to win at Dots-and-Boxes, and therefore we cannot consider that not playing loony moves is a canonical restriction.

Proposition 54. *In a game of Dots-and-Boxes, the winner does not always have a non-loony option that is a winning option.*

Proof. In Figure 17 (left) we show an example of game where Next only can win by playing a loony move, even though there are non-loony options. Player A, leading 4-2, is the next player. With the dashed move he will eventually win 8-7. Should he not play a loony move, his best option is the dashed edge in Figure 18 (left), after which he would end up losing 7-8. □

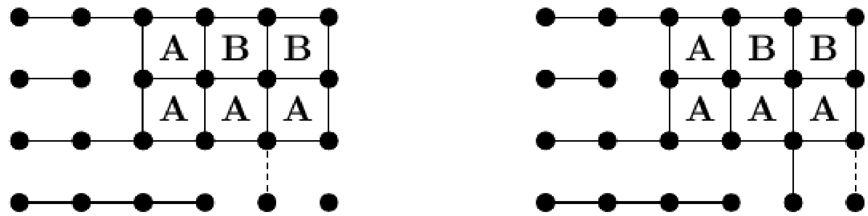

FIG. 18. Left: loony move by Player A leads to a loss. Right: Player B's reply. Player A will lose 7-8, after taking only the 1-chain and 2 declined boxes from one of the 4-chains.

PROPOSITION 55. *Not playing half-hearted handouts*[3] *can be considered a canonical restriction in Dots-and-Boxes.*

PROOF. Assume a player wins by playing a half-hearted handout. Then we assert that he can also win by playing a hard-hearted handout. We only need to observe that after playing a half-hearted handout the opponent has two options: take the two boxes and play first on the rest of the game, or decline the two boxes and play second on the rest of the game. On the other hand, after a hard-hearted handout the opponent only has the former of the two. If the opponent cannot win having both options, he cannot win having only one of them. That is to say, narrowing the number of options of the opponent cannot improve his position. □

The following two restrictions can be considered canonical in both Nimstring and Dots-and-Boxes:

PROPOSITION 56. *i) When offered a string (resp., loop) always take, at least, all boxes but two (resp., four).*
ii) When offered a short chain, always take it.

PROOF. i) Suppose that in a game G you are offered an n-string. Let G' be the rest of the game. After you decline $k \geq 2$ boxes your opponent will take them and will move first in G' regardless of the value of k. Since you will claim $n-k$ boxes of the n-chain and your opponent k, you maximize your net gain of $n-2k$ by declining only 2 boxes. If offered a loop, the proof is analogous.
ii) Suppose that in a game G you are offered a short chain. As in the prior case, let G' be the rest of the game. Since you will move first[4] in G' whether you capture the boxes in the short chain or not, the best move is to claim them.

□

The prior proposition says nothing about what to do once we are down to the last 2 boxes of a string or the last 4 of an independent loop. While declining them is

[3]Recall that there are two kinds of moves on a 2-chain: hard-hearted handouts and half-hearted handouts (Figures 8 and 9 from Chapter 2, resp.).
[4]As we have shown that we can assume that players will not make any half-hearted handouts, neither in Nimstring nor in Dots (see Propositions 51 and 55).

the only way to win in Nimstring, in Dots-and-Boxes that option cannot even be considered canonical. For instance, when a Dots-and-Boxes game is reduced to a single independent chain, declining the last two boxes is strictly worse than taking them.

2. Canonical Play in 0 and *1 Nimstring Games

Berlekamp [**Ber00**, p. 50] stated that 0 and $*1$ components are resolved, in the sense that we can assume that the parity of the number of chains (double-crosses on strings) there is fixed:

> Any position in which the number of long chains is resolved has nimber 0 or 1, according as the parity of the number of long chains, plus the number of nodes (boxes) plus the number of edges (moves) is even or odd.

In order to prove this result, we need to introduce some notation, and propose a definition of the term "resolved".

NOTATION 57. *i) We say that a game G is BIG if $\mathcal{G}(G) \geq 2$.*

ii) Given a game G, we denote by G^* the game into which G has eventually become when reaching the loony endgame (that is, G^* is a position of G that is a loony endgame). In the same way, when we have a game as a disjunctive sum of components, as in $G = H + K$, we will write H^* and K^* to denote the games into which components H and K, resp., have eventually become when reaching the loony endgame.

iii) As in the prior chapter, we keep using d for the number of double-crosses that will be played on a game G, but now we will write $d(G)$ to make clear to which game, or component of a game, we are referring to.

DEFINITION 58. *A game H is* resolved *if there is some parity (either even or odd) such that, for any game K, the winner in $G = H + K$ has a winning strategy such that $d(H^*)$ has the given parity.*

2.1. 0 and *1 Games are Resolved.

PROPOSITION 59. *Consider a Nimstring game $G = H + K$.*

If $\mathcal{G}(H) < 2$, then there is a (canonical) way of playing for the winner in G such that

$$d(H^*) \equiv b(H) + e(H) + \mathcal{G}(H)$$

Observe that if neither H nor K are BIG then the canonical way of playing that we describe in the proof must ensure that the property holds for both H and K.

PROOF. Recall from the proof of Lemma 45 that the parity of $e(G) + b(G)$ always changes at each turn, because each turn the number of edges played minus the number of boxes claimed is equal to 1 (as there are no double-crosses before the loony endgame). Note that any turn played on a given component, either H or K, also changes the parity of edges plus boxes in that component. Also recall that in any game that is equal to $*m$ there are options to any $*k$ such that $0 \leq k < m$ (and maybe to some $*r$ such that $r > m$).

We have to proof that, fixed $H \in \{0, *1\}$, for any K there is a winning strategy in $G = H + K$ for the winner in G such that

i) If $H = 0$, then $d(H^*) \equiv b(H) + e(H)$.
ii) If $H = *1$, then $d(H^*) \not\equiv b(H) + e(H)$.

We consider four cases. In the first two $G = 0$, so Previous is the winner, while on the last two $G \neq 0$, and it is Next who wins. In Figure 19 we can see the diagram of changes of state of the second case as detailed by the following proof; observe that the diagram is valid for the other cases as well: just start on the corresponding state.

(1) $H = K = 0$:
Previous reverses each move by Next turning the same component back to 0. Eventually we reach the loony endgame $H^* + K^*$, where both components are zero and it is Next turn to play.
As Previous always replied in the same component where Next just played, an even number of turns has been played in each component, which implies that $e(H^*) + b(H^*) \equiv e(H) + b(H)$. Using that $d(H^*) = e(H^*) + b(H^*)$ by Corollary 50, we have that $d(H^*) \equiv e(H) + b(H)$. The same applies to K.
(2) $H = K = *1$:
While Next moves in any component to some BIG position, Previous reverses it back to $*1$. When Next moves in a component to 0 (he will be forced eventually to do so), Previous moves in the other to 0. Once both components are zero, Previous proceeds as in case 1 (see Figure 19). In this case, an odd number of turns has been played in each component because only one move in each component has not been reversed. So we have that $d(H^*) \equiv e(H^*) + b(H^*) \not\equiv e(H) + b(H)$, and analogously $d(K^*) \not\equiv e(K) + b(K)$.
(3) $H = 0, K = *1$:
Next makes his first move in K to some option $K' \in K$ such that $K' = 0$. Having made a move in K and no moves in H, Next then proceeds as in case 1 (playing the role that Previous plays there). Therefore we have $d(H^*) \equiv e(H^*) + b(H^*) \equiv e(H) + b(H)$, while $d(K^*) \equiv e(K^*) + b(K^*) \equiv e(K') + b(K') = e(K) + b(K) - 1$, so $d(K^*) \not\equiv e(K) + b(K)$.
(4) $H = 0$ or $H = *1$, with K BIG:
Next makes his first move in K to some option $K' \in K$ such that $K' = H$. Without any moves played in H, we are now in the situation of either case 1 or case 2. Next can proceed as in that case (playing the role that Previous plays there). Therefore, if $H = 0$ then $d(H^*) \equiv e(H^*) + b(H^*) \equiv e(H) + b(H)$, and if $H = *1$ then $d(H^*) \equiv e(H^*) + b(H^*) \not\equiv e(H) + b(H)$.

□

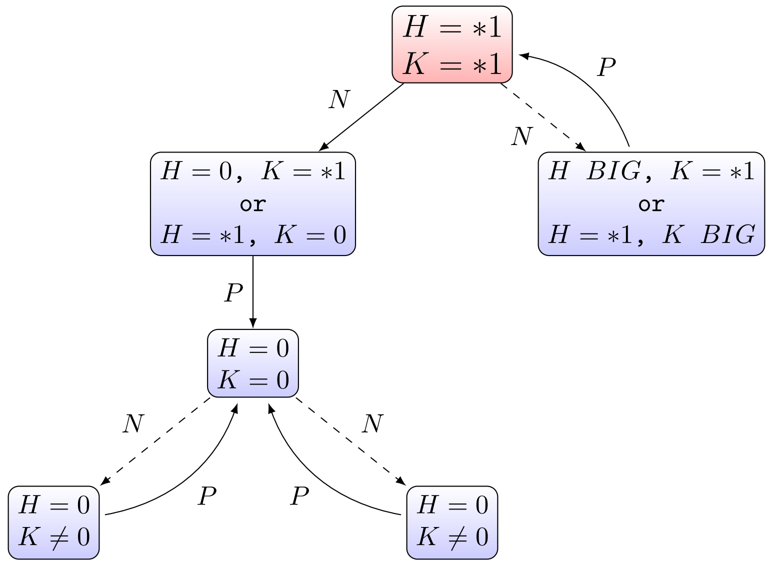

FIG. 19. Diagram of changes of state for case 2: Game starts on the state $H = *1; K = *1$ and must be on the state $H = 0; K = 0$ when reaching the loony endgame. N indicates moves by Next and P moves by Previous; dotted lines indicate moves that may or may not be available.

We will consider canonical the way of playing described on the proof. Since the parity of $d(H^*)$ that the winner in G can force does only depend on H, Proposition 59 proves that any game whose nim-value is 0 or 1 is resolved according to Definition 58. This result allows us to "forget" about 0 and $*1$ components and concentrate on components that are BIG. An example is the following result.

COROLLARY 60. *If* $H = 0$ *or* $H = *1$*, then Previous wins in* $G = H + K$ *iff* $d(K) \equiv e(K) + b(K) + \mathcal{G}(H)$.

PROOF. By Theorem 47, Previous wins in G iff $d(G) \equiv e(G) + b(G)$. Note that $d(G) = d(H) + d(K)$.

Since $d(H) \equiv b(H) + e(H) + \mathcal{G}(H)$, Previous wins iff $d(G) \equiv e(G) + b(G) = e(H) + b(H) + e(K) + b(K) \equiv d(H) + \mathcal{G}(H) + e(K) + b(K)$, that is, iff $d(K) = d(G) - d(H) \equiv b(K) + e(K) + \mathcal{G}(H)$.

□

In Proposition 59 we considered games decomposed into only two components. Now we show that, if we decompose a game into any number of components, the winner can force the parity of *all* 0 or *1 components simultaneously. Observe that this *simultaneity* is more than what our definition of *resolved* requires, and therefore we have a stronger result.

THEOREM 61. *Consider a decomposition of a Nimstring game in (not necessarily connected) components,* $G = H_1 + \cdots + H_n + K_1 + \cdots + K_m + L_1 + \cdots + L_s$*, such that* $H_1 = \cdots = H_n = 0$*,* $K_1 = \cdots = K_m = *1$ *and* $L_1, \cdots, L_s$ *are BIG. Then there is a (canonical) way of playing for the winner in* G *such that* $d(H_i^*) \equiv b(H_i) + e(H_i)$ *for all* $1 \leq i \leq n$*, and* $d(K_j^*) \not\equiv b(K_j) + e(K_j)$ *for all* $1 \leq j \leq m$*.*

PROOF. The proof is very similar to that of Proposition 59, so, instead of a case-by-case proof, we opt for a more compact one.

If $G \in \mathcal{P}$, the winning strategy for Previous (until reaching the loony endgame, where all components are 0) is based in the following replies:

i) Reverse any move that increases the nim-value of a component.
ii) Reply to any move that decreases the nim-value of a component with any winning move that also decreases the nim-value of some component. Such a move exists by Proposition 52.

Observe that, until reaching the loony endgame G^*, in each H_i an even number of moves takes place, as all the moves there by Next are reversed by Previous. On the other hand, in each K_j the number of moves is odd, since there is one move from $*1$ to 0, while the rest of the moves played there by Next are reversed. Therefore we have $d(H_i^*) \equiv e(H_i^*) + b(H_i^*) \equiv e(H_i) + b(H_i)$ for all $1 \leq i \leq n$ and $d(K_j^*) \equiv e(K_j^*) + b(K_j^*) \not\equiv e(K) + b(K)$ for all $1 \leq j \leq m$.

If $G \in \mathcal{N}$, Next plays any nim-value decreasing winning move, then proceeds as in the precedent case, and we have again $d(H_i^*) \equiv e(H_i) + b(H_i)$ for all $1 \leq i \leq n$ and $d(K_j^*) \not\equiv e(K) + b(K)$ for all $1 \leq j \leq m$. □

2.2. BIG Games are not Resolved. If a component of a game is BIG, we do not have a result similar to Proposition 59. This is because, for any BIG game H, there are always infinitely many games in which H is a component, and where the loser can force any parity he desires in H.

PROPOSITION 62. *Let* $G = H + K$*,* $H = *n$ *and* $K = *m$*, where* $n, m \geq 2$*. Then there are two strategies for the loser in* G *such that following one of them* $d(H^*)$ *is even, and following the other one* $d(H^*)$ *is odd.*

PROOF. <u>Case 1</u>: $G \in \mathcal{P}$.

We have that $G = 0$, $H = K$ and $d(G) \equiv e(G) + b(G)$.

Let $\bar{H}$ and $\bar{K}$ denote arbitrary positions of H and K, resp. Recall that a position of a game G, according to Definition 3, is any game obtained after playing some moves (maybe one or none) on G. Note that when it is Next's turn to play in some $\bar{G} = \bar{H} + \bar{K}$, since $\bar{G}$ must be 0, we have that $\bar{H} = \bar{K}$.

Consider the following strategies for the losing player (Next):

Strategy 1:

i) If $\bar{H} = \bar{K} \neq 0$, move in any component to 0.

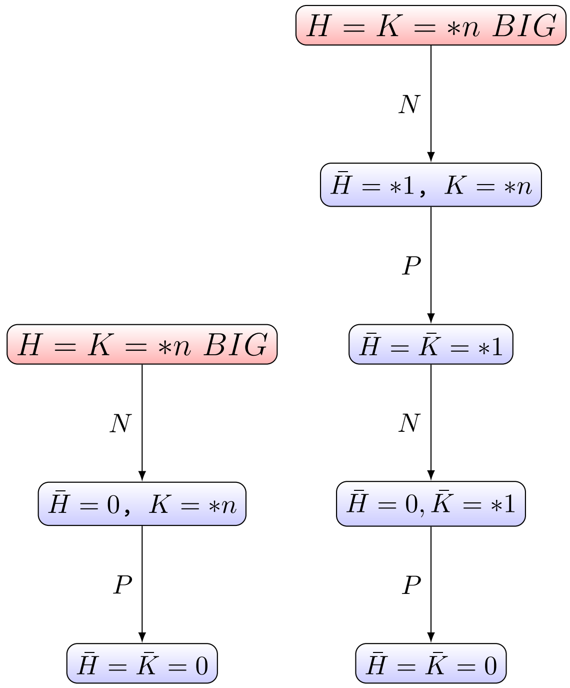

FIG. 20. Unreversed moves in $G \in \mathcal{P}$ if Next follows Strategies 1 (left) and 2 (right), which allow him to force odd or even $d(H^*)$, resp.

ii) If $\bar{H} = \bar{K} = 0$, and $\bar{G}$ is not a loony endgame, make any (non-loony) move.

Strategy 2:

i) In first move, move in H to $*1$. While Previous reverses by playing in $\bar{H}$ back to $*n$, move in $\bar{H}$ to $*1$ again.
ii) Once Previous has moved in K to $*1$ (eventually he will have to, otherwise the game would not end in a finite number of turns), move in $\bar{H}$ to 0. While Previous reverses by playing in $\bar{H}$ back to $*1$, move $\bar{H}$ to 0 again.
iii) Once Previous has moved in $\bar{K}$ to 0 (which will happen sooner or later), so both $\bar{H}$ and $\bar{K}$ are 0, follow Strategy 1.

Eventually we must reach a loony endgame G^* with $H^* = K^* = 0$ in both strategies. Let us count the parity of the number of moves that have been played in each component. Reversed moves do not affect parity. In Strategy 1 only one unreversed move has been played in each component, while in Strategy 2 two unreversed moves have taken place in each component. Those moves correspond to the changes in state in Figure 20, where we only represent unreversed moves.

Therefore in Strategy 1 we have $d(H^*) \equiv e(H^*) + b(H^*) \not\equiv e(H) + b(H)$. On the other hand, in Strategy 2 we have $d(H^*) \equiv e(H^*) + b(H^*) \equiv e(H) + b(H)$. In conclusion, the loser (Next) can force $d(H^*)$ to have any parity he desires. We have the same result for $d(K^*)$. (Observe that, though Next can choose the parity of

$d(H^*)$ or the parity of $d(K^*)$, he cannot choose both at the same time, or, since $d(G^*) = d(H^*) + d(K^*)$, he would win!)

Case 2: $G \in \mathcal{N}$.

After one move by Next, we are in case 1 (with the roles of Previous and Next swapped), and we have proved that in that case the loser can force any parity of $d(H^*)$.

□

We have just proved that BIG games are not resolved.

PROPOSITION 63. *If $G = H + K$ with $\mathcal{G}(H) < 2$ and $\mathcal{G}(K) \geq 2$, then the winner has a winning strategy in which the parity of $d(K^*)$ is fixed.*

PROOF. By Proposition 59, the winner has a winning strategy in which $d(H) \equiv b(H) + e(H) + \mathcal{G}(H)$. As $G \in \mathcal{N}$, by Theorem 47, we have $d(G^*) \not\equiv e(G) + b(G)$. Therefore $d(K^*) = d(G^*) - d(H^*) \not\equiv e(G) + b(G) - (b(H) + e(H) + \mathcal{G}(H)) \equiv e(K) + b(K) + \mathcal{G}(H)$. □

Proposition 63 shows that, in some games, the winner can force the parity of $d(K^*)$, where K is a BIG component. However this does not mean that K is resolved, which would mean that, for any game H, the winner in $H + K$ is able to force the parity of $d(K^*)$. That is not the case for any BIG game K: it is enough to consider any game H that is also BIG and, by Proposition 62, the loser in $H + K$ can choose the parity of $d(K^*)$.

2.3. Examples. Which components are resolved and which are not depends on the decomposition we are considering. We usually consider the decomposition of the game into its connected components.

Consider the game G in Figure 21 (left). Let G_1 be the component in the upper left, G_2 the one in the upper right, and G_3 the chain at the bottom. G_3 is resolved ($d(G_3^*) = 1$), being an independent chain, and $G_3 = 0$ because all its options are loony. The other two are unresolved[5], since $G_1 = *2$ and $G_2 = *3$. As

$$G = G_1 + G_2 + G_3 = 0 + *2 + *3 = *1,$$

we have $G \in \mathcal{N}$. Observe that, by Theorem 47, Next wins iff $d(G^*)$ is odd. We will show that, even though Next can force $d(G^*)$ to be odd, his opponent can choose the parity of $d(G_1)$ or the parity of $d(G_2)$, which correspond to the unresolved components.

Possible winning moves for Next would be:

i) In G_1, from $*2$ to $*3$ (if available).
ii) In G_2, from $*3$ to $*2$.
iii) In G_3, from 0 to $*1$ (if available).

[5]The computation of its nim-values can be found in [**Ber00**, p. 46] and in [**Ber01**].

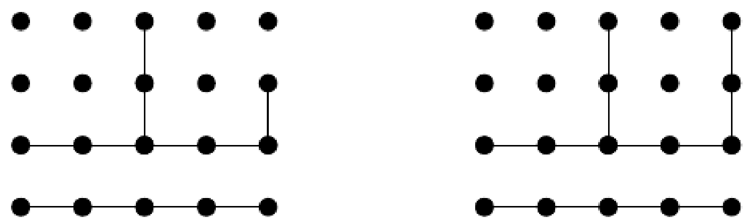

FIG. 21. Next plays a canonical winning move. $d(G^*)$ will be odd.

Those are the three cases that lead to a game with nim-sum equal to 0. Notice that only moves that decrease the nim-value of a component are guaranteed to exist. For instance, all options of G_3 are loony, and so $*1$ is not an option of G_3. On the other hand, it is possible, although uncanonical, to move in G_1 from $*2$ to $*3$ (moving to a symmetrical mirror image of G_2), but we already know that such a move can be reversed[6].

As an example, consider that Next plays in G_2 to some $*2$ option, for instance the one shown in Figure 21 (right). Let us prove that the loser can force any parity he wishes in G_2. Consider two of the possible moves by the loser (Figure 22 (top)). Below each one there is the only winning reply by Next.

Observe that, by Theorem 47, Next wins iff $d(G^*)$ is odd, so the replies are winning moves, though in the case on the left $d(G^*) = 3$ while in one on the right $d(G^*) = 1$. On the case on the left, both upper components will end up being a chain. However, on the case on the right, both will end up without chains. Formally, in the former $d(G_1^*) = d(G_2^*) = 1$, and on the latter $d(G_1^*) = d(G_2^*) = 0$.

We could also prove that the loser can force any parity he wishes in G_1: we would just have to consider the two moves in G_1 shown in Figure 22 (bottom), which lead to, respectively, $d(G_1^*) = 1$ and $d(G_1^*) = 0$.

In conclusion, the loser can choose the parity in either G_1 or G_2, then the winner will force in the other of the two components that same parity, to achieve an odd parity in the whole game G.

Observe that we could consider a decomposition in G other than in connected components, like $G = G_3 + H$ (where $H = G_1 + G_2$). As $G_3 = 0$ and $H = *2 + *3 = *1$, both components in this decomposition of G are resolved. In particular, $d(H^*) \equiv e(H) + b(H)$ will be even (either 0 or 2).

[6]Notice that we did not assume in Proposition 62 that the winner plays canonically; therefore he cannot prevent his opponent from choosing the parity of $d(G_2)$ by playing uncanonically.

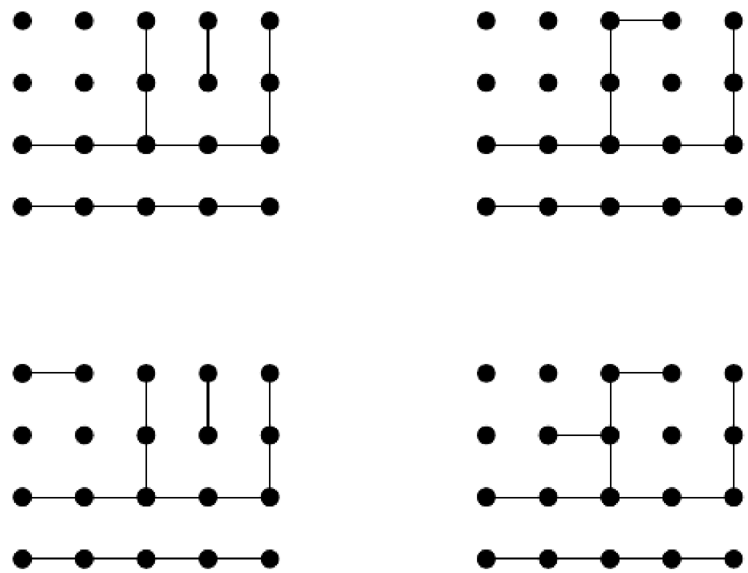

FIG. 22. On top, two possible replies in G_2: the left one creates a chain, while the right one ensures no chains in G_2. At the bottom, the corresponding replies in G_1: the left one ensures a chain there (for a total of 3 in G), while the right one avoids the creation of a chain in G_1 (so $d(G^*) = 1$).

Chapter 5
Optimal Play in Loony Endgames of Dots-and-Boxes

In Chapter 2 we proved that the player in control can claim, at least, half the boxes of the loony endgame. However, the proof was non-constructive, and besides usually he can do better than that. We expose some known results about optimal play: how each player must proceed in a loony endgame, and the net gain of boxes that the player in control can obtain, called the value of the game. In general it is not easy to compute the value of a game, so we introduce the controlled value, which is easier to compute, and in some situations coincides with the actual value. We have a much better understanding of a loony endgame if it is simple.

1. Value of a Game

We introduce some definitions and notation needed for this chapter, including the key notion of value of a game.

As we are considering loony endgames in this chapter, we will repeatedly refer to the player in control and his opponent using the same notation as the one used by Berlekamp in [**Ber00**].

NOTATION 64. *We call* Right *the player in control, and* Left *his opponent.*

When we have chains of length 3 in a loony endgame things tend to get more complicated, so it is convenient to distinguish between 3-chains and longer chains.

DEFINITION 65. *A* very long chain *is a chain of length* ≥ 4.

Some of the components of a non-simple loony endgame can be quite complicated. In Chapter 2 we introduced earmuffs, dippers and shackles (see Figures 12, 13 and 14); we will show two other examples in Figures 24 and 25. In any case, we already observed (Proposition 41) that whatever object Left offers behaves as either a loop (only when it is an independent loop) or a string (in any other case).

DEFINITION 66. *A* simple object *is either a string or a loop.*

To determine the best score differential that Right can obtain in a loony endgame, we consider the following definition.

Definition 67. *The* value *of a loony endgame G, $V(G)$, is the net gain for the player in control (that is, number of boxes claimed by him minus number of boxes claimed by his opponent) if he plays optimally.*

(Note that, although we do not need it here, we can extend the definition of value to arbitrary games by just replacing "player in control" by "previous player")

Two questions arise when faced with a loony endgame: which is the optimal play strategy for each player, and which is its value. Observe that, for Left, choosing an optimal move means choosing which string or loop to offer to his opponent. We proved in Proposition 56 that, if Right plays canonically, when offered a string (resp. loop) he will claim, at least, all the boxes but the last two (resp, four). Therefore, for Right, his only decision is if he must keep control by refusing those last boxes, or give up control by taking all the offered boxes (what is sometimes called "being greedy").

Notation 68. *Given a loony endgame G and a string or independent loop H, we write $V(G|H)$ to denote the net gain by Right assuming that we impose on Left to offer H in his next move. If, moreover, we assume that Right is enforced to keep control (and from then on both players play optimally), we write $V_C(G|H)$ to denote his net gain. On the other hand, if we assume that Right is enforced to take all boxes from H, thus giving up control, we will write $V_G(G|H)$ to denote his net gain.*

Since Left will offer the loop or string which minimizes the value of the game, we have that

$$V(G) = \min_{H \in \mathcal{H}} V(G|H)$$

where $\mathcal{H}$ is the set of all strings and loops in the loony endgame that Left can offer. On the other hand, as Right will keep control or not depending on which choice guarantees him a higher net gain,

$$V(G|H) = max\{V_C(G|H), V_G(G|H)\}$$

We close this section with a basic result that we already saw in Chapter 2 in a different formulation.

Proposition 69. *For any loony endgame G, $V(G) \geq 0$.*

Proof. Theorem 34 says that the Right will claim, at least, half the boxes of any simple loony endgame. As noted in subsection 1.2 of chapter 3, we can prove that the proposition also holds for non-simple loony endgames by replacing "chain" by "string" in its proof. □

2. Optimal Play by Left

Our goal in this section is to determine which simple object is an optimal offering by Left.

2.1. "Man-in-the-middle". In this subsection we describe a powerful technique that is very useful to prove some results. The idea is simple: if you copy the moves of a perfect player, but with some exceptions, and obtain the same score as him, the moves you played are no worse that the ones played by the perfect player.

The origin of the name *man-in-the-middle* is best understood with the following example. Imagine that you are playing against two chess masters. In Game 1 you play the black pieces and in Game 2 the white ones. You apply the strategy of copying the moves from one game to another in this way: Master 1, playing white, makes the opening move in Game 1. You play the same opening move on Game 2, and wait for the reply of Master 2 in that game. Then you copy that move on Game 1 and wait for the reply of Master 1, and so on. In this way you will either draw against both masters or win against one of them. In fact, the two masters are playing a game between themselves. You are only the *man-in-the-middle*.

Using this idea, we can prove some results with the following procedure: we will suppose that a player (the man-in-the-middle) plays the same game of Dots-and-Boxes against two perfect players (known as *gurus*) but playing different roles in each game: in one game he is the previous player and in the other he is the next player. Instead of copying all moves, as in the chess example, he will substitute some moves by moves which we claim are no worse than the ones being substituted. If, at the end of the game, the man-in-the-middle has not claimed less boxes in the game where he has not copied all the moves than the guru in the other game, we will have proven our point.

2.2. Optimal Chain and Loop Offering Order. Our first application of the man-in-the-middle technique is to prove that Left must offer the independent chains in increasing order of length. Loops must also be offered in increasing order of length.

Proposition 70. *i) Given a loony endgame G containing two independent chains c and c', with respective lengths $k < k'$, it is never worse for Left to offer c than c', i.e., $V(G|c) \leq V(G|c')$.*

ii) Given a loony endgame G containing two independent loops l and l' of respective lengths $k < k'$, it is never worse for Left to offer l than l', i.e., $V(G|l) \leq V(G|l')$.

Proof. We prove the result for independent chains. The proof for loops is analogous.

Assume the man-in-the-middle is playing the same game against two gurus. He copies all the moves of the gurus, except in the following situation: if Guru 1 offers the longer chain c' in Game 1 when no guru has still offered c in his game, then the man-in-the-middle offers the shorter chain c in Game 2. Guru 2 will either keep control by declining the last 2 boxes, or claim all of them. The man-in-the-middle does likewise in Game 1. At this point, the man-in-the-middle has claimed $k' - k$ more boxes in Game 1 than Guru 2 in Game 2. The man-in-the-middle continues copying moves.

At some point before the end of the game, one of these two things will happen first:

i) Guru 1 offers c in Game 1. Then the man-in-the middle offers c' in Game 2, waits for the reply of Guru 2, and follows the same strategy (keep control or claim all) in Game 1. As a result, the man-in-the middle has the same score in Game 2 as Guru 1 in Game 1.
ii) Guru 2 offers c' in Game 2. Then the man-in-the middle offers c in Game 1, waits for the reply of Guru 1, and follows the same strategy (keep control or claim all) in Game 1. As a result, the man-in-the middle has the same score in Game 2 as Guru 1 in Game 1.

In either case, from that point on until the end of the game, the man-in-the-middle copies the moves of the corresponding guru. As a result, the number of boxes claimed by the man-in-the-middle in Game 1 is no less than the number of boxes claimed by Guru 2 in Game 2, which proves that his initial deviation from the copy strategy (offering c) is no worse than an optimal move (the offering of c' by Guru 1).

□

Note that we have not proved that offering both chains is equivalent, only that offering the shorter chain is, at least, as good as offering the longer one. There may be a better strategy with which a strictly better result can be obtained by offering the shorter chain. (In fact, if that is the case, the situation that triggers that the man-in-the-middle does not copy the move by Guru 1 above will not happen, as we assume that the gurus are perfect players). An example is the following: consider a game G with three components, a 4-loop, a 3-chain and an n-chain, where $n \geq 6$. In this case it is strictly better for Left to offer the 3-chain than the n-chain: $V(G|\text{3-chain}) = V_C(G|\text{3-chain}) = n - 5$, while $V(G|n\text{-chain}) = V_C(G|n\text{-chain}) = n - 1$.

Another known result that applies to arbitrary loony endgames is that it is almost always preferable (better or equal) for Left to offer a loop than a longer chain.

Proposition 71. *In any loony endgame G containing an independent k-chain c and an independent k'-loop l, $k' \leq k$ and $k > 4$, $V(G|l) \leq V(G|c)$.*

The proof, which uses the man-in-the-middle argument, is available in [**Sco**].

2.3. Optimal Play by Left in Simple Loony Endgames.

Corollary 72. *In any simple loony endgame, either offering the shortest chain or offering the shortest loop is an optimal move by Left.*

Proof. Since the loony endgame is simple, Left can only offer independent chains or loops, and the result is a consequence of Proposition 70. □

We can also improve Proposition 71 for simple loony endgames, since it is always better for Left to offer any loop than any very long chain.

Proposition 73. *If G is a simple loony endgame containing a loop l and a very long chain c, then $V(G|l) \leq V(G|c)$.*

The proof is available in [**Buzz**].

COROLLARY 74. *Given a simple loony endgame G,*

i) If G contains no loops, an optimal move for Left is to offer the shortest chain c, i.e., $V(G) = V(G|c)$.
ii) If G contains loops but no 3-chains, an optimal move for Left is to offer the shortest loop l, i.e., $V(G) = V(G|l)$.
iii) If G contains both loops and 3-chains, either offering a 3-chain c or offering the shortest loop l is an optimal move for Left, i.e., $V(G) = min\{V(G|c), V(G|l)\}$.

The previous corollary settles the question of finding an optimal move for Left in simple loony endgames where all the chains are very long, or where there are no loops present. When a simple loony endgame contains both 3-chains and loops, to determine if Left has to offer a 3-chain or the shortest loop we can refer to [**Buzz**], where Buzzard and Ciere give an algorithm to determine an optimal offering.

3. Optimal Play by Right

Now let us face the question of what must be the answer of Right when offered a string or loop. As we already observed, his only decision when offered a string is if he must decline the last two boxes in order to keep control, or if it is better for him to claim them and give up control. Analogously, when offered a loop he only has to choose between declining or not the last four boxes.

3.1. When to Keep Control. As we showed in the proof of Theorem 34, the decision of keeping control or claiming all boxes depends on the value of the rest of the game.

PROPOSITION 75. *Given a loony endgame G, assume Left offers a string or a loop H. Let G' be the rest of the game, $G = H + G'$.*

i) If H is a string, Right's optimal strategy is to keep control if the value of the rest of the game is at least 2, and to claim all the boxes of the chain otherwise, i.e.,

$$V(G|H) = V_C(G|H) \Leftrightarrow V(G') \geq 2$$

.

ii) If H is a loop, Right's optimal strategy is to keep control if the value of the rest of the game is at least 4, and to claim all the boxes of the loop otherwise, i.e.,

$$V(G|H) = V_C(G|H) \Leftrightarrow V(G') \geq 4$$

.

PROOF. If H is a string of length c, keeping control means a net gain in the string of $c-4$ boxes ($c-2$ claimed by Right, minus 2 claimed by Left) and being in control in the remaining of the game, G'. That gives Right a total net gain of $V_C(G|H) = c - 4 + V(G')$, while claiming all c boxes and giving up control to his opponent gives a net gain of $V_G(G|H) = c - V(G')$. Therefore it is optimal for Right to keep control when $c - 4 + V(G') \geq c - V(G')$, that is, when $V(G') \geq 2$.

On the other hand, if H is a loop of length l, keeping control gives Right a net gain of $V_C(G|H) = l - 8 + V(G')$, while claiming all the boxes and giving up control to his opponent gives him a net gain of $V_G(G|H) = l - V(G')$. Therefore it is optimal for Right to keep control when $V(G') \geq 4$.

□

Note that when H is a string and $V(G') = 2$, as well as when H is a loop and $V(G') = 4$, it is indifferent to keep or give up control. We can canonically assume that Right keeps control in this case.

4. Finding the Value of a Game

Since we have just shown that the optimal choice by Right depends on the value of the loony endgame that remains after the string or loop just offered by Left is claimed, we have reduced the problem of finding an optimal move to finding the value of a loony endgame. In fact, in order to apply Proposition 75, it is enough to know if the value of a loony endgame G is either $V(G) \leq 2$, $2 \leq V(G) \leq 4$ or $V(G) \geq 4$, but finding the value of a game is interesting per se. For instance, if Left is ahead in the score by t boxes when reaching the loony endgame G, Right can win the game iff $V(G) > t$. (Recall that Left can be ahead in the score due to sacrifices by Right and/or the short chains' exchange that precedes the loony endgames, as described in Section 4 of Chapter 2).

4.1. Fully Controlled Value. A possible strategy, described in Definition 32, is to keep control. Although it was defined for simple loony endgames, it is easily generalizable by considering that Right declines the last 2 boxes of any string and the last 4 of each independent loop. This strategy is a winning one when playing Nimstring (in fact, it is the only winning strategy), but it is not always the case in Dots-and-Boxes. An easy example is a game consisting of a 4-loop and a 3-chain. The optimal move by Left is to offer the loop. Right must claim all 4 boxes and offer the chain, thus winning 4-3, while if he kept control by declining the 4 boxes of the loop (and then claiming the chain) he would lose 3-4.

Definition 76. *The* fully controlled value *of a loony endgame G, $FCV(G)$, is the net gain that the player in control obtains if he keeps control until the end of the game.*

As we showed in the proof of Proposition 33, if b is the total number of boxes of the loony endgame, n the number of strings offered and m the number of loops offered, Right will claim $b - 2n - 4m$ boxes with this strategy, and Left the remaining $2n + 4m$ boxes, so $FCV(G) = b - 4n - 8m$.

While n is fixed, $n = \frac{1}{2}(v - 2j)$, as shown in Proposition 43 (as the number d of double-crosses played on strings is equal to the number n of strings offered),

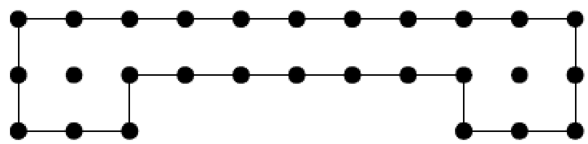

FIG. 23. An example of non-simple loony game where $CV(G) \geq 2$ but $V(G) > CV(G)$: in this pair of earmuffs $CV(G) = 2$ and $V(G) = 6$.

m depends on the order in which the strings and loops are offered by Left. For instance, when we have earmuffs (Figure 23) we can end up with $m = 1$ or $m = 2$, depending on what Left offers first: one of the loops or the *band* (string), resp., as we explained in Section 2.1 of Chapter 3.

PROPOSITION 77. *Given a loony endgame G where b uncaptured boxes remain, if Left offers n strings of respective lengths $c_1, \cdots, c_n$ and m independent loops of respective lengths $l_1, \cdots, l_m$, we have that*

$$FCV(G) = \sum_{i=1}^{n} c_i - 4n + \sum_{j=1}^{m} l_j - 8m = b - 4n - 8m$$

PROOF. Right must decline the last 2 string of each chain and the last 4 of each loop. Therefore a c-string contributes to the $FCV(G)$ by $(c-2)-2 = c-4$. An independent l-loop contributes to the $FCV(G)$ by $(l-4)-4 = l-8$. Adding it all, we have that $FCV(G) = (c_1 - 4) + \cdots + (c_n - 4) + (l_1 - 8) + \cdots + (l_m - 8)$.

On the other hand, since Left will claim 2 boxes per string and 4 per loop, for a total of $2n+4m$, Right's net gain is $FCV(G) = (b-2n-4m)-(2n+4m) = b-4n-8m$.

□

4.2. Controlled Value. Now we consider the following strategy: Right keeps control until offered the penultimate simple object of the loony endgame G. Then he plays optimally. Strictly speaking, having in consideration non-simple loony endgames, we should say that Right must keep control unless he is offered an object such that, if he claimed all the possible boxes, all that would remain in the game would be an independent chain, an independent loop or nothing at all. For instance, consider the earmuffs in Figure 23. If Left offers the band (central string), since if Right claimed all its boxes there would still remain two loops, Right is enforced to keep control and claim only 4 boxes of the string. Then Left will take the last 2 boxes of the string and offer one of the loops ("penultimate object"), and Right will be free to choose between keeping control or not.

Note that, in fact, the only choice that we are leaving Right if we enforce him to follow this strategy is when offered the second to last simple object, since he has to keep control in the prior turns, and it is always optimal to take all boxes when offered the last simple object.

PROPOSITION 78. *In a loony endgame G the optimal play by Right in his last turns is as follows:*

i) When offered a simple object such that, if he claimed all the offered boxes all that would remain would be either an independent chain, a loop or nothing, Right must claim all boxes if and only if the offered object is an independent loop and the remaining object is a 3-chain. Otherwise, he must keep control.
ii) When the game is reduced to a single independent chain or loop, Right must claim all of its boxes.

PROOF. Let H be the penultimate object being offered, in the sense that if Right claims all its boxes all that remains is an independent chain or loop G'. Obviously, when a player is offered G', his optimal move is to claim all the boxes. Therefore $V(G')$ is equal to the length of G'.

According to Proposition 75, if H is a string Right has to keep control if $V(G') \geq 2$. But that is always the case, as $V(G')$ is the length of the last chain or loop, which is, at least, 3. On the other hand, if H is a loop Right must keep control if $V(G') \geq 4$. So the only exception is when G' is a 3-chain. □

We are interested in finding the net gain by Right (i.e., boxes he claims minus boxes claimed by his opponent) when following the strategy described above.

DEFINITION 79. *The* controlled value *of a loony endgame G, $CV(G)$, is the net gain obtained by Right if the players play in the following way:*

i) Right is forced to keep control all the time, except in the case when he is offered an object such that, if he claimed all the possible boxes of it, all that would remain in the game would be an independent chain, an independent loop or nothing at all. In this case he chooses the best option.
ii) Left plays optimally throughout the whole endgame, and knowing that Right is going to play as described.

PROPOSITION 80. *In any loony endgame G, we have that $V(G) \geq CV(G)$.*

PROOF. Since Left is able to play optimally throughout all the loony endgame, the net gain by Right cannot be better than $V(G)$.

□

On the other hand, as Right is forced to keep control except in, at most, his last two turns, in some loony endgames this may coincide with his optimal (unrestricted) strategy (then $V(G) = CV(G)$) while in other loony endgames may not (and then $V(G) > CV(G)$). An example where $V(G) > CV(G)$ is a game G formed by 3 loops of length 4. When offered the first 4-loop, the optimal decision by Right would be to claim all 4 boxes, as the value of the rest of the game G' (the other two 4-loops) is $V(G') = 0$. But when playing the controlled value strategy, Right is compelled to keep control and decline the 4 boxes of the first loop. Therefore $V(G) = 4$, while $CV(G) = -4$.

One of the interesting questions we can ask is "when is $CV(G)$ equal to $V(G)$?". Or, what is the same, "when the strategy we just described is optimal?". Observe that for games G where $CV(G) = V(G)$, Right has an optimal strategy that is very easy to apply: he just needs to keep control until close to the end, and then apply Proposition 78.

4.3. Terminal Bonus.

DEFINITION 81. *The* terminal bonus *of a loony endgame G is $TB(G) = CV(G) - FCV(G)$.*

Firstly let us consider simple loony endgames.

PROPOSITION 82. *Let G be a simple loony endgame.*

i) $TB(G) = 8$ *iff the last two objects are loops.*
ii) $TB(G) = 6$ *iff the last two objects are a 3-chain and a loop, and in this case Left will offer the loop first.*
iii) $TB(G) = 4$ *otherwise.*

PROOF. By Proposition 78 we know that Right will keep control when offered the second to last object, except when it is a loop and what remains is a 3-chain. Therefore, except in that case, since the only difference between the FCV(G) and the CV(G) is that Right will claim all the boxes in the last object, the terminal bonus is 4 when the last object is a chain, and 8 when it is a loop.

In the exception, the optimal play by Left is to offer the loop first:

If Left offers the loop of length l, followed by a 3-chain, Right will take all the boxes (giving up control) and obtain a net gain of $l - 3$ boxes. On the other hand, if Left offers the 3-chain first, Right will keep control and obtain a net gain of $(1 + l) - 2 = l - 1$ boxes. So Left's optimal play is to offer the loop first, and $TB(G) = CV(G) - FCV(G) = (l - 3) - (l - 9) = 6$.

□

We can extend the previous result to arbitrary loony endgames.

We need to introduce the *eight*, which is a component with two 3-joints and 3 strings that go from one of the joints to the other (Figure 24).

THEOREM 83. *Let G be an arbitrary loony endgame.*

i) $TB(G) = 8$ *if the last object(s) offered are either a pair of shackles, a dipper, an eight or two independent loops.*
ii) $TB(G) = 6$ *if the last two objects are a 3-chain and a loop. In this case, Left will offer the loop first.*
iii) $TB(G) = 4$ *otherwise.*

PROOF. The last object must be an independent chain or loop, while the second last can be a string or an independent loop. By Proposition 78, Right will keep

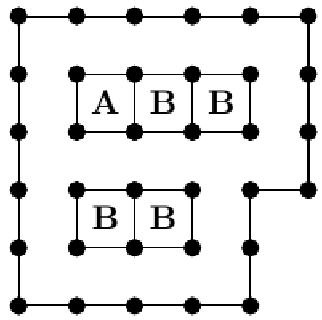

FIG. 24. An *eight* (the captured boxes do not form part of it): when Left offers one of the three strings that go from a 3-joint to the other, what will remain will be a loop.

control when offered the second last object unless it is a loop and what remains is a 3-chain. We can prove the theorem case by case, depending on the last two simple objects that Left offers.

a) A loop of length l, followed by a loop of length l'. In this case $CV(G) = CV_C(G) = (l-4+l')-4 = l+l'-8$. Therefore $TB(G) = CV(G) - FCV(G) = (l+l'-8)-(l+l'-16) = 8$.
b) A string of length c, followed by a loop of length l. Then $CV(G) = CV_C(G) = c-4+l$. Then we have $TB(G) = CV(G) - FCV(G) = (c-4+l)-(c+l-12) = 8$.
c) A loop of length l, followed by a 3-chain. This is the only case when Right does not keep control: $CV(G) = CV_G(G) = l-3$. Then we have $TB(G) = CV(G) - FCV(G) = (l-3)-(l-9) = 6$.
d) A string of length c, followed by an independent chain of length c'. Right will obtain a net gain of $CV(G) = CV_C(G) = (c-2+c')-2 = c+c'-4$. Therefore $TB(G) = CV(G) - FCV(G) = (c+c'-4)-(c+c'-8) = 4$.
e) An independent loop of length l, followed by a very long chain of length c. Then $CV(G) = CV_C(G) = l+c-8$. Then we have $TB(G) = CV(G) - FCV(G) = (l+c-8)-(l+c-12) = 4$.

Now let us consider all the possible terminal objects in the light of the cases considered above.

As Left can only offer two loops as last object if they are independent, case *a* corresponds only to the case where G is formed by two independent loops.

Unless G is simple, case *b* only happens when a string is attached to a loop. If the string has an end to the ground, G is a dipper. If the string has both ends attached to the loop, it can be to the same box (and G is a pair of shackles) or to two different boxes (and we have an eight). We already observed in Section 2.1 of Chapter 3 that dippers and shackles behave as a chain followed by a loop. We can easily verify that this is also the case for an eight. Therefore in all those cases (corresponding to *a* and *b*), $TB(G) = 8$.

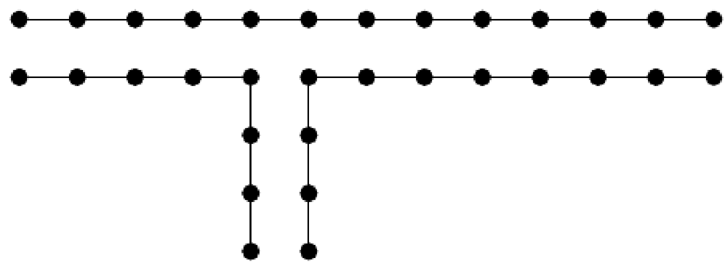

FIG. 25. A T: after offering one of the three legs (strings), what will remain is an independent chain.

In all the remaining cases, except in case c, that can only happen when G consists of a loop and an independent 3-chain, $TB(G) = 4$. An example of case d is a T, that is, a component consisting on 3-strings that go from a 3-joint to the ground (Figure 25), while case e corresponds only to a game consisting on a loop plus a very long independent chain.

□

COROLLARY 84. *For any loony endgame G, we have $4 \leq TB(G) \leq 8$.*

PROPOSITION 85. *Let G be a loony endgame. In order to minimize $CV(G)$, Left must maximize the number m of loops offered.*

PROOF. Using Proposition 77, $CV(G) = FCV(G)+TB(G) = b-4n-8m+TB(G)$. Since b and n are fixed, and $4 \leq TB(G) \leq 8$, Left must maximize the number m of loops offered in order to minimize $CV(G)$.

Another way to prove that Left must maximize m is to use $n = \frac{1}{2}(v - 2j)$ (Proposition 43) to obtain $CV(G) = b - 2v + 4j - 8m + TB(G)$, where b, v and j are fixed. □

We have shown how to determine the terminal bonus of an arbitrary loony endgame, but we need to know which objects will be offered last in order to apply Theorem 83. If the loony endgame is simple, the following proposition allows us to obtain the terminal value in a direct way.

PROPOSITION 86. *In a simple loony endgame G,*

i) if G contains a very long chain then $TB(G) = 4$.
ii) if G contains, at least, two 3-chains, then $TB(G) = 4$.
iii) if G consists of independent loops, then $TB(G) = 8$.
iv) if G has a 3-chain, and all the other components are independent loops, then $TB(G) = 6$.

PROOF. Left will minimize $CV(G)$. In order to do so, since $FCV(G)$ is independent of the way both players play, Left will try to minimize the terminal value. According to Proposition 82, he can play optimally by leaving a very long chain, if available, as last object to offer, so that $TB(G) = 4$. If all the components are loops, the last object will be a loop, so $TB(G) = 8$ no matter what Left does. Lastly, assume that G contains 3-chains and loops, and no very long chains. If there is more than one 3-chain, Left can accomplish $TB(G) = 4$ by leaving two of them as the last two objects. Otherwise, if there is only one 3-chain, Left will do best leaving it as last object (as shown in the lemma), as the penultimate will be a loop, which implies that $TB(G) = 6$.

□

Observe that, if G is simple, the number m of loops offered by Left is equal to the number of independent loops. In this case it is trivial to compute the fully controlled value using $FCV(G) = b - 4n - 8m$ (Proposition 77). Since we can use Proposition 86 to obtain $TB(G)$, the controlled value of a simple loony endgame can be easily determined as the sum of the fully controlled value and the terminal bonus.

4.4. Some Known Results. Under some conditions, $CV(G)$ and $V(G)$ turn out to be the same.

THEOREM 87. *Let G be a simple loony endgame. If $CV(G) \geq 2$ then $CV(G) = V(G)$.*

The proof is available in [**Sco**]. The simplest example that the bound is tight is a game G consisting of a 3-chain and two 6-loops, whose controlled value is $CV(G) = 1$ (as $FCV(G) = -5$ and $TB(G) = 6$) but $V(G) = 3$. The optimal play in G is as follows: Left offers the 3-chain, Right controls, Left offers a 6-loop, Right controls, Left offers the other 6-loop, Right claims the 6 boxes. The controlled value is not equal because, if Left knows that Right is enforced to control, he will offer a 6-loop first (instead of the 3-chain).

An example of non-simple loony game where $CV(G) \geq 2$ but $V(G) > CV(G)$ are the earmuffs in Figure 23. In that case $CV(G) = 2$, but $V(G) = 6$.

COROLLARY 88. *If G contains neither 3-chains nor loops of length 4 or 6, then $CV(G) = V(G)$.*

PROOF. From Proposition 77,

$$FCV(G) = \sum_{i=1}^{n} c_i - 4n + \sum_{j=1}^{m} l_j - 8m$$

If G is a simple loony endgame without any 3-chains,

$$\sum_{i=1}^{n} c_i - 4n = \sum_{i=1}^{n} (c_i - 4) \geq 0,$$

as each $c_i - 4$ is non-negative. Besides, if G contains no loops of length < 8, then

$$\sum_{j=1}^{m} l_j - 8m = \sum_{j=1}^{m} (l_j - 8) \geq 0.$$

Therefore $FCV(G) \geq 0$. Since $TB(G) \geq 4$ for any loony endgame G, $CV(G) = FCV(G) + TB(G) \geq 4$, and then, by Proposition 87, $CV(G) = V(G)$. □

Observe that if $CV(G) < 0$ we cannot have $CV(G) = V(G)$, since $V(G) \geq 0$ for any loony endgame G. Therefore, for simple loony endgames, only in the case $CV(G) \in \{0, 1\}$ we do not know if $CV(G)$ and $V(G)$ are equal or not.

For arbitrary loony endgames, Berlekamp and Scott [**Ber02**] proved that if $CV(G) \geq 10$, then $CV(G) = V(G)$. The proof is based on finding a strategy in which Left can force Right to stay in control until the end, thus being Right unable of obtaining a better net gain than $CV(G)$. Besides, the given bound is tight.

5. Solving Dots-and-Boxes

5.1. Complexity of Dots-and-Boxes. Many classic 2-player games are *PSPACE-complete*, as Reversi, Hex or Gobang (Five-in-a-row) or *EXPTIME-complete*, as Go, Chess or Checkers. The complexity of Dots-and-Boxes is so far unknown. We do not know if it belongs to *NP*, for instance.

However we know that computing the controlled value of loony endgames is *NP-hard*, as observed in [**Ber01**]. While in simple loony endgames we can easily compute the controlled value, in arbitrary loony endgames that is not the case, since the number of plays of independent loops cannot be determined easily. To minimise the controlled value, Left must maximize the number m of loops offered (Proposition 85). Determining the maximum number of pairwise (node-)disjoint cycles of an arbitrary graph is known to be an *NP-hard* problem. Eppstein [**Epp**] considers that this result can lead to prove that Dots-and-Boxes is also *NP-hard*:

> *Winning Ways* describes a generalized version of the game that is NP-hard, by a reduction from finding many vertex-disjoint cycles in graphs. The same result would seem to apply as well to positions from the actual game, by specializing their reduction to trivalent planar graphs. (This is very closely related to, but not quite the same as, maximum independent sets in maximal planar graphs.)

Hearn [**Hea06**] considers that Dots-and-Boxes, as a 2-player game of bounded length, would probably be *PSPACE-complete*.

Nimstring can be considered a particular case of Strings-and-Coins. Given any game G of Nimstring with n vertices, consider a game G' of Strings-and-Coins such that $G' = G + H$, where H is a chain of length $n + 1$. The winner in G' will be the player that claims the $(n + 1)$-chain, no matter how many boxes he sacrifices in G to achieve his goal. The player that forces his opponent to play in H is the winner, and he can do so if and only if he is able to win the Nimstring game G (i.e., forcing his opponent to claim the last boxes of G and offer the $(n + 1)$-chain). Therefore Strings-and-Coins is, at least, as difficult as Nimstring, since an algorithm to solve it could be used to solve Nimstring by adding to any Nimstring instance an $(n+1)$-chain to obtain an instance of Strings-and-Coins, and then use the algorithm to solve it.

Let us consider Nimstring restricted to dual Dots-and-Boxes games (which are, in particular, planar graphs, with maximum degree 4, and no cycles of odd length), and call this game *Nimdots* (that would simply be Dots-and-Boxes played with the normal play condition). The problem of solving Nimdots can be reduced to the problem of solving Dots-and-Boxes by the same reduction between Nimstring and String-and-Coins.

If we call $NIMSTRING$ the problem of finding an optimal move in an arbitrary Nimstring game, and in an analogous way define the other problems, we have that

$$NIMDOTS \leq_p NIMSTRING \leq_p STRINGS\_AND\_COINS$$

$$NIMDOTS \leq_p DOTS\_AND\_BOXES \leq_p STRINGS\_AND\_COINS$$

The last reduction is trivial, as Strings-and-Coins is a generalisation of Dots-and-Boxes.

5.2. Dots-and-Boxes on Small Boards. A winning strategy for 3x3 Dots-and-Boxes (meaning 3x3 boxes) is shown in *Winning Ways* [**Ber01**].

David Wilson proved that 4x4 Dots-and-Boxes is a draw. As a curiosity, all first moves are equivalent (i.e., guarantee a tie). You can play the opening moves against Wilson's analyser at [**Wil**].

Barker and Korf [**Bar**] proved recently that 4x5 Dots-and-Boxes is also a draw.

William Fraser [**Fra**], winner of the category of Dots-and-Boxes in the 18th Computer Olympiad held at Leiden University on June 30th, 2015, with his computer player The Shark, claims to have found an opening move in 5x5 Dots-and-Boxes (in fact, 8 symmetrically equivalent moves) that guarantee a victory to the first player (by just one box, 13-12). As far as we know, his results remain unpublished.

Discussion, Conclusions and Future Work

Some considerations about our work:

This work was made with the idea of being completely self-contained, and does not assume any previous knowledge in the field of Combinatorial Game Theory. Our contribution is mainly to fill in some gaps in the literature, as explained in the Introduction.

Chapter 1:

In Chapter 1 we introduced the basic concepts of Impartial Combinatorial Games. Initially we considered making an introduction to general games, but that option was discarded to shorten the chapter and because we would have introduced many concepts that were not needed. Some of the included results for impartial games could have been derived from results in general Combinatorial Theory but, since we had not presented them, we could not use them to prove our results.

Chapter 2:

We tried in most cases to keep the same concept names and notation found in the literature throughout our work, mainly as the one found in books like [**Ber00, Ber01, Alb**]. One notable exception is not defining half-hearted handouts as loony moves. As we already mentioned, this does not change anything in the theory, and we thought it was preferable to present the results in that way.

The Short Chains' Exchange (Proposition 39) is an original contribution.

In this Chapter we could have included some theorems and equivalences from [**Ber01**] that allow us to simplify Nimstring positions. There are also some tables with the nim-values of some kinds of components (for instance, of any T like the one in Figure 25, or some rectangular arrays of boxes). Future work could include finding new equivalences and/or nim-values of other kinds of components.

Chapter 3:

We used the number d of double crosses played in strings, as a formal substitute for the intuitive concept of *chain*.

We provided a generalisation of the *chain rule* available in *Winning Ways*, the *General Chain Formula* (Theorem 47), as well as Corollary 49, which tell us we can (almost) arbitrarily fill the board when d is resolved.

A questionable decision: we started considering simple loony endgames (in Chapter 2), instead of arbitrary loony endgames from the very beginning, to help the reader.

Chapter 4:

In the first section we compiled several canonical restrictions. The second section is completely original, based on Berlekamp's assertion in [**Ber00**] that components which nim-value is 0 or $*1$ are resolved, while BIG components are not. We proposed a definition of *resolved component*, since it was not rigorously defined, that allowed us to prove Berlekamp's claim in the formulations of Proposition 59, Theorem 61, and Proposition 62.

Chapter 5:

We offer an explicit way to find the Terminal Value in arbitrary loony endgames according to the objects offered last (Theorem 83). We had to coin the (perhaps not very fortunate) term *eight* for one kind of component, while others were already in use (earmuffs, dipper, shackles).

An interesting future work could be to try to find if there is another interesting subset of loony endgames, other than simple loony endgames, where the condition on G such that $CV(G) = V(G)$ is not as strong as with arbitrary loony endgames.

References

[Alb] Michael H. Albert, Richard J. Nowakowski, David Wolfe, “Lessons in Play”, A K Peters Ltd. (2007).

[Bar] Joseph K. Barker, Richard E Korf, “Solving Dots-and-Boxes”, in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI Press (2012).

[Ber94] Elwyn Berlekamp, David Wolfe, “Mathematical Go: Chilling Gets the Last Point”, A K Peters Ltd. (1994).

[Ber00] Elwyn R. Berlekamp, “The Dots-and-Boxes Game: Sophisticated Child’s Play”, A K Peters Ltd. (2000).

[Ber01] Elwyn R. Berlekamp, John H. Conway, Richard Guy, “Winning ways for your Mathematical Plays, 2nd edition”, A K Peters Ltd., volumes 1 (2001) & 3 (2003).

[Ber02] Elwyn R. Berlekamp, Katherine Scott, “Forcing Your Opponent to Stay in Control of a Loony Dots-and-Boxes Endgame”, in “More Games of No Chance”, edited by Richard J. Nowakowski, *MSRI Publications - Volume 42*, Cambridge University Press (2002).

[Bou] Charles L. Bouton, “Nim, a game with a complete mathematical theory”, *The Annals of Mathematics, Vol.3, No 1/4, p. 35-39* (1901-02).

[Buzz] Kevin Buzzard, Michael Ciere, “Playing Simple Dots and Boxes Endgames Optimally”, in *Integers: Electronic Journal of Combinatorial Number Theory*, volume 14 (2014).

[Con] John H. Conway, “On Numbers and Games”, Academic Press Inc. (1976).

[Dem] Erik D. Demaine, Robert A. Hearn, “Playing Games with Algorithms: Algorithmic Combinatorial Game Theory” in “Games of No Chance 3”, edited by Michael H. Albert and Richard J. Nowakowski, *MSRI Publications - Volume 56*, Cambridge University Press (2009).

[Elk] N.D. Elkies, “On Numbers and Endgames”, in “Games of No Chance”, edited by Richard J. Nowakowski, *MSRI Publications - Volume 29*, Cambridge University Press (1996).

[Epp] David Eppstein, “Computational Complexity of Games and Puzzles”, available at `www.ics.uci.edu/~eppstein/cgt/hard.html`

[Fra] William Fraser, note available at `http://www.littlegolem.net/jsp/forum/topic2.jsp?forum=110&topic=148`

[Gru] P. M. Grundy, “Mathematics and Games”, at *Eureka: The Archimedeans’ Journal, vol. 27, p. 6-8* (1964), originally published in vol. 2 (1939).

[Hea06] Robert A. Hearn, “Games, Puzzles and Computation”, PhD thesis, Massachussets Institute of Technology (2006).

[Hea09] Robert A. Hearn, Erik D. Demaine, “Games, Puzzles, and Computation”, A K Peters Ltd. (2009).

[Luc] Édouard Lucas, “La Pipopipette - Jeu de Combinaisons”, *Jeux Scientifiques (pour servir à l’Histoire, à l’Enseignement et à la Pratique du Calcul et du Dessin), première série, №2*, Chambon & Baye, Paris (1889).

[Sch] Dierk Schleicher, Michael Stoll, “An Introduction to Conway’s Games and Numbers”, *Mosc. Math. J., 6:2* (2006).

[Sco] Katherine W. Scott, “Loony Endgames in Dots & Boxes”, MS Thesis, University of California at Berkeley (2000).

[Sie] Aaron N. Siegel, “Combinatorial Game Theory”, *Graduate Studies in Mathematics, vol. 146*, American Mathematical Society (2013).

[Spr] R. Sprague, “Über Mathematische Kampfspiele”, in *The Tôhoku Mathematical Journal, First Series, Vol. 41, p.438-444*, Mathematical Institute of Tôhoku University (1935).

[Wil] David Wilson, `http://wilson.engr.wisc.edu/boxes/play.php?game=4x4`